Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal (Eds.)

LNCS 15301

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part I

Springer

MOREMEDIA

Lecture Notes in Computer Science **15301**

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal
Editors

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part I

Springer

Editors
Apostolos Antonacopoulos
University of Salford
Salford, UK

Subhasis Chaudhuri
Indian Institute of Technology Bombay
Mumbai, India

Rama Chellappa
Johns Hopkins University
Baltimore, MD, USA

Cheng-Lin Liu
Chinese Academy of Sciences
Beijing, China

Saumik Bhattacharya
IIT Kharagpur
Kharagpur, India

Umapada Pal
Indian Statistical Institute Kolkata
Kolkata, India

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-78106-3 ISBN 978-3-031-78107-0 (eBook)
https://doi.org/10.1007/978-3-031-78107-0

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

President's Address

On behalf of the Executive Committee of the International Association for Pattern Recognition (IAPR), I am pleased to welcome you to the 27th International Conference on Pattern Recognition (ICPR 2024), the main scientific event of the IAPR.

After a completely digital ICPR in the middle of the COVID pandemic and the first hybrid version in 2022, we can now enjoy a fully back-to-normal ICPR this year. I look forward to hearing inspirational talks and keynotes, catching up with colleagues during the breaks and making new contacts in an informal way. At the same time, the conference landscape has changed. Hybrid meetings have made their entrance and will continue. It is exciting to experience how this will influence the conference. Planning for a major event like ICPR must take place over a period of several years. This means many decisions had to be made under a cloud of uncertainty, adding to the already large effort needed to produce a successful conference. It is with enormous gratitude, then, that we must thank the team of organizers for their hard work, flexibility, and creativity in organizing this ICPR. ICPR always provides a wonderful opportunity for the community to gather together. I can think of no better location than Kolkata to renew the bonds of our international research community.

Each ICPR is a bit different owing to the vision of its organizing committee. For 2024, the conference has six different tracks reflecting major themes in pattern recognition: Artificial Intelligence, Pattern Recognition and Machine Learning; Computer and Robot Vision; Image, Speech, Signal and Video Processing; Biometrics and Human Computer Interaction; Document Analysis and Recognition; and Biomedical Imaging and Bioinformatics. This reflects the richness of our field. ICPR 2024 also features two dozen workshops, seven tutorials, and 15 competitions; there is something for everyone. Many thanks to those who are leading these activities, which together add significant value to attending ICPR, whether in person or virtually. Because it is important for ICPR to be as accessible as possible to colleagues from all around the world, we are pleased that the IAPR, working with the ICPR organizers, is continuing our practice of awarding travel stipends to a number of early-career authors who demonstrate financial need. Last but not least, we are thankful to the Springer LNCS team for their effort to publish these proceedings.

Among the presentations from distinguished keynote speakers, we are looking forward to the three IAPR Prize Lectures at ICPR 2024. This year we honor the achievements of Tin Kam Ho (IBM Research) with the IAPR's most prestigious King-Sun Fu Prize "for pioneering contributions to multi-classifier systems, random decision forests, and data complexity analysis". The King-Sun Fu Prize is given in recognition of an outstanding technical contribution to the field of pattern recognition. It honors the memory of Professor King-Sun Fu who was instrumental in the founding of IAPR, served as its first president, and is widely recognized for his extensive contributions to the field of pattern recognition.

The Maria Petrou Prize is given to a living female scientist/engineer who has made substantial contributions to the field of Pattern Recognition and whose past contributions, current research activity and future potential may be regarded as a model to both aspiring and established researchers. It honours the memory of Professor Maria Petrou as a scientist of the first rank, and particularly her role as a pioneer for women researchers. This year, the Maria Petrou Prize is given to Guoying Zhao (University of Oulu), "for contributions to video analysis for facial micro-behavior recognition and remote biosignal reading (RPPG) for heart rate analysis and face anti-spoofing".

The J.K. Aggarwal Prize is given to a young scientist who has brought a substantial contribution to a field that is relevant to the IAPR community and whose research work has had a major impact on the field. Professor Aggarwal is widely recognized for his extensive contributions to the field of pattern recognition and for his participation in IAPR's activities. This year, the J.K. Aggarwal Prize goes to Xiaolong Wang (UC San Diego) "for groundbreaking contributions to advancing visual representation learning, utilizing self-supervised and attention-based models to establish fundamental frameworks for creating versatile, general-purpose pattern recognition systems".

During the conference we will also recognize 21 new IAPR Fellows selected from a field of very strong candidates. In addition, a number of Best Scientific Paper and Best Student Paper awards will be presented, along with the Best Industry Related Paper Award and the Piero Zamperoni Best Student Paper Award. Congratulations to the recipients of these very well-deserved awards!

I would like to close by again thanking everyone involved in making ICPR 2024 a tremendous success; your hard work is deeply appreciated. These thanks extend to all who chaired the various aspects of the conference and the associated workshops, my ExCo colleagues, and the IAPR Standing and Technical Committees. Linda O'Gorman, the IAPR Secretariat, deserves special recognition for her experience, historical perspective, and attention to detail when it comes to supporting many of the IAPR's most important activities. Her tasks became so numerous that she recently got support from Carolyn Buckley (layout, newsletter), Ugur Halici (ICPR matters), and Rosemary Stramka (secretariat). The IAPR website got a completely new design. Ed Sobczak has taken care of our web presence for so many years already. A big thank you to all of you!

This is, of course, the 27th ICPR conference. Knowing that ICPR is organized every two years, and that the first conference in the series (1973!) pre-dated the formal founding of the IAPR by a few years, it is also exciting to consider that we are celebrating over 50 years of ICPR and at the same time approaching the official IAPR 50th anniversary in 2028: you'll get all information you need at ICPR 2024. In the meantime, I offer my thanks and my best wishes to all who are involved in supporting the IAPR throughout the world.

September 2024

Arjan Kuijper
President of the IAPR

Preface

It is our great pleasure to welcome you to the proceedings of the 27th International Conference on Pattern Recognition (ICPR 2024), held in Kolkata, India. The city, formerly known as 'Calcutta', is the home of the fabled Indian Statistical Institute (ISI), which has been at the forefront of statistical pattern recognition for almost a century. Concepts like the Mahalanobis distance, Bhattacharyya bound, Cramer–Rao bound, and Fisher–Rao metric were invented by pioneers associated with ISI. The first ICPR (called IJCPR then) was held in 1973, and the second in 1974. Subsequently, ICPR has been held every other year. The International Association for Pattern Recognition (IAPR) was founded in 1978 and became the sponsor of the ICPR series. Over the past 50 years, ICPR has attracted huge numbers of scientists, engineers and students from all over the world and contributed to advancing research, development and applications in pattern recognition technology.

ICPR 2024 was held at the Biswa Bangla Convention Centre, one of the largest such facilities in South Asia, situated just 7 kilometers from Kolkata Airport (CCU). According to ChatGPT "Kolkata is often called the 'Cultural Capital of India'. The city has a deep connection to literature, music, theater, and art. It was home to Nobel laureate Rabindranath Tagore, and the Bengali film industry has produced globally renowned filmmakers like Satyajit Ray. The city boasts remarkable colonial architecture, with landmarks like Victoria Memorial, Howrah Bridge, and the Indian Museum (the oldest and largest museum in India). Kolkata's streets are dotted with old mansions and buildings that tell stories of its colonial past. Walking through the city can feel like stepping back into a different era. Finally, Kolkata is also known for its street food."

ICPR 2024 followed a two-round paper submission format. We received a total of 2135 papers (1501 papers in round-1 submissions, and 634 papers in round-2 submissions). Each paper, on average, received 2.84 reviews, in single-blind mode. For the first-round papers we had a rebuttal option available to authors.

In total, 945 papers (669 from round-1 and 276 from round-2) were accepted for presentation, resulting in an acceptance rate of 44.26%, which is consistent with previous ICPR events. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

The main conference ran over December 2–5, 2024. The main program included the presentation of 188 oral papers (19.89% of the accepted papers), 757 poster papers and 12 competition papers (out of 15 submitted). A total 10 oral sessions were held concurrently in four meeting rooms with a total of 40 oral sessions. In total 24 workshops and 7 tutorials were held on December 1, 2024.

The plenary sessions included three prize lectures and three invited presentations. The prize lectures were delivered by Tin Kam Ho (IBM Research, USA; King Sun

Fu Prize winner), Xiaolong Wang (University of California, San Diego, USA; J.K. Aggarwal Prize winner), and Guoying Zhao (University of Oulu, Finland; Maria Petrou Prize winner). The invited speakers were Timothy Hospedales (University of Edinburgh, UK), Venu Govindaraju (University at Buffalo, USA), and Shuicheng Yan (Skywork AI, Singapore).

Several best paper awards were presented in ICPR: the Piero Zamperoni Award for the best paper authored by a student, the BIRPA Best Industry Related Paper Award, and the Best Paper Awards and Best Student Paper Awards for each of the six tracks of ICPR 2024.

The organization of such a large conference would not be possible without the help of many volunteers. Our special gratitude goes to the Program Chairs (Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa and Cheng-Lin Liu), for their leadership in organizing the program. Thanks to our Publication Chairs (Ananda S. Chowdhury and Wataru Ohyama) for handling the overwhelming workload of publishing the conference proceedings. We also thank our Competition Chairs (Richard Zanibbi, Lianwen Jin and Laurence Likforman-Sulem) for arranging 12 important competitions as part of ICPR 2024. We are thankful to our Workshop Chairs (P. Shivakumara, Stephanie Schuckers, Jean-Marc Ogier and Prabir Bhattacharya) and Tutorial Chairs (B.B. Chaudhuri, Michael R. Jenkin and Guoying Zhao) for arranging the workshops and tutorials on emerging topics. ICPR 2024, for the first time, held a Doctoral Consortium. We would like to thank our Doctoral Consortium Chairs (Véronique Eglin, Dan Lopresti and Mayank Vatsa) for organizing it.

Thanks go to the Track Chairs and the meta reviewers who devoted significant time to the review process and preparation of the program. We also sincerely thank the reviewers who provided valuable feedback to the authors.

Finally, we acknowledge the work of other conference committee members, like the Organizing Chairs and Organizing Committee Members, Finance Chairs, Award Chair, Sponsorship Chairs, and Exhibition and Demonstration Chairs, Visa Chair, Publicity Chairs, and Women in ICPR Chairs, whose efforts made this event successful. We also thank our event manager Alpcord Network for their help.

We hope that all the participants found the technical program informative and enjoyed the sights, culture and cuisine of Kolkata.

October 2024

Umapada Pal
Josef Kittler
Anil Jain

Organization

General Chairs

Umapada Pal	Indian Statistical Institute, Kolkata, India
Josef Kittler	University of Surrey, UK
Anil Jain	Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos	University of Salford, UK
Subhasis Chaudhuri	Indian Institute of Technology, Bombay, India
Rama Chellappa	Johns Hopkins University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China

Publication Chairs

Ananda S. Chowdhury	Jadavpur University, India
Wataru Ohyama	Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi	Rochester Institute of Technology, USA
Lianwen Jin	South China University of Technology, China
Laurence Likforman-Sulem	Télécom Paris, France

Workshop Chairs

P. Shivakumara	University of Salford, UK
Stephanie Schuckers	Clarkson University, USA
Jean-Marc Ogier	Université de la Rochelle, France
Prabir Bhattacharya	Concordia University, Canada

Tutorial Chairs

B. B. Chaudhuri	Indian Statistical Institute, Kolkata, India
Michael R. Jenkin	York University, Canada
Guoying Zhao	University of Oulu, Finland

Doctoral Consortium Chairs

Véronique Eglin	CNRS, France
Daniel P. Lopresti	Lehigh University, USA
Mayank Vatsa	Indian Institute of Technology, Jodhpur, India

Organizing Chairs

Saumik Bhattacharya	Indian Institute of Technology, Kharagpur, India
Palash Ghosal	Sikkim Manipal University, India

Organizing Committee

Santanu Phadikar	West Bengal University of Technology, India
SK Md Obaidullah	Aliah University, India
Sayantari Ghosh	National Institute of Technology Durgapur, India
Himadri Mukherjee	West Bengal State University, India
Nilamadhaba Tripathy	Clarivate Analytics, USA
Chayan Halder	West Bengal State University, India
Shibaprasad Sen	Techno Main Salt Lake, India

Finance Chairs

Kaushik Roy	West Bengal State University, India
Michael Blumenstein	University of Technology Sydney, Australia

Awards Committee Chair

Arpan Pal	Tata Consultancy Services, India

Sponsorship Chairs

P. J. Narayanan	Indian Institute of Technology, Hyderabad, India
Yasushi Yagi	Osaka University, Japan
Venu Govindaraju	University at Buffalo, USA
Alberto Bel Bimbo	Università di Firenze, Italy

Exhibition and Demonstration Chairs

Arjun Jain	FastCode AI, India
Agnimitra Biswas	National Institute of Technology, Silchar, India

International Liaison, Visa Chair

Balasubramanian Raman	Indian Institute of Technology, Roorkee, India

Publicity Chairs

Dipti Prasad Mukherjee	Indian Statistical Institute, Kolkata, India
Bob Fisher	University of Edinburgh, UK
Xiaojun Wu	Jiangnan University, China

Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian Statistical Institute, Kolkata, India

Event Manager

Alpcord Network

Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis

Larry O'Gorman	Nokia Bell Labs, USA
Dacheng Tao	University of Sydney, Australia
Petia Radeva	University of Barcelona, Spain
Susmita Mitra	Indian Statistical Institute, Kolkata, India
Jiliang Tang	Michigan State University, USA

Track Chairs – Computer and Robot Vision

C. V. Jawahar	International Institute of Information Technology (IIIT), Hyderabad, India
João Paulo Papa	São Paulo State University, Brazil
Maja Pantic	Imperial College London, UK
Gang Hua	Dolby Laboratories, USA
Junwei Han	Northwestern Polytechnical University, China

Track Chairs – Image, Speech, Signal and Video Processing

P. K. Biswas	Indian Institute of Technology, Kharagpur, India
Shang-Hong Lai	National Tsing Hua University, Taiwan
Hugo Jair Escalante	INAOE, CINVESTAV, Mexico
Sergio Escalera	Universitat de Barcelona, Spain
Prem Natarajan	University of Southern California, USA

Track Chairs – Biometrics and Human Computer Interaction

Richa Singh	Indian Institute of Technology, Jodhpur, India
Massimo Tistarelli	University of Sassari, Italy
Vishal Patel	Johns Hopkins University, USA
Wei-Shi Zheng	Sun Yat-sen University, China
Jian Wang	Snap, USA

Track Chairs – Document Analysis and Recognition

Xiang Bai	Huazhong University of Science and Technology, China
David Doermann	University at Buffalo, USA
Josep Llados	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

Track Chairs – Biomedical Imaging and Bioinformatics

Jayanta Mukhopadhyay	Indian Institute of Technology, Kharagpur, India
Xiaoyi Jiang	Universität Münster, Germany
Seong-Whan Lee	Korea University, Korea

Metareviewers (Conference Papers and Competition Papers)

Wael Abd-Almageed	University of Southern California, USA
Maya Aghaei	NHL Stenden University, Netherlands
Alireza Alaei	Southern Cross University, Australia
Rajagopalan N. Ambasamudram	Indian Institute of Technology, Madras, India
Suyash P. Awate	Indian Institute of Technology, Bombay, India
Inci M. Baytas	Bogazici University, Turkey
Aparna Bharati	Lehigh University, USA
Brojeshwar Bhowmick	Tata Consultancy Services, India
Jean-Christophe Burie	University of La Rochelle, France
Gustavo Carneiro	University of Surrey, UK
Chee Seng Chan	Universiti Malaya, Malaysia
Sumohana S. Channappayya	Indian Institute of Technology, Hyderabad, India
Dongdong Chen	Microsoft, USA
Shengyong Chen	Tianjin University of Technology, China
Jun Cheng	Institute for Infocomm Research, A*STAR, Singapore
Albert Clapés	University of Barcelona, Spain
Oscar Dalmau	Center for Research in Mathematics, Mexico

Tyler Derr	Vanderbilt University, USA
Abhinav Dhall	Indian Institute of Technology, Ropar, India
Bo Du	Wuhan University, China
Yuxuan Du	University of Sydney, Australia
Ayman S. El-Baz	University of Louisville, USA
Francisco Escolano	University of Alicante, Spain
Siamac Fazli	Nazarbayev University, Kazakhstan
Jianjiang Feng	Tsinghua University, China
Gernot A. Fink	TU Dortmund University, Germany
Alicia Fornes	CVC, Spain
Junbin Gao	University of Sydney, Australia
Yan Gao	Amazon, USA
Yongsheng Gao	Griffith University, Australia
Caren Han	University of Melbourne, Australia
Ran He	Institute of Automation, Chinese Academy of Sciences, China
Tin Kam Ho	IBM, USA
Di Huang	Beihang University, China
Kaizhu Huang	Duke Kunshan University, China
Donato Impedovo	University of Bari, Italy
Julio Jacques	University of Barcelona and Computer Vision Center, Spain
Lianwen Jin	South China University of Technology, China
Wei Jin	Emory University, USA
Danilo Samuel Jodas	São Paulo State University, Brazil
Manjunath V. Joshi	DA-IICT, India
Jayashree Kalpathy-Cramer	Massachusetts General Hospital, USA
Dimosthenis Karatzas	Computer Vision Centre, Spain
Hamid Karimi	Utah State University, USA
Baiying Lei	Shenzhen University, China
Guoqi Li	Chinese Academy of Sciences, and Peng Cheng Lab, China
Laurence Likforman-Sulem	Institut Polytechnique de Paris/Télécom Paris, France
Aishan Liu	Beihang University, China
Bo Liu	Bytedance, USA
Chen Liu	Clarkson University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China
Hongmin Liu	University of Science and Technology Beijing, China
Hui Liu	Michigan State University, USA

Jing Liu	Institute of Automation, Chinese Academy of Sciences, China
Li Liu	University of Oulu, Finland
Qingshan Liu	Nanjing University of Posts and Telecommunications, China
Adrian P. Lopez-Monroy	Centro de Investigacion en Matematicas AC, Mexico
Daniel P. Lopresti	Lehigh University, USA
Shijian Lu	Nanyang Technological University, Singapore
Yong Luo	Wuhan University, China
Andreas K. Maier	FAU Erlangen-Nuremberg, Germany
Davide Maltoni	University of Bologna, Italy
Hong Man	Stevens Institute of Technology, USA
Lingtong Min	Northwestern Polytechnical University, China
Paolo Napoletano	University of Milano-Bicocca, Italy
Kamal Nasrollahi	Milestone Systems, Aalborg University, Denmark
Marcos Ortega	University of A Coruña, Spain
Shivakumara Palaiahnakote	University of Salford, UK
P. Jonathon Phillips	NIST, USA
Filiberto Pla	University Jaume I, Spain
Ajit Rajwade	Indian Institute of Technology, Bombay, India
Shanmuganathan Raman	Indian Institute of Technology, Gandhinagar, India
Imran Razzak	UNSW, Australia
Beatriz Remeseiro	University of Oviedo, Spain
Gustavo Rohde	University of Virginia, USA
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sanjoy K. Saha	Jadavpur University, India
Joan Andreu Sánchez	Universitat Politècnica de València, Spain
Claudio F. Santos	UFSCar, Brazil
Shin'ichi Satoh	National Institute of Informatics, Japan
Stephanie Schuckers	Clarkson University, USA
Srirangaraj Setlur	University at Buffalo, SUNY, USA
Debdoot Sheet	Indian Institute of Technology, Kharagpur, India
Jun Shen	University of Wollongong, Australia
Li Shen	JD Explore Academy, China
Chen Shengyong	Zhejiang University of Technology and Tianjin University of Technology, China
Andy Song	RMIT University, Australia
Akihiro Sugimoto	National Institute of Informatics, Japan
Qianru Sun	Singapore Management University, Singapore
Arijit Sur	Indian Institute of Technology, Guwahati, India
Estefania Talavera	University of Twente, Netherlands

Wei Tang	University of Illinois at Chicago, USA
Joao M. Tavares	Universidade do Porto, Portugal
Jun Wan	NLPR, CASIA, China
Le Wang	Xi'an Jiaotong University, China
Lei Wang	Australian National University, Australia
Xiaoyang Wang	Tencent AI Lab, USA
Xinggang Wang	Huazhong University of Science and Technology, China
Xiao-Jun Wu	Jiangnan University, China
Yiding Yang	Bytedance, China
Xiwen Yao	Northwestern Polytechnical University, China
Xu-Cheng Yin	University of Science and Technology Beijing, China
Baosheng Yu	University of Sydney, Australia
Shiqi Yu	Southern University of Science and Technology, China
Xin Yuan	Westlake University, China
Yibing Zhan	JD Explore Academy, China
Jing Zhang	University of Sydney, Australia
Lefei Zhang	Wuhan University, China
Min-Ling Zhang	Southeast University, China
Wenbin Zhang	Florida International University, USA
Jiahuan Zhou	Peking University, China
Sanping Zhou	Xi'an Jiaotong University, China
Tianyi Zhou	University of Maryland, USA
Lei Zhu	Shandong Normal University, China
Pengfei Zhu	Tianjin University, China
Wangmeng Zuo	Harbin Institute of Technology, China

Reviewers (Competition Papers)

Liangcai Gao	Da-Han Wang
Mingxin Huang	Yang Xue
Lei Kang	Wentao Yang
Wenhui Liao	Jiaxin Zhang
Yuliang Liu	Yiwu Zhong
Yongxin Shi	

Reviewers (Conference Papers)

Aakanksha Aakanksha
Aayush Singla
Abdul Muqeet
Abhay Yadav
Abhijeet Vijay Nandedkar
Abhimanyu Sahu
Abhinav Rajvanshi
Abhisek Ray
Abhishek Shrivastava
Abhra Chaudhuri
Aditi Roy
Adriano Simonetto
Adrien Maglo
Ahmed Abdulkadir
Ahmed Boudissa
Ahmed Hamdi
Ahmed Rida Sekkat
Ahmed Sharafeldeen
Aiman Farooq
Aishwarya Venkataramanan
Ajay Kumar
Ajay Kumar Reddy Poreddy
Ajita Rattani
Ajoy Mondal
Akbar K.
Akbar Telikani
Akshay Agarwal
Akshit Jindal
Al Zadid Sultan Bin Habib
Albert Clapés
Alceu Britto
Alejandro Peña
Alessandro Ortis
Alessia Auriemma Citarella
Alexandre Stenger
Alexandros Sopasakis
Alexia Toumpa
Ali Khan
Alik Pramanick
Alireza Alaei
Alper Yilmaz
Aman Verma
Amit Bhardwaj
Amit More
Amit Nandedkar
Amitava Chatterjee
Amos L. Abbott
Amrita Mohan
Anand Mishra
Ananda S. Chowdhury
Anastasia Zakharova
Anastasios L. Kesidis
Andras Horvath
Andre Gustavo Hochuli
André P. Kelm
Andre Wyzykowski
Andrea Bottino
Andrea Lagorio
Andrea Torsello
Andreas Fischer
Andreas K. Maier
Andreu Girbau Xalabarder
Andrew Beng Jin Teoh
Andrew Shin
Andy J. Ma
Aneesh S. Chivukula
Ángela Casado-García
Anh Quoc Nguyen
Anindya Sen
Anirban Saha
Anjali Gautam
Ankan Bhattacharyya
Ankit Jha
Anna Scius-Bertrand
Annalisa Franco
Antoine Doucet
Antonino Staiano
Antonio Fernández
Antonio Parziale
Anu Singha
Anustup Choudhury
Anwesan Pal
Anwesha Sengupta
Archisman Adhikary
Arjan Kuijper
Arnab Kumar Das

Arnav Bhavsar
Arnav Varma
Arpita Dutta
Arshad Jamal
Artur Jordao
Arunkumar Chinnaswamy
Aryan Jadon
Aryaz Baradarani
Ashima Anand
Ashis Dhara
Ashish Phophalia
Ashok K. Bhateja
Ashutosh Vaish
Ashwani Kumar
Asifuzzaman Lasker
Atefeh Khoshkhahtinat
Athira Nambiar
Attilio Fiandrotti
Avandra S. Hemachandra
Avik Hati
Avinash Sharma
B. H. Shekar
B. Uma Shankar
Bala Krishna Thunakala
Balaji Tk
Balázs Pálffy
Banafsheh Adami
Bang-Dang Pham
Baochang Zhang
Baodi Liu
Bashirul Azam Biswas
Beiduo Chen
Benedikt Kottler
Beomseok Oh
Berkay Aydin
Berlin S. Shaheema
Bertrand Kerautret
Bettina Finzel
Bhavana Singh
Bibhas C. Dhara
Bilge Gunsel
Bin Chen
Bin Li
Bin Liu
Bin Yao
Bin-Bin Jia
Binbin Yong
Bindita Chaudhuri
Bindu Madhavi Tummala
Binh M. Le
Bi-Ru Dai
Bo Huang
Bo Jiang
Bob Zhang
Bowen Liu
Bowen Zhang
Boyang Zhang
Boyu Diao
Boyun Li
Brian M. Sadler
Bruce A. Maxwell
Bryan Bo Cao
Buddhika L. Semage
Bushra Jalil
Byeong-Seok Shin
Byung-Gyu Kim
Caihua Liu
Cairong Zhao
Camille Kurtz
Carlos A. Caetano
Carlos D. Martã-Nez-Hinarejos
Ce Wang
Cevahir Cigla
Chakravarthy Bhagvati
Chandrakanth Vipparla
Changchun Zhang
Changde Du
Changkun Ye
Changxu Cheng
Chao Fan
Chao Guo
Chao Qu
Chao Wen
Chayan Halder
Che-Jui Chang
Chen Feng
Chenan Wang
Cheng Yu
Chenghao Qian
Cheng-Lin Liu

Chengxu Liu
Chenru Jiang
Chensheng Peng
Chetan Ralekar
Chih-Wei Lin
Chih-Yi Chiu
Chinmay Sahu
Chintan Patel
Chintan Shah
Chiranjoy Chattopadhyay
Chong Wang
Choudhary Shyam Prakash
Christophe Charrier
Christos Smailis
Chuanwei Zhou
Chun-Ming Tsai
Chunpeng Wang
Ciro Russo
Claudio De Stefano
Claudio F. Santos
Claudio Marrocco
Connor Levenson
Constantine Dovrolis
Constantine Kotropoulos
Dai Shi
Dakshina Ranjan Kisku
Dan Anitei
Dandan Zhu
Daniela Pamplona
Danli Wang
Danqing Huang
Daoan Zhang
Daqing Hou
David A. Clausi
David Freire Obregon
David Münch
David Pujol Perich
Davide Marelli
De Zhang
Debalina Barik
Debapriya Roy (Kundu)
Debashis Das
Debashis Das Chakladar
Debi Prosad Dogra
Debraj D. Basu
Decheng Liu
Deen Dayal Mohan
Deep A. Patel
Deepak Kumar
Dengpan Liu
Denis Coquenet
Désiré Sidibé
Devesh Walawalkar
Dewan Md. Farid
Di Ming
Di Qiu
Di Yuan
Dian Jia
Dianmo Sheng
Diego Thomas
Diganta Saha
Dimitri Bulatov
Dimpy Varshni
Dingcheng Yang
Dipanjan Das
Dipanjyoti Paul
Divya Biligere Shivanna
Divya Saxena
Divya Sharma
Dmitrii Matveichev
Dmitry Minskiy
Dmitry V. Sorokin
Dong Zhang
Donghua Wang
Donglin Zhang
Dongming Wu
Dongqiangzi Ye
Dongqing Zou
Dongrui Liu
Dongyang Zhang
Dongzhan Zhou
Douglas Rodrigues
Duarte Folgado
Duc Minh Vo
Duoxuan Pei
Durai Arun Pannir Selvam
Durga Bhavani S.
Eckart Michaelsen
Elena Goyanes
Élodie Puybareau

Emanuele Vivoli
Emna Ghorbel
Enrique Naredo
Enyu Cai
Eric Patterson
Ernest Valveny
Eva Blanco-Mallo
Eva Breznik
Evangelos Sartinas
Fabio Solari
Fabiola De Marco
Fan Wang
Fangda Li
Fangyuan Lei
Fangzhou Lin
Fangzhou Luo
Fares Bougourzi
Farman Ali
Fatiha Mokdad
Fei Shen
Fei Teng
Fei Zhu
Feiyan Hu
Felipe Gomes Oliveira
Feng Li
Fengbei Liu
Fenghua Zhu
Fillipe D. M. De Souza
Flavio Piccoli
Flavio Prieto
Florian Kleber
Francesc Serratosa
Francesco Bianconi
Francesco Castro
Francesco Ponzio
Francisco Javier Hernández López
Frédéric Rayar
Furkan Osman Kar
Fushuo Huo
Fuxiao Liu
Fu-Zhao Ou
Gabriel Turinici
Gabrielle Flood
Gajjala Viswanatha Reddy
Gaku Nakano
Galal Binamakhashen
Ganesh Krishnasamy
Gang Pan
Gangyan Zeng
Gani Rahmon
Gaurav Harit
Gennaro Vessio
Genoveffa Tortora
George Azzopardi
Gerard Ortega
Gerardo E. Altamirano-Gomez
Gernot A. Fink
Gibran Benitez-Garcia
Gil Ben-Artzi
Gilbert Lim
Giorgia Minello
Giorgio Fumera
Giovanna Castellano
Giovanni Puglisi
Giulia Orrù
Giuliana Ramella
Gökçe Uludoğan
Gopi Ramena
Gorthi Rama Krishna Sai Subrahmanyam
Gourav Datta
Gowri Srinivasa
Gozde Sahin
Gregory Randall
Guanjie Huang
Guanjun Li
Guanwen Zhang
Guanyu Xu
Guanyu Yang
Guanzhou Ke
Guhnoo Yun
Guido Borghi
Guilherme Brandão Martins
Guillaume Caron
Guillaume Tochon
Guocai Du
Guohao Li
Guoqiang Zhong
Guorong Li
Guotao Li
Gurman Gill

Haechang Lee
Haichao Zhang
Haidong Xie
Haifeng Zhao
Haimei Zhao
Hainan Cui
Haixia Wang
Haiyan Guo
Hakime Ozturk
Hamid Kazemi
Han Gao
Hang Zou
Hanjia Lyu
Hanjoo Cho
Hanqing Zhao
Hanyuan Liu
Hanzhou Wu
Hao Li
Hao Meng
Hao Sun
Hao Wang
Hao Xing
Hao Zhao
Haoan Feng
Haodi Feng
Haofeng Li
Haoji Hu
Haojie Hao
Haojun Ai
Haopeng Zhang
Haoran Li
Haoran Wang
Haorui Ji
Haoxiang Ma
Haoyu Chen
Haoyue Shi
Harald Koestler
Harbinder Singh
Harris V. Georgiou
Hasan F. Ates
Hasan S. M. Al-Khaffaf
Hatef Otroshi Shahreza
Hebeizi Li
Heng Zhang
Hengli Wang
Hengyue Liu
Hertog Nugroho
Hieyong Jeong
Himadri Mukherjee
Hoai Ngo
Hoda Mohaghegh
Hong Liu
Hong Man
Hongcheng Wang
Hongjian Zhan
Hongxi Wei
Hongyu Hu
Hoseong Kim
Hossein Ebrahimnezhad
Hossein Malekmohamadi
Hrishav Bakul Barua
Hsueh-Yi Sean Lin
Hua Wei
Huafeng Li
Huali Xu
Huaming Chen
Huan Wang
Huang Chen
Huanran Chen
Hua-Wen Chang
Huawen Liu
Huayi Zhan
Hugo Jair Escalante
Hui Chen
Hui Li
Huichen Yang
Huiqiang Jiang
Huiyuan Yang
Huizi Yu
Hung T. Nguyen
Hyeongyu Kim
Hyeonjeong Park
Hyeonjun Lee
Hymalai Bello
Hyung-Gun Chi
Hyunsoo Kim
I-Chen Lin
Ik Hyun Lee
Ilan Shimshoni
Imad Eddine Toubal

Imran Sarker
Inderjot Singh Saggu
Indrani Mukherjee
Indranil Sur
Ines Rieger
Ioannis Pierros
Irina Rabaev
Ivan V. Medri
J. Rafid Siddiqui
Jacek Komorowski
Jacopo Bonato
Jacson Rodrigues Correia-Silva
Jaekoo Lee
Jaime Cardoso
Jakob Gawlikowski
Jakub Nalepa
James L. Wayman
Jan Čech
Jangho Lee
Jani Boutellier
Javier Gurrola-Ramos
Javier Lorenzo-Navarro
Jayasree Saha
Jean Lee
Jean Paul Barddal
Jean-Bernard Hayet
Jean-Philippe G. Tarel
Jean-Yves Ramel
Jenny Benois-Pineau
Jens Bayer
Jerin Geo James
Jesús Miguel García-Gorrostieta
Jia Qu
Jiahong Chen
Jiaji Wang
Jian Hou
Jian Liang
Jian Xu
Jian Zhu
Jianfeng Lu
Jianfeng Ren
Jiangfan Liu
Jianguo Wang
Jiangyan Yi
Jiangyong Duan
Jianhua Yang
Jianhua Zhang
Jianhui Chen
Jianjia Wang
Jianli Xiao
Jianqiang Xiao
Jianwu Wang
Jianxin Zhang
Jianxiong Gao
Jianxiong Zhou
Jianyu Wang
Jianzhong Wang
Jiaru Zhang
Jiashu Liao
Jiaxin Chen
Jiaxin Lu
Jiaxing Ye
Jiaxuan Chen
Jiaxuan Li
Jiayi He
Jiayin Lin
Jie Ou
Jiehua Zhang
Jiejie Zhao
Jignesh S. Bhatt
Jin Gao
Jin Hou
Jin Hu
Jin Shang
Jing Tian
Jing Yu Chen
Jingfeng Yao
Jinglun Feng
Jingtong Yue
Jingwei Guo
Jingwen Xu
Jingyuan Xia
Jingzhe Ma
Jinhong Wang
Jinjia Wang
Jinlai Zhang
Jinlong Fan
Jinming Su
Jinrong He
Jintao Huang

Jinwoo Ahn
Jinwoo Choi
Jinyang Liu
Jinyu Tian
Jionghao Lin
Jiuding Duan
Jiwei Shen
Jiyan Pan
Jiyoun Kim
João Papa
Johan Debayle
John Atanbori
John Wilson
John Zhang
Jónathan Heras
Joohi Chauhan
Jorge Calvo-Zaragoza
Jorge Figueroa
Jorma Laaksonen
José Joaquim De Moura Ramos
Jose Vicent
Joseph Damilola Akinyemi
Josiane Zerubia
Juan Wen
Judit Szücs
Juepeng Zheng
Juha Roning
Jumana H. Alsubhi
Jun Cheng
Jun Ni
Jun Wan
Junghyun Cho
Junjie Liang
Junjie Ye
Junlin Hu
Juntong Ni
Junxin Lu
Junxuan Li
Junyaup Kim
Junyeong Kim
Jürgen Seiler
Jushang Qiu
Juyang Weng
Jyostna Devi Bodapati
Jyoti Singh Kirar
Kai Jiang
Kaiqiang Song
Kalidas Yeturu
Kalle Åström
Kamalakar Vijay Thakare
Kang Gu
Kang Ma
Kanji Tanaka
Karthik Seemakurthy
Kaushik Roy
Kavisha Jayathunge
Kazuki Uehara
Ke Shi
Keigo Kimura
Keiji Yanai
Kelton A. P. Costa
Kenneth Camilleri
Kenny Davila
Ketan Atul Bapat
Ketan Kotwal
Kevin Desai
Keyu Long
Khadiga Mohamed Ali
Khakon Das
Khan Muhammad
Kilho Son
Kim-Ngan Nguyen
Kishan Kc
Kishor P. Upla
Klaas Dijkstra
Komal Bharti
Konstantinos Triaridis
Kostas Ioannidis
Koyel Ghosh
Kripabandhu Ghosh
Krishnendu Ghosh
Kshitij S. Jadhav
Kuan Yan
Kun Ding
Kun Xia
Kun Zeng
Kunal Banerjee
Kunal Biswas
Kunchi Li
Kurban Ubul

Lahiru N. Wijayasingha
Laines Schmalwasser
Lakshman Mahto
Lala Shakti Swarup Ray
Lale Akarun
Lan Yan
Lawrence Amadi
Lee Kang Il
Lei Fan
Lei Shi
Lei Wang
Leonardo Rossi
Lequan Lin
Levente Tamas
Li Bing
Li Li
Li Ma
Li Song
Lia Morra
Liang Xie
Liang Zhao
Lianwen Jin
Libing Zeng
Lidia Sánchez-González
Lidong Zeng
Lijun Li
Likang Wang
Lili Zhao
Lin Chen
Lin Huang
Linfei Wang
Ling Lo
Lingchen Meng
Lingheng Meng
Lingxiao Li
Lingzhong Fan
Liqi Yan
Liqiang Jing
Lisa Gutzeit
Liu Ziyi
Liushuai Shi
Liviu-Daniel Stefan
Liyuan Ma
Liyun Zhu
Lizuo Jin
Longteng Guo
Lorena Álvarez Rodríguez
Lorenzo Putzu
Lu Leng
Lu Pang
Lu Wang
Luan Pham
Luc Brun
Luca Guarnera
Luca Piano
Lucas Alexandre Ramos
Lucas Goncalves
Lucas M. Gago
Luigi Celona
Luis C. S. Afonso
Luis Gerardo De La Fraga
Luis S. Luevano
Luis Teixeira
Lunke Fei
M. Hassaballah
Maddimsetti Srinivas
Mahendran N.
Mahesh Mohan M. R.
Maiko Lie
Mainak Singha
Makoto Hirose
Malay Bhattacharyya
Mamadou Dian Bah
Man Yao
Manali J. Patel
Manav Prabhakar
Manikandan V. M.
Manish Bhatt
Manjunath Shantharamu
Manuel Curado
Manuel Günther
Manuel Marques
Marc A. Kastner
Marc Chaumont
Marc Cheong
Marc Lalonde
Marco Cotogni
Marcos C. Santana
Mario Molinara
Mariofanna Milanova

Markus Bauer
Marlon Becker
Mårten Wadenbäck
Martin G. Ljungqvist
Martin Kampel
Martina Pastorino
Marwan Torki
Masashi Nishiyama
Masayuki Tanaka
Massimo O. Spata
Matteo Ferrara
Matthew D. Dawkins
Matthew Gadd
Matthew S. Watson
Maura Pintor
Max Ehrlich
Maxim Popov
Mayukh Das
Md Baharul Islam
Md Sajid
Meghna Kapoor
Meghna P. Ayyar
Mei Wang
Meiqi Wu
Melissa L. Tijink
Meng Li
Meng Liu
Meng-Luen Wu
Mengnan Liu
Mengxi China Guo
Mengya Han
Michaël Clément
Michal Kawulok
Mickael Coustaty
Miguel Domingo
Milind G. Padalkar
Ming Liu
Ming Ma
Mingchen Feng
Mingde Yao
Minghao Li
Mingjie Sun
Ming-Kuang Daniel Wu
Mingle Xu
Mingyong Li
Mingyuan Jiu
Minh P. Nguyen
Minh Q. Tran
Minheng Ni
Minsu Kim
Minyi Zhao
Mirko Paolo Barbato
Mo Zhou
Modesto Castrillón-Santana
Mohamed Amine Mezghich
Mohamed Dahmane
Mohamed Elsharkawy
Mohamed Yousuf
Mohammad Hashemi
Mohammad Khalooei
Mohammad Khateri
Mohammad Mahdi Dehshibi
Mohammad Sadil Khan
Mohammed Mahmoud
Moises Diaz
Monalisha Mahapatra
Monidipa Das
Mostafa Kamali Tabrizi
Mridul Ghosh
Mrinal Kanti Bhowmik
Muchao Ye
Mugalodi Ramesha Rakesh
Muhammad Rameez Ur Rahman
Muhammad Suhaib Kanroo
Muming Zhao
Munender Varshney
Munsif Ali
Na Lv
Nader Karimi
Nagabhushan Somraj
Nakkwan Choi
Nakul Agarwal
Nan Pu
Nan Zhou
Nancy Mehta
Nand Kumar Yadav
Nandakishor Nandakishor
Nandyala Hemachandra
Nanfeng Jiang
Narayan Hegde

Narayan Ji Mishra
Narayan Vetrekar
Narendra D. Londhe
Nathalie Girard
Nati Ofir
Naval Kishore Mehta
Nazmul Shahadat
Neeti Narayan
Neha Bhargava
Nemanja Djuric
Newlin Shebiah R.
Ngo Ba Hung
Nhat-Tan Bui
Niaz Ahmad
Nick Theisen
Nicolas Passat
Nicolas Ragot
Nicolas Sidere
Nikolaos Mitianoudis
Nikolas Ebert
Nilah Ravi Nair
Nilesh A. Ahuja
Nilkanta Sahu
Nils Murrugarra-Llerena
Nina S. T. Hirata
Ninad Aithal
Ning Xu
Ningzhi Wang
Niraj Kumar
Nirmal S. Punjabi
Nisha Varghese
Norio Tagawa
Obaidullah Md Sk
Oguzhan Ulucan
Olfa Mechi
Oliver Tüselmann
Orazio Pontorno
Oriol Ramos Terrades
Osman Akin
Ouadi Beya
Ozge Mercanoglu Sincan
Pabitra Mitra
Padmanabha Reddy Y. C. A.
Palaash Agrawal
Palaiahnakote Shivakumara
Palash Ghosal
Pallav Dutta
Paolo Rota
Paramanand Chandramouli
Paria Mehrani
Parth Agrawal
Partha Basuchowdhuri
Patrick Horain
Pavan Kumar
Pavan Kumar Anasosalu Vasu
Pedro Castro
Peipei Li
Peipei Yang
Peisong Shen
Peiyu Li
Peng Li
Pengfei He
Pengrui Quan
Pengxin Zeng
Pengyu Yan
Peter Eisert
Petra Gomez-Krämer
Pierrick Bruneau
Ping Cao
Pingping Zhang
Pintu Kumar
Pooja Kumari
Pooja Sahani
Prabhu Prasad Dev
Pradeep Kumar
Pradeep Singh
Pranjal Sahu
Prasun Roy
Prateek Keserwani
Prateek Mittal
Praveen Kumar Chandaliya
Praveen Tirupattur
Pravin Nair
Preeti Gopal
Preety Singh
Prem Shanker Yadav
Prerana Mukherjee
Prerna A. Mishra
Prianka Dey
Priyanka Mudgal

Qc Kha Ng
Qi Li
Qi Ming
Qi Wang
Qi Zuo
Qian Li
Qiang Gan
Qiang He
Qiang Wu
Qiangqiang Zhou
Qianli Zhao
Qiansen Hong
Qiao Wang
Qidong Huang
Qihua Dong
Qin Yuke
Qing Guo
Qingbei Guo
Qingchao Zhang
Qingjie Liu
Qinhong Yang
Qiushi Shi
Qixiang Chen
Quan Gan
Quanlong Guan
Rachit Chhaya
Radu Tudor Ionescu
Rafal Zdunek
Raghavendra Ramachandra
Rahimul I. Mazumdar
Rahul Kumar Ray
Rajib Dutta
Rajib Ghosh
Rakesh Kumar
Rakesh Paul
Rama Chellappa
Rami O. Skaik
Ramon Aranda
Ran Wei
Ranga Raju Vatsavai
Ranganath Krishnan
Rasha Friji
Rashmi S.
Razaib Tariq
Rémi Giraud
René Schuster
Renlong Hang
Renrong Shao
Renu Sharma
Reza Sadeghian
Richard Zanibbi
Rimon Elias
Rishabh Shukla
Rita Delussu
Riya Verma
Robert J. Ravier
Robert Sablatnig
Robin Strand
Rocco Pietrini
Rocio Diaz Martin
Rocio Gonzalez-Diaz
Rohit Venkata Sai Dulam
Romain Giot
Romi Banerjee
Ru Wang
Ruben Machucho
Ruddy Théodose
Ruggero Pintus
Rui Deng
Rui P. Paiva
Rui Zhao
Ruifan Li
Ruigang Fu
Ruikun Li
Ruirui Li
Ruixiang Jiang
Ruowei Jiang
Rushi Lan
Rustam Zhumagambetov
S. Amutha
S. Divakar Bhat
Sagar Goyal
Sahar Siddiqui
Sahbi Bahroun
Sai Karthikeya Vemuri
Saibal Dutta
Saihui Hou
Sajad Ahmad Rather
Saksham Aggarwal
Sakthi U.

Salimeh Sekeh
Samar Bouazizi
Samia Boukir
Samir F. Harb
Samit Biswas
Samrat Mukhopadhyay
Samriddha Sanyal
Sandika Biswas
Sandip Purnapatra
Sanghyun Jo
Sangwoo Cho
Sanjay Kumar
Sankaran Iyer
Sanket Biswas
Santanu Roy
Santosh D. Pandure
Santosh Ku Behera
Santosh Nanabhau Palaskar
Santosh Prakash Chouhan
Sarah S. Alotaibi
Sasanka Katreddi
Sathyanarayanan N. Aakur
Saurabh Yadav
Sayan Rakshit
Scott McCloskey
Sebastian Bunda
Sejuti Rahman
Selim Aksoy
Sen Wang
Seraj A. Mostafa
Shanmuganathan Raman
Shao-Yuan Lo
Shaoyuan Xu
Sharia Arfin Tanim
Shehreen Azad
Sheng Wan
Shengdong Zhang
Shengwei Qin
Shenyuan Gao
Sherry X. Chen
Shibaprasad Sen
Shigeaki Namiki
Shiguang Liu
Shijie Ma
Shikun Li
Shinichiro Omachi
Shirley David
Shishir Shah
Shiv Ram Dubey
Shiva Baghel
Shivanand S. Gornale
Shogo Sato
Shotaro Miwa
Shreya Ghosh
Shreya Goyal
Shuai Su
Shuai Wang
Shuai Zheng
Shuaifeng Zhi
Shuang Qiu
Shuhei Tarashima
Shujing Lyu
Shuliang Wang
Shun Zhang
Shunming Li
Shunxin Wang
Shuping Zhao
Shuquan Ye
Shuwei Huo
Shuyue Lan
Shyi-Chyi Cheng
Si Chen
Siddarth Ravichandran
Sihan Chen
Siladittya Manna
Silambarasan Elkana Ebinazer
Simon Benaïchouche
Simon S. Woo
Simone Caldarella
Simone Milani
Simone Zini
Sina Lotfian
Sitao Luan
Sivaselvan B.
Siwei Li
Siwei Wang
Siwen Luo
Siyu Chen
Sk Aziz Ali
Sk Md Obaidullah

Sneha Shukla
Snehasis Banerjee
Snehasis Mukherjee
Snigdha Sen
Sofia Casarin
Soheila Farokhi
Soma Bandyopadhyay
Son Minh Nguyen
Son Xuan Ha
Sonal Kumar
Sonam Gupta
Sonam Nahar
Song Ouyang
Sotiris Kotsiantis
Souhaila Djaffal
Soumen Biswas
Soumen Sinha
Soumitri Chattopadhyay
Souvik Sengupta
Spiros Kostopoulos
Sreeraj Ramachandran
Sreya Banerjee
Srikanta Pal
Srinivas Arukonda
Stephane A. Guinard
Su O. Ruan
Subhadip Basu
Subhajit Paul
Subhankar Ghosh
Subhankar Mishra
Subhankar Roy
Subhash Chandra Pal
Subhayu Ghosh
Sudip Das
Sudipta Banerjee
Suhas Pillai
Sujit Das
Sukalpa Chanda
Sukhendu Das
Suklav Ghosh
Suman K. Ghosh
Suman Samui
Sumit Mishra
Sungho Suh
Sunny Gupta
Suraj Kumar Pandey
Surendrabikram Thapa
Suresh Sundaram
Sushil Bhattacharjee
Susmita Ghosh
Swakkhar Shatabda
Syed Ms Islam
Syed Tousiful Haque
Taegyeong Lee
Taihui Li
Takashi Shibata
Takeshi Oishi
Talha Ahmad Siddiqui
Tanguy Gernot
Tangwen Qian
Tanima Bhowmik
Tanpia Tasnim
Tao Dai
Tao Hu
Tao Sun
Taoran Yi
Tapan Shah
Taveena Lotey
Teng Huang
Tengqi Ye
Teresa Alarcon
Tetsuji Ogawa
Thanh Phuong Nguyen
Thanh Tuan Nguyen
Thattapon Surasak
Thibault Napolãon
Thierry Bouwmans
Thinh Truong Huynh Nguyen
Thomas De Min
Thomas E. K. Zielke
Thomas Swearingen
Tianatahina Jimmy Francky Randrianasoa
Tianheng Cheng
Tianjiao He
Tianyi Wei
Tianyuan Zhang
Tianyue Zheng
Tiecheng Song
Tilottama Goswami
Tim Büchner

Tim H. Langer
Tim Raven
Tingkai Liu
Tingting Yao
Tobias Meisen
Toby P. Breckon
Tong Chen
Tonghua Su
Tran Tuan Anh
Tri-Cong Pham
Trishna Saikia
Trung Quang Truong
Tuan T. Nguyen
Tuan Vo Van
Tushar Shinde
Ujjwal Karn
Ukrit Watchareeruetai
Uma Mudenagudi
Umarani Jayaraman
V. S. Malemath
Vallidevi Krishnamurthy
Ved Prakash
Venkata Krishna Kishore Kolli
Venkata R. Vavilthota
Venkatesh Thirugnana Sambandham
Verónica Maria Vasconcelos
Véronique Ve Eglin
Víctor E. Alonso-Pérez
Vinay Palakkode
Vinayak S. Nageli
Vincent J. Whannou De Dravo
Vincenzo Conti
Vincenzo Gattulli
Vineet Padmanabhan
Vishakha Pareek
Viswanath Gopalakrishnan
Vivek Singh Baghel
Vivekraj K.
Vladimir V. Arlazarov
Vu-Hoang Tran
W. Sylvia Lilly Jebarani
Wachirawit Ponghiran
Wafa Khlif
Wang An-Zhi
Wanli Xue
Wataru Ohyama
Wee Kheng Leow
Wei Chen
Wei Cheng
Wei Hua
Wei Lu
Wei Pan
Wei Tian
Wei Wang
Wei Wei
Wei Zhou
Weidi Liu
Weidong Yang
Weijun Tan
Weimin Lyu
Weinan Guan
Weining Wang
Weiqiang Wang
Weiwei Guo
Weixia Zhang
Wei-Xuan Bao
Weizhong Jiang
Wen Xie
Wenbin Qian
Wenbin Tian
Wenbin Wang
Wenbo Zheng
Wenhan Luo
Wenhao Wang
Wen-Hung Liao
Wenjie Li
Wenkui Yang
Wenwen Si
Wenwen Yu
Wenwen Zhang
Wenwu Yang
Wenxi Li
Wenxi Yue
Wenxue Cui
Wenzhuo Liu
Widhiyo Sudiyono
Willem Dijkstra
Wolfgang Fuhl
Xi Zhang
Xia Yuan

Xianda Zhang
Xiang Zhang
Xiangdong Su
Xiang-Ru Yu
Xiangtai Li
Xiangyu Xu
Xiao Guo
Xiao Hu
Xiao Wu
Xiao Yang
Xiaofeng Zhang
Xiaogang Du
Xiaoguang Zhao
Xiaoheng Jiang
Xiaohong Zhang
Xiaohua Huang
Xiaohua Li
Xiao-Hui Li
Xiaolong Sun
Xiaosong Li
Xiaotian Li
Xiaoting Wu
Xiaotong Luo
Xiaoyan Li
Xiaoyang Kang
Xiaoyi Dong
Xin Guo
Xin Lin
Xin Ma
Xinchi Zhou
Xingguang Zhang
Xingjian Leng
Xingpeng Zhang
Xingzheng Lyu
Xinjian Huang
Xinqi Fan
Xinqi Liu
Xinqiao Zhang
Xinrui Cui
Xizhan Gao
Xu Cao
Xu Ouyang
Xu Zhao
Xuan Shen
Xuan Zhou
Xuchen Li
Xuejing Lei
Xuelu Feng
Xueting Liu
Xuewei Li
Xueyi X. Wang
Xugong Qin
Xu-Qian Fan
Xuxu Liu
Xu-Yao Zhang
Yan Huang
Yan Li
Yan Wang
Yan Xia
Yan Zhuang
Yanan Li
Yanan Zhang
Yang Hou
Yang Jiao
Yang Liping
Yang Liu
Yang Qian
Yang Yang
Yang Zhao
Yangbin Chen
Yangfan Zhou
Yanhui Guo
Yanjia Huang
Yanjun Zhu
Yanming Zhang
Yanqing Shen
Yaoming Cai
Yaoxin Zhuo
Yaoyan Zheng
Yaping Zhang
Yaqian Liang
Yarong Feng
Yasmina Benmabrouk
Yasufumi Sakai
Yasutomo Kawanishi
Yazeed Alzahrani
Ye Du
Ye Duan
Yechao Zhang
Yeong-Jun Cho

Yi Huo
Yi Shi
Yi Yu
Yi Zhang
Yibo Liu
Yibo Wang
Yi-Chieh Wu
Yifan Chen
Yifei Huang
Yihao Ding
Yijie Tang
Yikun Bai
Yimin Wen
Yinan Yang
Yin-Dong Zheng
Yinfeng Yu
Ying Dai
Yingbo Li
Yiqiao Li
Yiqing Huang
Yisheng Lv
Yisong Xiao
Yite Wang
Yizhe Li
Yong Wang
Yonghao Dong
Yong-Hyuk Moon
Yongjie Li
Yongqian Li
Yongqiang Mao
Yongxu Liu
Yongyu Wang
Yongzhi Li
Youngha Hwang
Yousri Kessentini
Yu Wang
Yu Zhou
Yuan Tian
Yuan Zhang
Yuanbo Wen
Yuanxin Wang
Yubin Hu
Yubo Huang
Yuchen Ren
Yucheng Xing
Yuchong Yao
Yuecong Min
Yuewei Yang
Yufei Zhang
Yufeng Yin
Yugen Yi
Yuhang Ming
Yujia Zhang
Yujun Ma
Yukiko Kenmochi
Yun Hoyeoung
Yun Liu
Yunhe Feng
Yunxiao Shi
Yuru Wang
Yushun Tang
Yusuf Osmanlioglu
Yusuke Fujita
Yuta Nakashima
Yuwei Yang
Yuwu Lu
Yuxi Liu
Yuya Obinata
Yuyao Yan
Yuzhi Guo
Zaipeng Xie
Zander W. Blasingame
Zedong Wang
Zeliang Zhang
Zexin Ji
Zhanxiang Feng
Zhaofei Yu
Zhe Chen
Zhe Cui
Zhe Liu
Zhe Wang
Zhekun Luo
Zhen Yang
Zhenbo Li
Zhenchun Lei
Zhenfei Zhang
Zheng Liu
Zheng Wang
Zhengming Yu
Zhengyin Du

Zhengyun Cheng
Zhenshen Qu
Zhenwei Shi
Zhenzhong Kuang
Zhi Cai
Zhi Chen
Zhibo Chu
Zhicun Yin
Zhida Huang
Zhida Zhang
Zhifan Gao
Zhihang Ren
Zhihang Yuan
Zhihao Wang
Zhihua Xie
Zhihui Wang
Zhikang Zhang
Zhiming Zou
Zhiqi Shao
Zhiwei Dong
Zhiwei Qi
Zhixiang Wang
Zhixuan Li
Zhiyu Jiang
Zhiyuan Yan
Zhiyuan Yu
Zhiyuan Zhang
Zhong Chen
Zhongwei Teng
Zhongzhan Huang
Zhongzhi Yu
Zhuan Han
Zhuangzhuang Chen
Zhuo Liu
Zhuo Su
Zhuojun Zou
Zhuoyue Wang
Ziang Song
Zicheng Zhang
Zied Mnasri
Zifan Chen
Žiga Babnik
Zijing Chen
Zikai Zhang
Ziling Huang
Zilong Du
Ziqi Cai
Ziqi Zhou
Zi-Rui Wang
Zirui Zhou
Ziwen He
Ziyao Zeng
Ziyi Zhang
Ziyue Xiang
Zonglei Jing
Zongyi Xu

Contents – Part I

Semi-Supervised Variational Adversarial Active Learning via Learning to Rank and Agreement-Based Pseudo Labeling

Zongyao Lyu and William J. Beksi(✉)

The University of Texas at Arlington, Arlington, TX 76019, USA
william.beksi@uta.edu

Abstract. Active learning aims to alleviate the amount of labor involved in data labeling by automating the selection of unlabeled samples via an acquisition function. For example, variational adversarial active learning (VAAL) leverages an adversarial network to discriminate unlabeled samples from labeled ones using latent space information. However, VAAL has the following shortcomings: (i) it does not exploit target task information, and (ii) unlabeled data is only used for sample selection rather than model training. To address these limitations, we introduce novel techniques that significantly improve the use of abundant unlabeled data during training and take into account the task information. Concretely, we propose an improved pseudo-labeling algorithm that leverages information from all unlabeled data in a semi-supervised manner, thus allowing a model to explore a richer data space. In addition, we develop a ranking-based loss prediction module that converts predicted relative ranking information into a differentiable ranking loss. This loss can be embedded as a rank variable into the latent space of a variational autoencoder and then trained with a discriminator in an adversarial fashion for sample selection. We demonstrate the superior performance of our approach over the state of the art on various image classification and segmentation benchmark datasets.

Keywords: Active Learning · Semi-Supervised Learning · Image Classification and Segmentation

1 Introduction

Deep learning has shown impressive results on computer vision tasks mainly due to annotated large-scale datasets. Yet, acquiring labeled data can be extremely costly or even infeasible. To overcome this issue, active learning (AL) was introduced [6,31]. In AL, a model is initialized with a relatively small set of labeled training samples. Then, an AL algorithm progressively chooses samples for annotation that

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_1.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 1–16, 2025.
https://doi.org/10.1007/978-3-031-78107-0_1

yield high classification performance while minimizing labeling costs. By demonstrating a reduced requirement for training instances, AL has been applied to various computer vision applications including image categorization, image segmentation, text classification, and more.

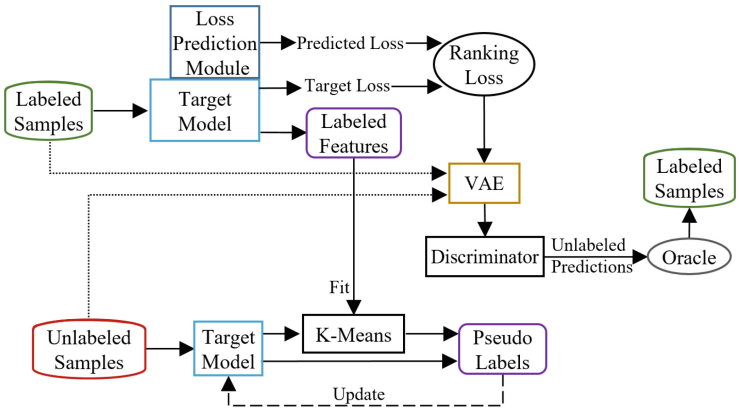


Fig. 1. An overview of SS-VAAL. First, a loss prediction module attached to the target model predicts losses on the input data. Next, the predicted losses along with the actual target losses are transformed into ranking losses via a pretrained ranking function. Unlabeled samples are then passed to the target model and subsequently through a k-means algorithm to acquire pseudo labels for additional training. Finally, a discriminator following a variational autoencoder is trained in an adversarial manner to select unlabeled samples for annotation.

Among the most prevalent AL strategies, pool-based approaches have access to a huge supply of unlabeled data. This provides valuable information about the underlying structure of the whole data distribution, especially for small labeling budgets. Nevertheless, many AL methods still fail to leverage valuable information within the unlabeled data during training. On the other hand, semi-supervised learning (SSL), in particular the technique of pseudo labeling, thrives on utilizing unlabeled data. Pseudo labeling is based on the concept whereby a model assigns "pseudo labels" to samples that produce high-confidence scores. It then integrates these samples into the training process. In contrast, AL typically selects only a handful of highly-informative samples (i.e., samples with low prediction confidence) at each learning step and regularly seeks user input. Although AL and pseudo labeling both aim to leverage a model's uncertainty, they look at different ends of the same spectrum. Hence, their combination can be expected to achieve increased performance [14].

In light of this observation, we propose to exploit both labeled and unlabeled data during model training by (i) predicting pseudo labels for unlabeled samples, and (ii) incorporating these samples and their pseudo labels into the labeled training data in every AL cycle. The idea of using unlabeled data for training is not new. Earlier work by Wang et al. [36] showed promising results by applying entropy-based pseudo labeling to

AL. However, pseudo labeling can perform poorly in its original formulation. The subpar performance is attributed to inaccurate high-confidence predictions made by poorly calibrated models. These predictions produce numerous incorrect pseudo labels [1]. To tackle this issue, we introduce a novel agreement-based clustering technique that assists in determining pseudo labels. Clustering algorithms can analyze enormous amounts of unlabeled data in an unsupervised way [7,26], and cluster centers are highly useful for querying labels from an oracle [18]. Our two-step process involves (i) separately clustering labeled and unlabeled data, (ii) assigning each piece of unlabeled data an initial pseudo label and a clustering label. A final pseudo label is confirmed only if these two labels agree. The end result is a significant reduction in the number of incorrect pseudo labels.

The second aspect of our work focuses on the sample selection strategy in AL. We base our approach on the VAAL [34] framework. VAAL uses an adversarial discriminator to discern between labeled and unlabeled data, which informs the sample selection process. Later adaptations of VAAL (e.g., TA-VAAL [19]) incorporate a loss prediction module, relaxing the task of exact loss prediction to loss ranking prediction. Additionally, a ranking conditional generative adversarial network (RankCGAN) [29] is employed to combine normalized ranking loss information into VAAL. To better integrate task-related information into the training process, we propose a learning-to-rank method for VAAL. This decision is inspired by the realization that the loss prediction can be interpreted as a ranking problem [23], a concept central to information retrieval. We refine the loss prediction process by applying a contemporary learning-to-rank technique for approximating non-differentiable operations in ranking-based scores. The loss prediction module estimates a loss for labeled input, converting the predicted loss and actual target loss into a differentiable ranking loss. This ranking loss, along with labeled and unlabeled data, is provided as input into an adversarial learning process that identifies unlabeled samples for annotation. Therefore, by explicitly exploiting the loss information directly related to the given task, task-related information is integrated into the AL process. The architecture of our proposed method, SS-VAAL, is depicted in Fig. 1.

To summarize, our contributions are the following.

1. We create a novel agreement-based pseudo-labeling technique that optimally harnesses rich information from abundant unlabeled data in each AL cycle, capitalizing on the advantages of unsupervised feature learning.
2. We devise an enhanced loss prediction module that employs a learning-to-rank method, yielding a more effective sample selection strategy. We develop a ranking method that explicitly ranks the predicted losses by taking into account the entire list of loss structures, as opposed to only the pairwise information considered in prior approaches.
3. We highlight the superior efficacy of our approach through its application to common image classification and segmentation benchmarks.

Our source code is publicly available [35].

2 Related Work

2.1 Active Learning

AL methods operate on an iterative principle of constructing a training set. This involves (i) cyclically training the classifier on the current labeled training set, and (ii) once the model converges, soliciting an oracle (e.g., human annotator) to label new points selected from a pool of unlabeled data based on the utilized heuristic. This type of AL belongs to pool-based AL, in which our methodology lies. Pool-based AL can be classified into three groups: (i) uncertainty (informativeness-based) methods [3,12,22], (ii) diversity (representativeness-based) methods [30], and (iii) hybrid methods [2,17, 38,39] based on the instance selection strategy they use. Among the various instance selection strategies, uncertainty-based selection is the most prevalent. It measures the uncertainties of new unlabeled samples using the predictions made by prior classifiers.

Diversity-based AL methods rely on selecting a few examples by increasing the diversity of a given batch. The core-set technique [30] was proposed to minimize the distance between the labeled and unlabeled data pool using the intermediate feature information of a convolutional deep neural network (DNN) model. It was shown to be an effective method for large-scale image classification tasks and was theoretically proven to work best when the number of classes is small. However, as the number of classes grows the performance deteriorates.

AL methods that combine uncertainty and diversity use a two-step process to select high-uncertainty points as the most informative points in a batch. Li et al. [24] presented an adaptive AL approach that combines an information density and uncertainty measure together to label critical instances for image classification. Sinha et al. [34] observed that the uncertainty-based batch query strategy often results in a lack of sample diversity and is vulnerable to outliers. As a remedy, they proposed VAAL, a method that utilizes an adversarial learning approach to distinguish the spatial coding features of labeled and unlabeled data, thereby mitigating outlier interference. It also employs both labeled and unlabeled data to jointly train a variational autoencoder (VAE) in a semi-supervised fashion. Sample selection in VAAL is based on the prediction from the discriminator adversarially trained with the VAE. While VAAL incorporates unlabeled data during the adversarial learning process, it neglects this data during target task learning due to its inherently task-agnostic nature. An extended version of VAAL [42] was proposed to combine task-aware and task-agnostic approaches with an uncertainty indicator and a unified representation for both labeled and unlabeled data.

Task-aware VAAL (TA-VAAL) [19] is an alternative extension of task-agnostic VAAL that combines task-aware and task-agnostic approaches. TA-VAAL adapts VAAL to consider the data distribution of both labeled and unlabeled pools by combining them with a learning loss approach [40]. The learning loss is a task-agnostic method. It includes a loss prediction module that learns to predict the target loss of unlabeled data and selects data with the highest predicted loss for labeling. TA-VAAL relaxes the task of learning loss prediction to ranking loss prediction and employs RankCGAN to incorporate normalized ranking loss information into VAAL. However, the main difference between VAAL and TA-VAAL is the use of task-related information for learning the ranking function in conjunction with information from unlabeled

data. Even so, unlabeled data is not directly applied to target task learning. *To rectify this, we propose a novel pseudo-labeling technique that can be integrated into each AL cycle, enabling the comprehensive utilization of the rich information contained within unlabeled data for direct learning of the target task.* Another recent method, multi-classifier adversarial optimization for active learning (MAOAL) [13], employs multiple classifiers trained adversarially to more precisely define inter-class decision boundaries while aligning feature distributions between labeled and unlabeled data. We demonstrate that our method outperforms MAOAL in image classification tasks.

2.2 Semi-Supervised Learning

SSL is a strategy that leverages both labeled and unlabeled data for model training, with an emphasis on utilizing abundantly available unlabeled data. Several techniques have been proposed to exploit the relationship between labeled and unlabeled data to achieve better performance. A notable technique is pseudo labeling [21] where a model, once trained, is used to predict labels for unlabeled data. These pseudo-labeled data are then used in subsequent training iterations. Other methods, such as multi-view training [33] and consistency regularization [28], leverage the structure or inherent properties of the data to derive meaningful information from the unlabeled portion.

Several efforts have been made to combine SSL and AL methods to make better use of the unlabeled data during training [4,32,36]. A common strategy in this integrated approach is to apply pseudo labeling techniques during each AL cycle. This enriches the training set and improves model accuracy by combining SSL's efficient use of unlabeled data with AL's selective querying, offering a cost-effective solution for scenarios with limited labeled data. Although simple to implement, pseudo labeling can perform relatively poorly in its original formulation. The underperformance of pseudo labeling is generally attributed to incorrect high-confidence predictions from models that are not properly calibrated. This causes a proliferation of wrong pseudo labels, thus resulting in a noisy training process [27]. *Our enhanced pseudo-labeling approach addresses this limitation by incorporating unsupervised feature learning through the use of clustering.* Clustering algorithms are employed to group the unlabeled data, and the cluster centers are used for verifying the predicted pseudo labels. This greatly reduces the number of incorrect pseudo labels as the labels are assigned based on the proximity to cluster centers, which represents the classes better than individual instances.

3 Method

Let (X_L, Y_L) be a pool of data and their labels, and X_U the pool of unlabeled data. Training starts with K available labeled sample pairs (X_L^K, Y_L^K). Given a fixed labeling budget in each AL cycle, b samples from the unlabeled pool are queried according to an acquisition function. Next, the samples are annotated by human experts and added to the labeled pool. The model is then iteratively trained on the updated labeled pool (X_L^{K+b}, Y_L^{K+b}), and this process is repeated until the labeling budget is exhausted.

SS-VAAL enhances the VAAL framework and its variant, TA-VAAL, as follows. VAAL employs adversarial learning to distinguish features of labeled and unlabeled data, which reduces outlier impact and leverages both labeled and unlabeled

data in a semi-supervised training scheme. TA-VAAL, building on the groundwork of VAAL, utilizes global data structures and local task-related information for sample queries. Our methodology improves upon these predecessors by harnessing the full potential of the data distribution and model uncertainty, hence further refining the query strategy in the AL process.

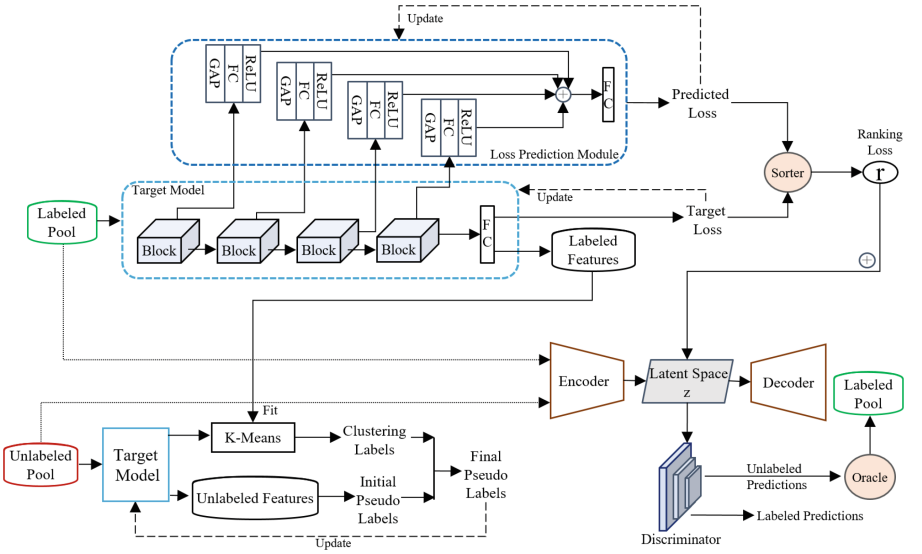


Fig. 2. The detailed architecture of SS-VAAL. (Stage 1) A loss prediction module is attached to the target model to predict losses on the input data. These predicted losses, along with the actual losses obtained from the target model, are transformed into ranking losses via a pretrained ranking function. Features of the labeled samples are extracted from the target model to fit a k-means algorithm. (Stage 2) Unlabeled samples are processed through the target model to obtain initial pseudo labels. The k-means algorithm, already fit with labeled features, is also applied to the unlabeled samples to obtain clustering labels for them. Initial pseudo and clustering labels are combined to determine the final pseudo labels. These unlabeled samples and their pseudo labels are then used for additional training of the target model. (Stage 3) Both labeled and unlabeled samples are fed into an encoder network to learn the latent variables. The learned and rank variables are trained adversarially with a discriminator. Sample selection is based on the predicted probability from the discriminator.

3.1 Clustering-Assisted Pseudo Labeling

Both VAAL and TA-VAAL do not fully use unlabeled data in the target learning task. Therefore, we propose to exploit both types of data during model training as follows. Given X_L and X_U for labeled and unlabeled examples, respectively, we apply a classifier f on the unlabeled data $f(X_U)$, and select and assign pseudo labels $\hat{y}$ for the most certain predictions. Traditionally, the labeled set will be directly augmented by

$y = y + \hat{y}$ for the next round of training. Nonetheless, pseudo labeling in its initial form may produce high-confidence predictions that are incorrect, resulting in numerous erroneous pseudo labels and ultimately causing an unstable training process.

To mitigate this issue, we present a semi-supervised pre-clustering technique for each pseudo label selection process that enhances robustness by reducing incorrect pseudo labels. In each AL cycle, we first train a model on the available labeled data. We modify the network to output both the probability score and the feature vector from the last fully-connected layer before sending it to the softmax function. Then, we fit a k-means clustering algorithm on the output features of the labeled training data. This allows the algorithm to learn the structure of the labeled data and predict clusters each of whose centroid corresponds to one of the classes of the dataset. One thing to note is that the cluster assignments won't necessarily correspond directly to the classes of the dataset being trained. This is because clustering algorithms (e.g., k-means) do not have any inherent knowledge of class labels and thus the cluster labels they assign have no intrinsic meaning. To be meaningful, we map the clustering labels to the actual classes to ensure that they correspond to each other. This is done by assigning each cluster label to the most frequent true class label within that cluster based on the labeled training data.

Next, we train a classifier on all unlabeled data to get the predicted probability vectors

$$\sum^{U} \mathbf{p}(y_i = j \,|\, \mathbf{x}_i) = f(X_U) \rightarrow \mathcal{R}^c, \tag{1}$$

where c is the number of total classes. We assign initial pseudo labels to the unlabeled data with the most certain predictions only when their associated probabilities are larger than a threshold τ (we set τ = 0.95 in the experiments), i.e.,

$$\begin{aligned} j^* &= \max_j \mathbf{p}(y_i = j \,|\, \mathbf{x}_i), \\ \hat{y}_i &= \begin{cases} \arg j, & j > \tau \\ 0, & otherwise. \end{cases} \end{aligned} \tag{2}$$

Then, we apply the k-means function learned on the labeled data to the unlabeled data to predict the clusters they belong to. Each unlabeled sample is grouped to the nearest cluster and assigned a label to which the cluster centroid corresponds.

Each unlabeled data point will now have both an initial pseudo label and a clustering label. Lastly, we compare the temporary pseudo labels with the clustering labels to determine a final pseudo label for each unlabeled data only if they agree with each other. By doing so, we reduce the number of incorrect pseudo labels, thus taking full advantage of the abundant unlabeled data for model training. Stage 2 in Fig. 2 shows this agreement-based pseudo-labeling process. We demonstrate improvement over conventional pseudo labeling through an ablation study in the supplementary material.

3.2 Loss Prediction with Learning-to-Rank

In LL4AL [40], Yoo and Kweon designed a loss prediction module attached to the target network and jointly learned to predict the losses of unlabeled inputs. The loss is

predicted as a measure of uncertainty, directly guiding the sample selection process. LL4AL has proven to be effective, yet the "loss-prediction loss" that is key to this approach is not trivial to calculate. The loss module adapts roughly to the scale changes of the loss instead of fitting to the exact value. Similar to TA-VAAL, we incorporate task-related information into the learning process by combining VAAL with the loss prediction module. *Unlike TA-VAAL, which employs a GAN-based ranking method to address this issue, our approach integrates VAAL with a listwise learning-to-rank technique that explicitly ranks the predicted losses thus taking into account the entire list of loss structures.* This decision stems from the observation that learning the loss prediction can be seen as a ranking problem. Additionally, the loss in TA-VAAL resembles the original LL4AL as both only consider the neighboring data pairs and ignore the overall list structure. This motivates us to use a more appropriate listwise ranking scheme. Ranking is crucial for many computing tasks, such as information retrieval, and it is often addressed via a listwise approach (e.g., [5,25]). This involves taking ranked lists of objects as instances and training a ranking function through the minimization of a listwise loss function defined on the predicted and ground-truth lists [37].

SoDeep [10] is a method for approximating the non-differentiable sorting operation in ranking-based losses. It uses a DNN as a sorter to approximate the ranking function and it is pretrained separately on synthetic values and their ground-truth ranks. The trained sorter can then be applied directly in downstream tasks by combining it with an existing model (e.g., the loss prediction module) and converting the value list given by the model into a ranking list. The ranking loss between the predicted and ground-truth ranks can then be calculated and backpropagated through the differentiable sorter and used to update the weights of the model. Figure 3 illustrates the sorter architecture. We find this process works well with the loss prediction task in the loss module. *Therefore, we apply SoDeep to the loss prediction module and learn to predict the ranking loss as a variable that injects task-related information into the subsequent adversarial learning process, which increases the robustness of the unlabeled sample selection.* Concretely, we substitute the loss prediction module into the sorter architecture as the DNN target model to produce the predicted scores where the target losses are used as the ground-truth scores.

The upper-right side of Fig. 2 displays the architecture of the modified loss learning process. We retain the basic structure of the original loss prediction module. Given an input, the target model generates a prediction, while the loss prediction module takes multi-layer features as inputs that are extracted from multiple mid-level blocks of the target model. These features are connected to multiple identical blocks each of which consists of a global average pooling layer and a fully-connected layer. Then, the outputs are concatenated and passed through another fully-connected layer to be converted to a scalar value as the predicted loss $\mathcal{L}_{pred}$. The target prediction and annotation are used to calculate a loss $\mathcal{L}_{target}$, which assists in training the target model. This target loss is treated as the ground-truth loss for the loss prediction module and used to compute the loss-prediction loss. Specifically, the predicted loss and the target loss are passed through the pretrained SoDeep sorter and converted to a differentiable ranking loss

$$\mathcal{L}_{ranking} = SoDeep(\mathcal{L}_{pred}, \mathcal{L}_{target}), \tag{3}$$

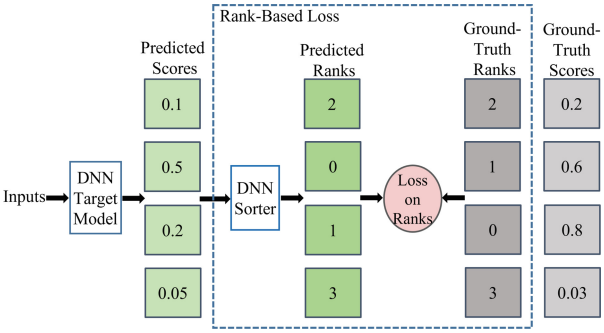


Fig. 3. An overview of the SoDeep sorter architecture. A pretrained differentiable DNN sorter converts the raw scores into ranks given by the target model. A loss is then applied to the predicted rank, backpropagated through the differentiable sorter, and used to update the weights.

which can be used to update the weights of the model. The objective function of the task learner with the ranking loss module is

$$\mathcal{L} = \mathcal{L}_{target}(\hat{y}_L, y_L) + \lambda \mathcal{L}_{ranking}, \tag{4}$$

where $\hat{y}_L$ and y_L are the predicted and ground-truth labels, respectively, and λ is a scaling constant. This training process is illustrated as Stage 1 in Fig. 2. The learned ranking loss is embedded as a task-related rank variable in the latent space of a VAE for the subsequent adversarial learning process, which is described in detail in Sect. 3.3. Stage 1 of the two-stage training is summarized in Algorithm 1.

Algorithm 1. Target Model Training

Require: Labeled data pool (X_L, Y_L), unlabeled data pool X_U, pretrained SoDeep sorter S, initialized model θ_T, training epochs N, threshold τ

Ensure:

for $i = 1$ to N **do**
  Train target model θ_T on labeled data (X_L, Y_L) to obtain features and target loss $\mathcal{L}_{target}$
  Obtain predicted loss $\mathcal{L}_{pred}$ through loss prediction module by fusing multi-level features
  $\mathcal{L}_{ranking} \leftarrow S(\mathcal{L}_{target}, \mathcal{L}_{pred})$
  Fit features to k-means algorithm
  Apply k-means on unlabeled data X_U and predict clustering labels (CL)
  Predict initial pseudo labels (IPL) $\hat{y}_i$ for unlabeled data X_U using (2)
  Final pseudo labels $\leftarrow$ IPL $\cap$ CL
  Train model on labeled and pseudo labeled data
end for
return Trained model θ_T

3.3 Joint Training with a Variational Autoencoder and Discriminator

For sample selection, we extend VAAL by utilizing a VAE and an adversarial network (discriminator) to distinguish labeled from unlabeled data. *Unlike VAAL, which only considers the data distribution for adversarial learning, we incorporate task-related information by embedding the ranking loss as a rank variable in the latent space for training both the VAE and the discriminator.* Let p_θ and q_ϕ be the encoder and decoder parameterized by θ and ϕ, $\mathbf{z}_L$ and $\mathbf{z}_U$ the latent variables generated from the encoder for labeled and unlabeled data, and r_L the rank variable for the labeled data. Let $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$ be the unit Gaussian prior. The transductive learning of the VAE to capture latent representation information on both labeled and unlabeled data is characterized by

$$\begin{aligned}\mathcal{L}_{VAE}^{trans} = \mathbb{E}[\log q_\phi(\mathbf{x}_L \mid \mathbf{z}_L, r_L)] - \beta KL(p_\theta(\mathbf{z}_L \mid \mathbf{x}_L)||p(\mathbf{z})) \\ + \mathbb{E}[\log q_\phi(\mathbf{x}_U \mid \mathbf{z}_U, \hat{l}_U)] - \beta KL(p_\theta(\mathbf{z}_U \mid \mathbf{x}_U)||p(\mathbf{z})),\end{aligned} \tag{5}$$

where $\hat{l}_U$ is the predicted loss $\mathcal{L}_{pred}$ over unlabeled data, β is the Lagrangian parameter, and $\mathbb{E}$ denotes the expectation [16].

With the latent representations $\mathbf{z}_L$ and $\mathbf{z}_U$ learned by the VAE of both the labeled and unlabeled data, the objective function of the VAE in adversarial training is then

$$\mathcal{L}_{VAE}^{adv} = -\mathbb{E}[\log(D(p_\theta(\mathbf{z}_L \mid \mathbf{x}_L, r_L)))] - \mathbb{E}[\log(D(p_\theta(\mathbf{z}_U \mid \mathbf{x}_U, \hat{l}_U)))]. \tag{6}$$

Combining (5) and (6), the overall objective function of the VAE is

$$\mathcal{L}_{VAE} = \mathcal{L}_{VAE}^{trans} + \eta \mathcal{L}_{VAE}^{adv}, \tag{7}$$

where η is a coefficient hyperparameter. The objective function of the discriminator D during adversarial training is

$$\mathcal{L}_{D}^{adv} = -\mathbb{E}[\log(D(p_\theta(\mathbf{z}_L \mid \mathbf{x}_L, r_L)))] - \mathbb{E}[\log(1 - D(p_\theta(\mathbf{z}_U \mid \mathbf{x}_U, \hat{l}_U)))], \tag{8}$$

and the overall objective function of the adversarial training is

$$\min_{p_\theta} \max_{D} \mathbb{E}[\log(D(p_\theta(\mathbf{z}_L \mid \mathbf{x}_L, r_L)))] + \mathbb{E}[\log(1 - D(p_\theta(\mathbf{z}_U \mid \mathbf{x}_U, \hat{l}_U)))]. \tag{9}$$

The VAE and discriminator are trained in an adversarial manner. Specifically, the VAE maps the labeled $p_\theta(\mathbf{z}_L \mid \mathbf{x}_L)$ and unlabeled $p_\theta(\mathbf{z}_U \mid \mathbf{x}_U)$ data into the latent space with binary labels 1 and 0, respectively, and tries to trick the discriminator into classifying all the inputs as labeled. On the other hand, the discriminator tries to distinguish the unlabeled data from the labeled data by predicting the probability of each sample being from the labeled pool. Thus, the adversarial network is trained to serve as the sampling scheme via the discriminator by predicting the samples associated with the latent representations of z_L and z_U to be from the labeled pool x_L or the unlabeled pool x_U according to its predicted probability $D(\cdot)$. In short, sample selection is based on the predicted probability of the discriminator adversarially trained with the VAE. The smaller the probability, the more likely the sample will be selected for annotating. This adversarial training process is shown as Stage 3 in Fig. 2 and summarized in Algorithm 2.

Algorithm 2. Adversarial Training and Sample Selection

Require: Labeled data (X_L, Y_L), unlabeled data X_U, rank variable (i.e., ranking loss) r_L, trained model θ_T, initialized models θ_{VAE} and θ_D, training epochs N, labeling budget b
Ensure:
for $i = 1$ to N **do**
  Compute $\mathcal{L}_{VAE}^{trans}$, $\mathcal{L}_{VAE}^{adv}$, and $\mathcal{L}_{VAE}$ using (5), (6), and (7), respectively
  Compute $\mathcal{L}_D^{adv}$ using (8)
  Update θ_{VAE} and θ_D using (9)
  Select samples X_b with $\min_b D(X_U)$
  Query labels for X_b: $Y_b \leftarrow Oracle(X_b)$
  $(X_L, Y_L) \leftarrow (X_L, Y_L) \cup (X_b, Y_b)$
  $X_U \leftarrow X_U - X_b$
end for
return Updated (X_L, Y_L), X_U

4 Experiments

To evaluate the proposed SS-VAAL framework, we carried out extensive experiments on two computer vision tasks: image classification and semantic segmentation.

4.1 Active Learning for Image Classification

Datasets. To evaluate SS-VAAL, we performed experiments on the following commonly used datasets: CIFAR-10, CIFAR-100 [20], Caltech-101 [11], and ImageNet [9]. Both the CIFAR-10 and CIFAR-100 datasets consist of 50,000 training images and 10,000 test images that are 32×32 in size. The Caltech 101 dataset contains 9,146 images, split between 101 different object categories. Each object category contains between 40 and 800 images, each of which is approximately 300×200 pixels. ImageNet is a large-scale dataset with more than 1.2 million images from 1,000 classes.

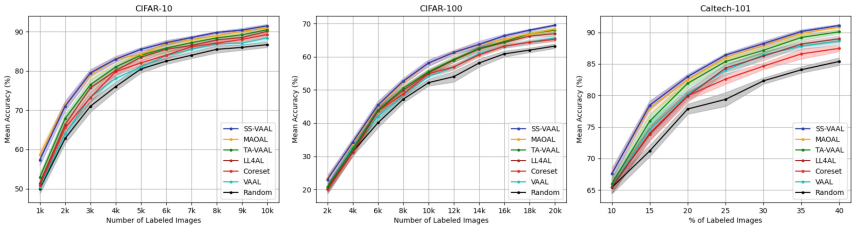
Implementation Details. We first trained a SoDeep sorter to rank the losses. Given the close performance of several available sorter options, we opted for the LSTM sorter. The sorter was trained with a sequence length of 128 for 300 epochs on synthetic data consisting of vectors of generated scalars associated with their ground-truth rank vectors. This training is separate from the AL process. After training was complete, the sorter was applied to the loss prediction module to convert the predicted and target losses into ranking losses for the AL process.

For CIFAR-10 and CIFAR-100, we applied the same data augmentation as the compared methods, including a 32×32 random crop from 36×36 zero-padded images, random horizontal flip, and normalization with the mean and standard deviation of the training set. The target model underwent 200 epochs of training on labeled data with a batch size of 128, then 100 epochs of semi-supervised training on pseudo-labeled data. The initial learning rate was set to 0.1, and reduced to 0.01 and 0.001 at 160 and 240 epochs, respectively. For training, we employed ResNet-18 [15] as the target network with the loss prediction module described in Sect. 3 using stochastic gradient descent

with the momentum set to 0.9 and a weight decay of 0.0005. Experiments began with an initial labeled pool of 1000/2000 images from the CIFAR-10/CIFAR-100 training set, respectively. At each stage, the budget size was 1000 (CIFAR-10)/2000 (CIFAR-100) samples. The pool of unlabeled data consisted of the residual training set from which samples were selected for labeling by an oracle. Upon labeling, these samples were incorporated back into the initial training set and the process was carried out again on the updated training set.

For Caltech-101 and ImageNet, the images were resized to 224×224 and we initiated the process with 10% of the samples from the dataset as labeled data with a budget size equivalent to 5% of the dataset. All other settings remained the same as those used for CIFAR-10 and CIFAR-100, except that the main task was trained for 100 epochs for the ImageNet dataset. The effectiveness of our approach was assessed based on the accuracy of the test data. We compared against a random sampling strategy baseline and state-of-the-art methods including the core-set approach [30], LL4AL [40], VAAL [34], TA-VAAL [19], and MAOAL [13].

Results. All the compared against methods were averaged across 5 trials on the CIFAR-10, CIFAR-100, and Caltech-101 datasets, and across 2 trials on ImageNet. Figure 4 and Fig. 6 (see supplementary material) show the classification accuracy on the benchmark datasets. The results obtained for the competing methods are largely in line with those reported in the literature. Our comprehensive methodology, SS-VAAL, incorporates both the ranking loss prediction module and the clustering-assisted pseudo labeling. The empirical results consistently show that SS-VAAL surpasses all the competing methods at each AL stage.



(a) (b) (c)

Fig. 4. Image classification comparison on the (a) CIFAR-10, (b) CIFAR-100, and (c) Caltech-101 datasets.

4.2 Active Learning for Semantic Segmentation

Experimental Setup. To evaluate the effectiveness of our AL approach in more complex environments, we analyzed the task of semantic segmentation using Cityscapes [8], a large-scale dataset of urban street scene videos. Consistent with the settings in [34], we utilized the dilated residual network [41] as the semantic segmentation model.

Performance was measured by the mean intersection over union (mIoU) metric on the Cityscapes validation set. All other experimental settings were kept consistent with those used in the image classification experiments.

Results. All the compared against methods were averaged across 3 trials and are shown in Fig. 5. Our method consistently outperforms all the other methods on the task of semantic segmentation on the Cityscapes dataset as evidenced by its higher mIoU scores.

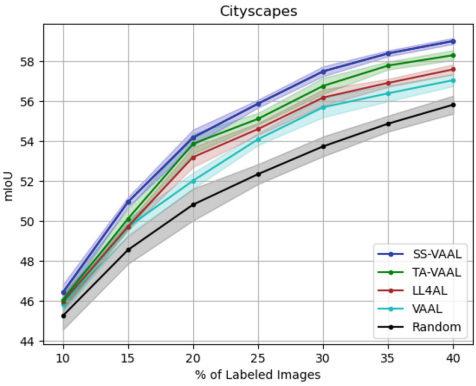


Fig. 5. Semantic segmentation results on the Cityscapes dataset.

4.3 Ablation Study

To assess the impact of each proposed component, we executed an ablation study for the classification task on the CIFAR-10, CIFAR-100, and Caltech-101 datasets. The results are presented in the supplementary material, here we report the main observations. SS-VAAL (w/ ranking only), which refers to the enhancement of VAAL by integrating the ranking loss-based module, outperforms VAAL and LL4AL. This confirms the benefits of considering task-related information in task learning. Moreover, it outperforms TA-VAAL, indicating that our selection of the listwise ranking method more effectively conveys task-related information than that of TA-VAAL (Fig. 7–Fig. 9).

Conversely, SS-VAAL (w/ CAPL only), which entails implementing the proposed clustering-assisted pseudo-labeling procedure at every stage of model training, yields a noticeable improvement over all the other methods. This highlights the effectiveness of exploiting unlabeled data during model training. It also offers a modest improvement over the SS-VAAL (w/ ranking only) configuration, implying that leveraging unlabeled data for training contributes more to the performance improvement than employing alternative means for conveying task-related information (Fig. 10–Fig. 12). Additionally, we contrast this configuration with SS-VAAL (w/ PL only), which represents the use of the conventional pseudo-labeling technique. The increase in performance underscores the effectiveness of our refinement of this method (Fig. 13–Fig. 15).

5 Conclusion

In this paper we developed key enhancements to both better optimize the use of vast amounts of unlabeled data during training and incorporate task-related information. Our approach, SS-VAAL, includes a novel pseudo-labeling algorithm that allows a model to delve deeper into the data space, thus enhancing its representation ability by exploiting all unlabeled data in a semi-supervised way in every AL cycle. SS-VAAL also incorporates a ranking-based loss prediction module that converts predicted losses into a differentiable ranking loss. It can be inserted as a rank variable into VAAL's latent space for adversarial training. Evaluations on image classification and segmentation benchmarks demonstrate the increased performance of SS-VAAL over state-of-the-art techniques.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1–8. IEEE (2020)
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: Proceedings of the International Conference on Learning Representations (2020)
3. Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9368–9377 (2018)
4. Buchert, F., Navab, N., Kim, S.T.: Exploiting diversity of unlabeled data for label-efficient semi-supervised active learning. In: Proceedings of the International Conference on Pattern Recognition, pp. 2063–2069 (2022)
5. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the International Conference on Machine learning, pp. 129–136 (2007)
6. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. J. Artif. Intell. Res. **4**, 129–145 (1996)
7. Coletta, L.F., Ponti, M., Hruschka, E.R., Acharya, A., Ghosh, J.: Combining clustering and active learning for the detection and learning of new image classes. Neurocomputing **358**, 150–165 (2019)
8. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
10. Engilberge, M., Chevallier, L., Pérez, P., Cord, M.: Sodeep: a sorting deep net to learn ranking loss surrogates. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10792–10801 (2019)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 178–178 (2004)
12. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the International Conference on Machine Learning, pp. 1183–1192 (2017)

13. Geng, L., Liu, N., Qin, J.: Multi-classifier adversarial optimization for active learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 7687–7695 (2023)
14. Gilhuber, S., Jahn, P., Ma, Y., Seidl, T.: Verips: verified pseudo-label selection for deep active learning. In: Proceedings of the IEEE International Conference on Data Mining, pp. 951–956 (2022)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Higgins, I., et al.: beta-vae: learning basic visual concepts with a constrained variational framework. In: Proceedings of the International Conference on Learning Representations (2017)
17. Huang, S.J., Jin, R., Zhou, Z.H.: Active learning by querying informative and representative examples. IEEE Trans. Pattern Anal. Mach. Intell. **10**(36), 1936–1949 (2014)
18. Huang, Z., He, Y., Vogt, S., Sick, B.: Uncertainty and utility sampling with pre-clustering. In: Proceedings of the Workshop on Interactive Adaptive Learning (2021)
19. Kim, K., Park, D., Kim, K.I., Chun, S.Y.: Task-aware variational adversarial active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8166–8175 (2021)
20. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report. University of Toronto, Toronto, Ontario (2009)
21. Lee, D.H.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Proceedings of the Workshop on Challenges in Representation Learning, vol. 3, p. 896 (2013)
22. Lewis, D.D.: A sequential algorithm for training text classifiers: corrigendum and additional data. In: Proceedings of the ACM Sigir Forum, vol. 29, pp. 13–19. ACM, New York (1995)
23. Li, M., Liu, X., van de Weijer, J., Raducanu, B.: Learning to rank for active learning: a listwise approach. In: Proceedings of the International Conference on Pattern Recognition, pp. 5587–5594 (2020)
24. Li, X., Guo, Y.: Adaptive active learning for image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 859–866 (2013)
25. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends® Inf. Retr. **3**(3), 225–331 (2009)
26. Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: Proceedings of the International Conference on Machine Learning, p. 79 (2004)
27. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: an uncertainty-aware pseudo-label selection framework for semi-supervised learning. In: Proceedings of the International Conference on Learning Representations (2021)
28. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Proceedings of the Advances in Neural Information Processing Systems, vol. 29 (2016)
29. Saquil, Y., Kim, K.I., Hall, P.: Ranking cgans: subjective control over semantic image attributes. In: Proceedings of the British Machine Vision Conference (2018)
30. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: Proceedings of the International Conference on Learning Representations (2018)
31. Settles, B.: Active learning literature survey. Technical Report. University of Wisconsin-Madison, Department of Computer Science (2009)
32. Siméoni, O., Budnik, M., Avrithis, Y., Gravier, G.: Rethinking deep active learning: using unlabeled data at model training. In: Proceedings of the International Conference on Pattern Recognition, pp. 1220–1227 (2020)

33. Sindhwani, V., Niyogi, P., Belkin, M.: Beyond the point cloud: from transductive to semi-supervised learning. In: Proceedings of the International Conference on Machine Learning, pp. 824–831 (2005)
34. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5972–5981 (2019)
35. https://github.com/robotic-vision-lab/Semi-Supervised-Variational-Adversarial-Active-Learning
36. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2591–2600 (2016)
37. Xia, F., Liu, T.Y., Wang, J., Zhang, W., Li, H.: Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the International Conference on Machine Learning, pp. 1192–1199 (2008)
38. Yan, X., et al.: A clustering-based active learning method to query informative and representative samples. Appl. Intell. **52**(11), 13250–13267 (2022)
39. Yang, Y., Ma, Z., Nie, F., Chang, X., Hauptmann, A.G.: Multi-class active learning by uncertainty sampling with diversity maximization. Int. J. Comput. Vision **113**, 113–127 (2015)
40. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 93–102 (2019)
41. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 472–480 (2017)
42. Zhang, B., Li, L., Yang, S., Wang, S., Zha, Z.J., Huang, Q.: State-relabeling adversarial active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8756–8765 (2020)

Deep Evidential Active Learning with Uncertainty-Aware Determinantal Point Process

Yuxian Zhou[1], Xiaodong Yue[2(✉)], Yufei Chen[3], and Shaorong Xie[1]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
{zyxcarrie,yswantfly}@shu.edu.cn
[2] Artificial Intelligence Institute, Shanghai University, Shanghai, China
srxie@shu.edu.cn
[3] College of Electronic and Information Engineering, Tongji University, Shanghai, China
yufeichen@tongji.edu.cn

Abstract. Deep learning method requires a substantial amount of labeled data to achieve the state-of-the-art performance. However, annotating a large volume of data is often costly and impractical. Active Learning is a approach that reduces labeling costs by intelligently selecting and annotating the most crucial data points, which benefits from the integration of uncertainty and diversity as key criteria for sampling. Existing uncertainty-based methods often fall short in capturing the distinct sources of uncertainty, resulting in a diminished quality of uncertainty estimation. Additionally, clustering is commonly used to ensure diversity, which requires multiple iterations but overlooks the global correlations present throughout the entire unlabeled dataset. As a result, a well-designed ad hoc combination is essential to balance uncertainty and diversity. To address above limitations, we propose Evidential Uncertainty-aware Determinantal Point Process active learning method. Specifically, we employ the theory of Subjective Logic to measure multifaceted uncertainty including vacuity and dissonance. On these grounds, we first focus on the samples with high dissonance and employ Determinantal Point Process to select the samples considering both vacuity and diversity. The proposed method explores and exploits the information associated with the latent feature space near the decision boundary to select the most valuable samples. The experimental results on various real-world datasets reveal the superiority of our method.

Keywords: Active Learning · Evidential Deep Learning · Multifaceted Uncertainty · Determinantal Point Process

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_2.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 17–32, 2025.
https://doi.org/10.1007/978-3-031-78107-0_2

1 Introduction

Deep learning models have shown advanced performance in various domains. Yet, this advantage becomes obvious only when an extensive amount of labeled training data is accessible. In practical scenarios, labeled data is often scarce and annotating a large number of unlabeled samples is impractical. This limitation is detrimental to model performance, especially in knowledge-rich domains. A survey [4] shows that training samples do not improve the performance of the model all the same. So a carefully chosen small training set can be better than a randomly chosen large labeled training set. Active learning algorithms aim to query and annotate the most valuable examples so that a high-performance classifier can be trained with the least amount of labeling effort. Two primary criteria commonly used in devising AL querying strategies are uncertainty and diversity. Recent uncertainty-based sampling strategies [9,29,31] aim to query the most informative samples, which typically only consider single-source uncertainty and overlook the diversity in sample selection. Existing selection strategies for diverse subsets [3,6,33] are not considered global correlations among the entire unlabeled data pool. Furthermore, some methods [32,35] integrating uncertainty and diversity must carefully control the trade-off between multiple criteria. To address these limitations, we consider selecting samples with multifaceted uncertainty, paying simultaneous attention to the diversity within the subsets. Evidential Deep Learning (EDL) [28] is employed to obtain the Dirichlet distribution of prediction and uncertainty based on Subjective Logic (SL) [14]. Two distinct uncertainty *vacuity* and *dissonance* correspond to the lack of evidence and the conflict of evidence. Initially, we prioritize samples exhibiting strong conflict of evidence, characterized by high dissonance. Subsequently, we utilize Determinantal Point Process (DPP) for sample selection, considering both vacuity and diversity. The contributions of this paper are summarized below.

- *Explore multifaceted uncertainty measure.* Evidential uncertainty included vacuity and dissonance concerns samples with different characteristics. It provides valuable information to design sampling strategy.
- *Uncertainty-aware determinantal point process for balancing uncertainty and diversity.* We first incorporate DPP with evidential uncertainty measure into active learning. We can assess the capability of samples to explore new areas for the model at both the feature and decision levels.
- *Propose a two-stage sampling strategy based on multifaceted uncertainty measure.* We propose a two-stage sampling strategy, making it focus on conflict samples and exploring new areas.

2 Related Work

Active Learning in Deep Learning. AL frameworks currently have three main types of query strategies [4,20]: (1) Membership query synthesis. The learners generate query instances in the input space based on the hypothesis model [2]. (2) Stream-based AL. The instances are sampled from some sources

one after another, just like a stream of instances [36]. (3) Pool-based AL. The learner has access to all the unlabeled instances before initiating the sampling process. The learner evaluates or ranks these instances from the unlabeled pool to decide whether to query their labels or not. In real-world scenarios, a significant amount of unlabeled data often exists, which can be thought of as an unlabeled data pool. Pool-based sampling methods are divided into three types. At the data level, representativeness [23,27,32] and diversity [3,6,33] can help the model select informative and diverse samples. At the model level, they often select samples that are beneficial to model performance improvement, such as gradient length [3], fisher information [30] and the loss of models [34]. Furthermore, various metrics are employed to assess the quality of sample, such as uncertainty [26,29,31], distance from adversarial examples [7], etc. These methods are commonly integrated into hybrid approaches to enhance performance.

Uncertainty-Based Learning. Quantifying uncertainty is a crucial research topic in deep learning. One line of research concentrates on combining deep learning with Bayesian probability theory [16]. Another line explores to estimate the uncertainty by other methods [13,21]. While recent efforts provide abundant uncertainty measure, there is a scarcity of methods on how to effectively leverage uncertainty measure for AL. For example, the prior network [24] requires training properly as its parameters must capture knowledge from both the in-domain distribution and the decision boundary, making it not suitable for AL. Considering the shortcomings of existing uncertainty measures for AL, we consider designing our strategy based on evidential uncertainty including vacuity (i.e., lack of evidence) and dissonance (i.e., conflict of evidence). EDL [28] treats the predictions of the neural network as subjective opinions. Based on SL [14], vacuity and dissonance can be computed, which are more suitable for active sampling.

Determinantal Point Processes. Determinantal Point Process is an elegant probabilistic model used in a variety of machine learning tasks. In the discrete case, the point process is a probability measure over all subsets. It links the probability of selecting a subset to the computation of the determinant, which is associated with the kernel matrix that defines global measure of similarity between samples. Similar items are less likely to appear simultaneously, because DPP assigns higher probabilities to more diverse subsets. DPP algorithms include basic DPP [19], k-DPP [18] and Markov DPP [1]. Different DPP algorithms have different application scenarios. Basic DPP [19] can select the subset which has unfixed size. For k-DPP [18] algorithm, we can select a subset of k samples. Markov DPP [1] is also proposed to ensure the diversity of samples in the time series. However, there is few work that applies DPP to active learning.

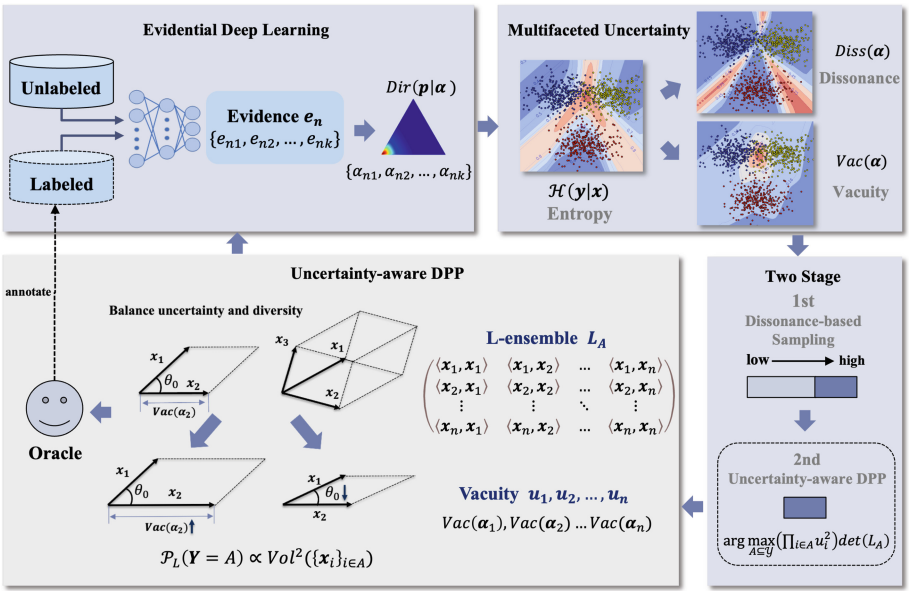


Fig. 1. Module Overview. We utilize Evidential Deep Learning as the evidence generator, which can quantify multifaceted uncertainty including vacuity and dissonance based on Subjective Logic. Two-stage sampling strategy is proposed to select samples with high quality.

3 Method

AL models actively select valuable samples for annotation using various strategies. To select more valuable samples, we propose Evidential Uncertainty-Aware Determinantal Point Process (EUDPP) sampling strategy in Fig. 1, which takes into account both multifaceted uncertainty and diversity. In the first round, we select samples with strong conflict of evidence. Then we select a subset among them based on diversity and vacuity. Specifically, we combine vacuity with similarity matrix for sampling, called uncertainty-aware DPP. In order to make the sampling strategy more effective, we use representative loss providing guidance on the distance of intra-class samples in the feature space.

3.1 Multifaceted Uncertainty Quantification

Entropy is the uncertainty measure that represents the uncertainty of the entire output prediction distribution. Traditional works [8,13] select samples with larger entropy. A challenge associated with entropy-based sampling is that samples with high entropy may not effectively contribute to the enhancement of model performance, since the predicted entropy may be inaccurate, particularly in the early stage of AL. Furthermore, it lacks the ability to differentiate among sources of uncertainty and fails to facilitate the model in learning the

decision boundary. To differentiate among the sources of uncertainty, Eviential Deep Learning [28] can output the collected evidence and measure multifaceted uncertainty. Specifically, Subjective Logic provides a principled way to form subjective opinion and uncertainty mass based on the parameters of Dirichlet distribution [14]. In the multi-class setting, EDL considers a frame of K mutually exclusive singletons (i.e., class labels) by assigning a belief mass $\{b_k\}_{k=1}^{K}$ and providing an overall uncertainty mass u. These K + 1 mass values are all non-negative and add up to one

$$u + \sum_{k=1}^{K} b_k = 1, \tag{1}$$

where $u \geq 0$ and $b_k \geq 0$ for $k = 1, 2, ..., K$. Let α_k be the Dirichlet distribution parameters corresponding to the K class. We can obtain Dirichlet distribution characterized as $Dir(\mathbf{p}|\boldsymbol{\alpha})$. The belief mass b_k and the uncertainty mass u are computed as

$$b_k = \frac{\alpha_k - 1}{S} \quad \text{and} \quad u = \frac{K}{S}, \tag{2}$$

where $S = \sum_{i=1}^{K} \alpha_k$ is referred as Dirichlet strength. The model outputs evidence of samples about each singleton. Let $e_k \geq 0$ be the evidence derived for the k^{th} singleton, which can be computed by $e_k = \alpha_k - 1$.

Insufficient Evidence: Vacuity. The uncertainty mass is related to Dirichlet strength S, which represents the amount of evidence. So we can consider the uncertainty caused by lack of evidence as vacuity: $Vac(\boldsymbol{\alpha}) = K/S$. In Fig. 1, we can see the areas with clusters overlap or fewer samples have higher vacuity. We should choose additional samples from these areas to mitigate uncertainty about lack of evidence.

Conflicting Evidence: Dissonance. Uncertainty that results from the relationship between belief masses can be viewed as the conflict of evidence [15], which is defined as

$$Diss\left(\boldsymbol{\alpha}\right) = \sum_{k=1}^{K} \left(\frac{b_k \sum_{j \neq k} b_j Bal\left(b_j, b_k\right)}{\sum_{j \neq k} b_j} \right), \tag{3}$$

$$Bal\left(b_j, b_k\right) = \begin{cases} 1 - \frac{|b_j - b_k|}{b_j + b_k} & \text{if } b_i b_j \neq 0, \\ 0 & \text{if } \min\left(b_i, b_j\right) = 0. \end{cases} \tag{4}$$

High dissonance indicates that these samples cannot be classified with certainty because both classes are assigned an equal amount of supporting evidence, leading to the strong conflict between belief masses. Samples with high $Diss(\boldsymbol{\alpha})$ tend to be located at the decision boundary, which are beneficial for the model classification task.

By clearly identifying the sources of uncertainty instead of employing them in a combined form as in entropy, evidential uncertainty provides a deeper insight.

We derive the relationship between evidential uncertainty and entropy. We can easily know $p(y = k) = 1/K$ when the entropy takes the maximum value. Thus the two theorems are formalized below.

Theorem 1. *Given a Dirichlet Strength S, if $S \to \infty$, we have*

$$q = \underset{x_i \in D_u}{argmax} Diss(\boldsymbol{\alpha}_i) \Rightarrow q = \underset{x_i \in D_u}{argmax} H(y_i|x_i). \tag{5}$$

Proof. If we have $Diss(\boldsymbol{\alpha}_q) = 1$, which indicates $e_{q1} = e_{q2} = \ldots = e_{qk}$ and $e_{qk} = \alpha_{qk} - 1$, then

$$\lim_{S \to \infty} p(y = k) = \lim_{S \to \infty} \frac{e_{qk} + 1}{S} = \lim_{S \to \infty} \frac{\frac{S}{K} + 1}{S} = \frac{1}{K} \tag{6}$$

Theorem 2. *Given a Dirichlet Strength S, we have*

$$q = \underset{x_i \in D_u}{argmax} Vac(\boldsymbol{\alpha}_i) \Rightarrow q = \underset{x_i \in D_u}{argmax} H(y_i|x_i). \tag{7}$$

Proof 1. If we have $Vac(\boldsymbol{\alpha}_q) = 1$, which indicates $e_{q1} = e_{q2} = \ldots = e_{qk} = 0$. i.e., $S = K$, then

$$p(y = k) = \frac{e_{qk} + 1}{S} = \frac{1}{K}. \tag{8}$$

For a high-entropy data, the uncertainty is caused by high dissonance (i.e., conflicting evidence and $S \gg K$) or a high vacuity (i.e., insufficient evidence and $S \approx K$). The multifaceted uncertainty provides valuable information to design an effective sampling strategy.

3.2 Balanced Uncertainty and Diversity Sample Selection with Determinantal Point Process

Uncertainty-based methods do not fully leverage the data distribution, while samples chosen solely based on data distribution may offer limited new information, indicating low uncertainty. Our sampling strategy meets two important criteria: (1) help the model to learn the decision boundary. (2) the selected subset must have diversity and information to obtain more various samples.

Dissonance-Based Sampling. The conflicting evidence indicates that samples are near the decision boundary where classes are heavily overlapped. Labeling samples with high dissonance helps the model have a better ability to discriminate.

$$q = \underset{x_i \in D_u}{\text{argmax}} Diss(\boldsymbol{\alpha}_i) \tag{9}$$

we obtain n_1 samples according Eq. 9, which are selected in the first stage.

Vacuity-Aware Determinantal Point Process Sampling. Diversity plays a crucial role in sampling strategies. Recent studies [3,32,33] propose sampling strategies based on diversity. In the first stage, we already obtain n_1 samples. In this stage, we pay attention to informative samples measured by $Vac(\boldsymbol{\alpha}_i)$ and diversity. We utilize determinantal point process with vacuity, which consider both data distribution and the impact of samples on the model. We regard $Vac(\boldsymbol{\alpha}_i)$ as scores of unlabeled samples and combine scores with DPP [18].

Definition 1. ***Marginal DPP Distribution.*** *Determinantal Point Process $\mathcal{P}$ is a probability distribution over 2^n subsets of a discrete set $\mathcal{Y} = \{1, 2, ..., n\}$. Drawing a random subset $\boldsymbol{Y}$ according to $\mathcal{P}$, for every subset $A \subseteq \boldsymbol{Y}$, the probability of subset selection is formalized as*

$$\Pr(A \subseteq \boldsymbol{Y}) = \det(K_A), \tag{10}$$

for the positive semidefinite matrix K indexed by the elements of $\boldsymbol{Y}$. $K_A \equiv [K_{ij}]_{i,j \in A}$ denotes a submatrix of K to the entries indexed by the elements of A. We refer to K as the marginal kernel, as all the eigenvalues of K should be less than or equal to one. Large values of k_{ij} imply that i and j tend not to co-occur. Therefore, DPPs can be used naturally to model diverse sets of items.

For modeling real-world data, the L-ensemble offers a more flexible approach to constructing DPPs, free from the constraints imposed by eigenvalues. The L-ensemble [18] defines a DPP via a positive semidefinite matrix $L \in \mathbb{R}^{n \times n}$ indexed by the elements of $\mathcal{Y}$

$$\mathcal{P}_L(\boldsymbol{Y} = A) = \frac{\det(L_A)}{\det(L + I)}, \tag{11}$$

$$L = \boldsymbol{X}^\top \boldsymbol{X} = \begin{pmatrix} \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle & \dots & \langle \boldsymbol{x}_1, \boldsymbol{x}_n \rangle \\ \langle \boldsymbol{x}_2, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_2, \boldsymbol{x}_2 \rangle & \dots & \langle \boldsymbol{x}_2, \boldsymbol{x}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{x}_n, \boldsymbol{x}_1 \rangle & \langle \boldsymbol{x}_n, \boldsymbol{x}_2 \rangle & \dots & \langle \boldsymbol{x}_n, \boldsymbol{x}_n \rangle \end{pmatrix}, \tag{12}$$

where I is an identity matrix and L_A denotes a submatrix of L. L_A is the set of $\langle x_i, x_j \rangle$ about samples in set A. For every subset $A \subseteq \mathcal{Y}$, the probability of subset is

$$\mathcal{P}_L(\mathbf{Y} = A) \propto \det(L_A) = \mathrm{Vol}^2\left(\{\boldsymbol{x}_i\}_{i \in A}\right), \tag{13}$$

where $\det(L_A)$ the determinant of L_A. Determinants have an intuitive geometric interpretation. In Fig. 1, the determinant of the matrix L is the square of the volume of the parallel polyhedron formed by the vectors.

Balancing Uncertainty and Diversity. In contrast to existing methods relying on pairwise dissimilarities [3,33], our approach defines diversity in the feature space of the entire subset based on Determinantal Point Process (DPP). Selecting a sample set that maintains maximum diversity is advantageous for the model, but without considering measures of uncertainty, the selected samples may contain little new information. Therefore, to balance diversity and uncertainty, we

employ vacuity $Vac(\boldsymbol{\alpha_i})$ as a score to assess the quality of samples, helping our model in gathering evidence. We construct the matrix of uncertainty scores

$$U = \begin{bmatrix} u_1 & 0 & \cdots & 0 \\ 0 & u_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_n \end{bmatrix}, \tag{14}$$

where u_i is $Vac(\boldsymbol{\alpha}_i)$. We combine the matrix of uncertainty scores U with matrix L. We define X as the matrix of feature vectors $\begin{bmatrix} \boldsymbol{x}_1 \, \boldsymbol{x}_2 \, \ldots \, \boldsymbol{x}_n \end{bmatrix}$. Let's start by combining the feature vectors $\boldsymbol{x}_n$ and scores u_n into weighted samples $\boldsymbol{x}'_n$

$$\begin{bmatrix} u_1\boldsymbol{x}_1 \, u_2\boldsymbol{x}_2 \, \ldots \, u_n\boldsymbol{x}_n \end{bmatrix} = \boldsymbol{X}' = \boldsymbol{X}U. \tag{15}$$

Subsequently, we can obtain the new matrix L with samples' score

$$\begin{aligned} L'\left(\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n\right) &= \boldsymbol{X'}^{\top}\boldsymbol{X}' \\ &= (\boldsymbol{X}U)^{\top}(\boldsymbol{X}U) \\ &= U^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}U, \end{aligned} \tag{16}$$

There are many kinds of metrics, such as euclidean distance, cosine similarity, RBF, etc. We use the following shorthand for similarity measure

$$L_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \frac{{\boldsymbol{x}_i}^T \boldsymbol{x}_j}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_j\|_2}. \tag{17}$$

Therefore, the probability (diversity) of the subset sampled by uncertainty-aware DPP can be equivalent to computing the determinant

$$\mathcal{P}_L(\boldsymbol{Y} = A) \propto \left(\prod_{i \in A} u_i^2\right) \det\left(L_A\right), \tag{18}$$

where the first term increases with the uncertainty scores of the selected samples, and the second term increases with the diversity of the selected subsets.

Furthermore, we have the geometric intuitions that the determinant of L'_A is equal to the squared volume of the parallel polyhedron spanned by the vectors $u_i\boldsymbol{x}_i$ for $i \in A$. Figure 1 shows the magnitude of u_i and the direction $\boldsymbol{x}_i$. The larger the volume is, the determinant of the matrix is increasing and the corresponding subset is also more diverse.

Fast Greedy MAP Inference for DPP. We can sample subsets by the MAP inference [10]

$$A_{\text{map}} = \arg\max_{A \subseteq \mathcal{Y}} \left(\prod_{i \in A} u_i^2\right) \det\left(L_A\right). \tag{19}$$

We know that MAP inference is a NP-hard problem and greedy algorithm is the widely used for it, justified by the fact that the log-probability of set in DPP is submodular. Due to the submodular nature of the MAP objective, greedy algorithms have been used with empirical success. The submodular maximization task is to find a subset maximizing a submodular function, which corresponds to the MAP inference in the DPP case [11]. Its greedy strategy is formalized as

$$j = \arg \max_{i \in D_u \setminus A} \left[log\left(u_i^2\right) + \log\det\left(L_{A\cup\{i\}}\right) - \log\det\left(L_A\right) \right], \tag{20}$$

where D_u is the unlabeled dataset and A represents subsets having been selected. We can use fast greedy map inference [5] to obtain n_2 samples. The selected samples contribute not only learn the model's decision boundary but also offer diversity and information, enabling exploration of new regions.

3.3 Evidential Deep Learning for Uncertainty Measure

EDL can form multinomial opinions for the classification [28] of a given sample i as a Dirichlet distribution $Dir(\boldsymbol{p}_i|\boldsymbol{\alpha}_i)$, where $\boldsymbol{p}_i$ is a simplex representing class assignment probabilities $\boldsymbol{p}_i = \boldsymbol{\alpha}_i/S$. By placing a Dirichlet distribution on the class probabilities, EDL learns to obtain the uncertainty of the prediction from the evidence collected from the data. The prediction for the classification problem is a Dirichlet distribution whose parameters are set by the output of the neural network. The loss function of prediction error is formalized as

$$\begin{aligned} \mathcal{L}_{ce}^{(i)}(\boldsymbol{\alpha_i}) &= \int \left[\sum_{j=1}^{K} -y_{ij} \log\left(p_{ij}\right) \right] \frac{1}{B\left(\boldsymbol{\alpha}_i\right)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} d\boldsymbol{p}_i \\ &= \sum_{j=1}^{K} y_{ij} \left(\psi\left(S_i\right) - \psi\left(\alpha_{ij}\right)\right), \end{aligned} \tag{21}$$

where $\psi(\cdot)$ is the *digamma* function and $B(\boldsymbol{\alpha})$ is the K-dimensional multinomial beta function. As long as the correct category has the highest evidence value, the neural network will generate evidence for other wrong labels. If it cannot be correctly classified, we prefer zero total evidence, i.e., $S = K$, corresponds to the uniform distribution and indicates uncertainty, i.e., $u = 1$. So we should add a component to normalize predictive distribution

$$\mathcal{L}_{edl} = \sum_{i=1}^{N} \mathcal{L}_{ce}^{(i)}(\boldsymbol{\alpha}_i) + \lambda \sum_{i=1}^{N} KL\left[Dir\left(\boldsymbol{p}_i \mid \tilde{\boldsymbol{\alpha}}_i\right) \| Dir\left(\boldsymbol{p}_i \mid \mathbf{1}\right)\right], \tag{22}$$

where $\lambda = min\{1, t/T\}$ is annealing coefficient with current epoch number t and total epoch number T. The Dirichlet parameters after removal of the non-misleading evidence from predicted parameters $\boldsymbol{\alpha}_i$ can be formalized as $\tilde{\boldsymbol{\alpha}}_i = \boldsymbol{y}_i + (1 - \boldsymbol{y}_i) \odot \boldsymbol{\alpha}_i$. Moreover, due to the sampling strategy considering diversity of subsets, the representative loss $\mathcal{L}_{rep} = \frac{1}{2} \sum_{i=1}^{m} \| f_i - \mathrm{c}_{y_i} \|_2^2$ contributes to make

intra-class samples of labeled set more clustered in the feature space, where f_i is feature vector and c_{y_i} is the center of class about training data. The distance of intra-class samples in the training set is reduced. At the same time, the distance between samples of the unlabeled set that are similar to the training data is also reduced. Based on our strategy for obtaining diverse subsets, $\mathcal{L}_{rep}$ can help us to select more valuable samples.

4 Experiment

We validate the effectiveness of our method on three datasets: MNIST [22], SVHN [25] and CIFAR10 [17]. Experiments include model performance comparison and the ablation study. For MNIST and SVHN datasets, the initial size of training datasets is set to 100 and we select 100 samples per round for a total of 15 rounds. For CIFAR10 dataset, the initial size of training datasets is set to 1000 and we select 1000 samples per round for a total of 20 rounds. The performance of the models is compared when starting with 5 classes and 10 classes. We use LeNet as the the network structure of feature extraction.

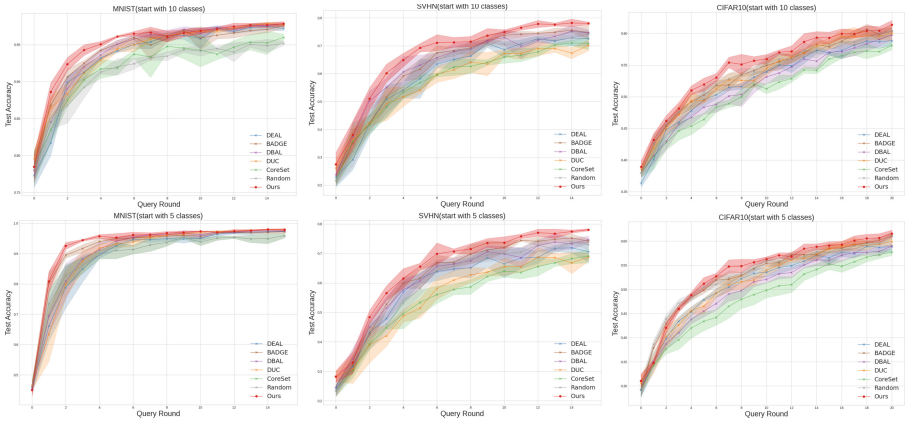


Fig. 2. Performance comparison. Test accuracy under a range of conditions. The shaded area represents the standard error.

4.1 Model Performance Comparison

Evaluating the quality of selected samples involves observing test accuracy after each round of sampling and the diversity of selected subsets. We compare our method with the following state-of-the-art approaches: DUC [31], DEAL [12], BADGE [3], Core-set [27], DBAL [9] and random sampling.

Test Accuracy. The purpose of sampling is to obtain high-quality samples added to the training set. Therefore, test accuracy is used as an external representation of the informative samples. In Fig. 2, we compare three types of sampling methods based on uncertainty, representativeness and diversity, respectively. The test accuracy of our method is higher than other methods. In the setting of starting categories of 5, our method still maintains a good performance.

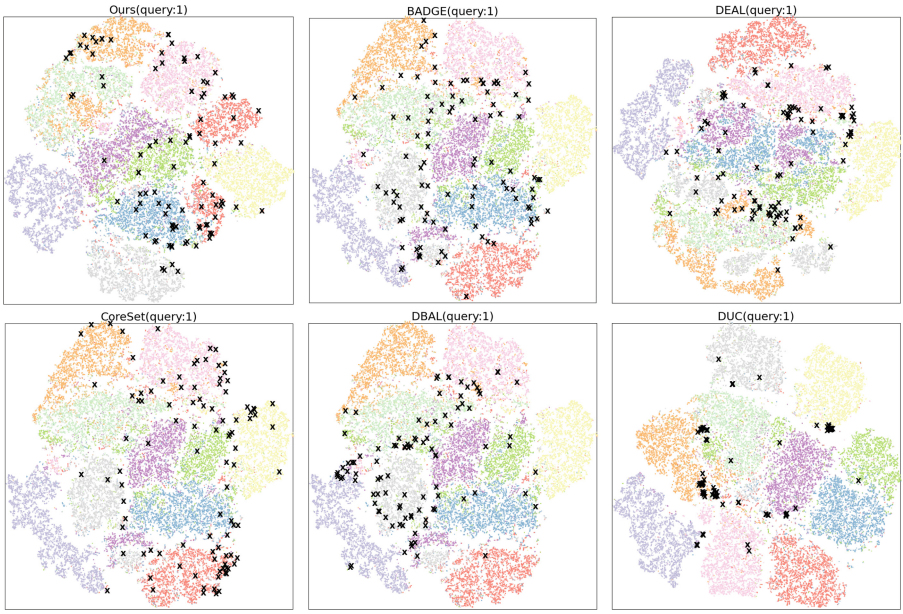


Fig. 3. T-SNE on MNIST dataset. Black points are the selected samples in the first round. Points in other colors are unlabeled samples.

Diversity. The diversity of subsets needs to be guaranteed to save the cost of human annotation. We measure the diversity of annotated samples respectively from the average cosine distance, class distribution and t-SNE. In Fig. 3, we select samples that effectively span most of the distribution area. Except for diversity-based methods, other methods select samples clustered in the feature space. Figure 4 shows the cosine distance and class distribution of the selected subsets. We can observe that our method selects the most dissimilar subsets, maximizing cosine distance, while maintaining uniformity in the number of selected samples for each class.

4.2 Ablation Study

Obviously, diversity is not necessarily conducive to the improvement of model classification performance without restrictions. Therefore, we give priority to

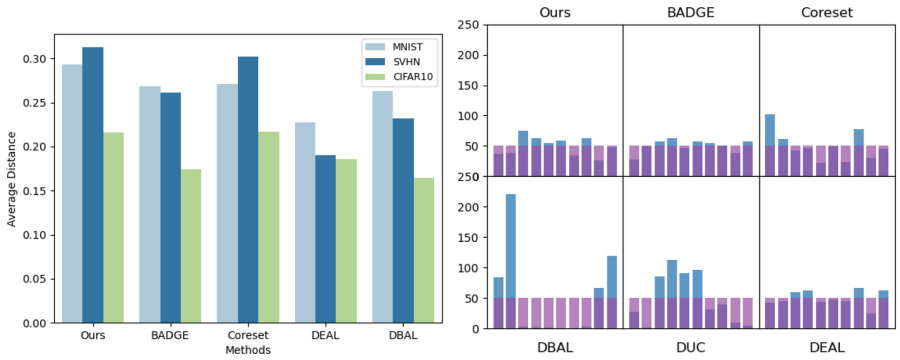


Fig. 4. Diversity comparison. (Left) The average cosine distances of selected samples. (Right) The category distribution of selected samples.

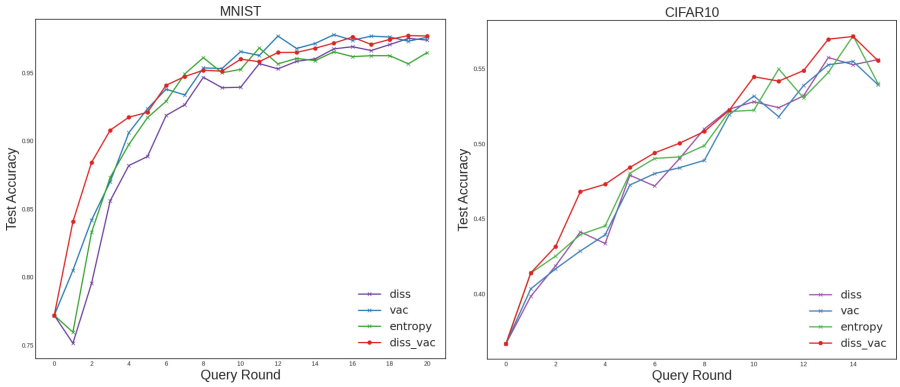


Fig. 5. Uncertainty-based sampling strategy comparison. Entropy, vacuity only, dissonance only and multifaceted uncertainty.

the samples at the decision boundary. Due to the small amount of initial training data, selecting samples at the boundary will limit the area covered by the training set. So in the second stage, we propose the uncertainty-aware DPP. We respectively performed ablation experiments on the performance of uncertainty estimation and the effectiveness of uncertainty-aware DPP sampling strategy.

Uncertainty Estimation. The uncertainty of a high-entropy sample may be caused by either insufficient evidence (i.e., high vacuity) or conflicting evidence (i.e., high dissonance). We use multifaceted uncertainty to sample in two stages separately. In Fig. 5, we compare the experimental results of single uncertainty, entropy and multifaceted uncertainty. We find that the multifaceted uncertainty is more beneficial to select high-quality samples.

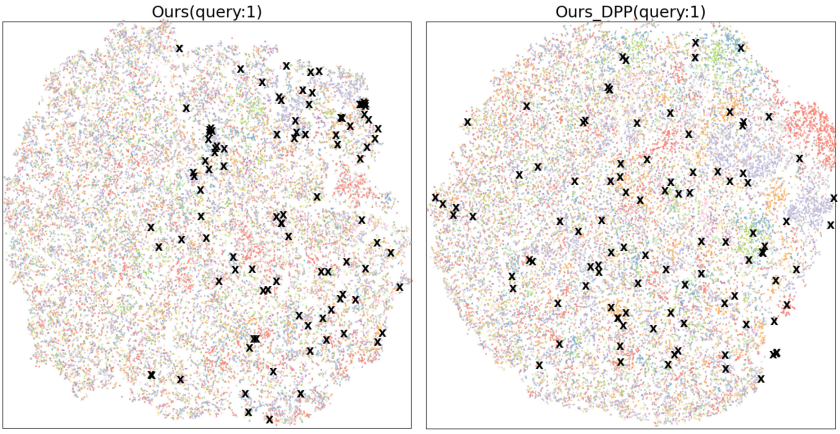


Fig. 6. T-SNE on CIFAR10 dataset. Uncertainty-based sampling strategy select samples clustered in the feature space. DPP can help increase diversity of subsets.

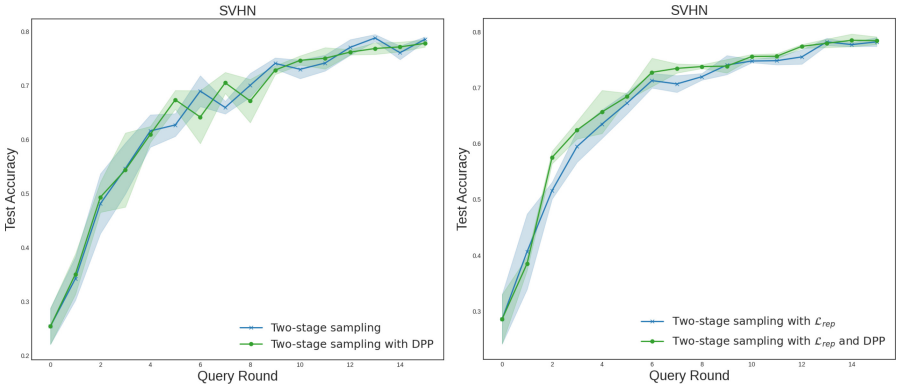


Fig. 7. Ablation study on $\mathcal{L}_{rep}$. It has a positive impact on the DPP-based sampling strategy, making it easier to select samples that have not been seen in the training set.

The Performance of the Uncertainty-Aware DPP. We are the first to apply the relevant theory of DPP in active learning. We use $Vac(\alpha_i)$ as the sample quality score combined with the similarity matrix for sample selection, considering the information about the model learning and diversity in the feature space. Figure 6 shows t-SNE of the samples obtained by the sampling strategy. We can see that selected subset based on our method has more diverse samples. The sampling strategy based on UDPP has a significant improvement on the diversity of subset. We add additional guidance to help the model focus on valuable samples. $\mathcal{L}_{rep}$ can reduce the diversity of intra-class samples in labeled set, which is beneficial to our sampling strategy. In Fig. 7, we can see the test accuracy has a certain increase, so adding $\mathcal{L}_{rep}$ can help our strategy based on UDPP have more noticeable effect.

5 Conclusion

In this paper, we propose a two-stage active learning framework called EUDPP that systematically incorporates multifaceted uncertainty and diversity, to investigate and leverage the data associated with the hidden feature space close to the decision boundary. Through theoretical analysis, we identify the relationship between entropy and multifaceted uncertainty including vacuity and dissonance, highlighting that our multifaceted uncertainty offers profound insights. In the initial stage, dissonance uncertainty is employed to select conflicting samples. Subsequently, vacuity uncertainty is integrated with the Determinantal Point Process to form the uncertainty-aware DPP, which identifies the most informative and diverse subset of conflicting samples while taking into account the global correlations within the entire unlabeled pool. Extensive experiments on various real-world datasets validates the effectiveness of our method.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Serial Nos. 61991410, 62173252, 61976134), OpenProject Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province, China (No. CICIP2021001), Natural Science Foundation of Shanghai (No. 21ZR1423900), and Shanghai Science and Technology Innovation Action Plan (22511101903).

References

1. Affandi, R.H., Kulesza, A., Fox, E.B.: Markov determinantal point processes. arXiv preprint arXiv:1210.4850 (2012)
2. Angluin, D.: Queries and concept learning. Mach. Learn. **2**, 319–342 (1988)
3. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: International Conference on Learning Representations (2019)
4. Budd, S., Robinson, E.C., Kainz, B.: A survey on active learning and human-in-the-loop deep learning for medical image analysis. Med. Image Anal. **71**, 102062 (2021)
5. Chen, L., Zhang, G., Zhou, E.: Fast greedy map inference for determinantal point process to improve recommendation diversity. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
6. Citovsky, G., et al.: Batch active learning at scale. In: Advances in Neural Information Processing Systems, vol. 34, pp. 11933–11944 (2021)
7. Ducoffe, M., Precioso, F.: Adversarial active learning for deep networks: a margin based approach. arXiv preprint arXiv:1802.09841 (2018)
8. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowl. Inf. Syst. **35**, 249–283 (2013)
9. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: International Conference on Machine Learning, pp. 1183–1192. PMLR (2017)
10. Gillenwater, J., Kulesza, A., Taskar, B.: Near-optimal map inference for determinantal point processes. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
11. Han, I., Kambadur, P., Park, K., Shin, J.: Faster greedy map inference for determinantal point processes. In: International Conference on Machine Learning, pp. 1384–1393. PMLR (2017)

12. Hemmer, P., Kühl, N., Schöffer, J.: Deal: deep evidential active learning for image classification. Deep Learn. Appl. **3**, 171–192 (2022)
13. Holub, A., Perona, P., Burl, M.C.: Entropy-based active learning for object recognition. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8. IEEE (2008)
14. Jøsang, A.: Subjective Logic, vol. 3. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-42337-1
15. Josang, A., Cho, J.H., Chen, F.: Uncertainty characteristics of subjective opinions. In: 2018 21st International Conference on Information Fusion (FUSION), pp. 1998–2005. IEEE (2018)
16. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
18. Kulesza, A., Taskar, B.: k-dpps: fixed-size determinantal point processes. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1193–1200 (2011)
19. Kulesza, A., Taskar, B., et al.: Determinantal point processes for machine learning. Found. Trends® Mach. Learn. **5**(2–3), 123–286 (2012)
20. Kumar, P., Gupta, A.: Active learning query strategies for classification, regression, and clustering: a survey. J. Comput. Sci. Technol. **35**, 913–945 (2020)
21. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Neural Information Processing Systems (2017)
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
23. Liu, P., Zhang, H., Eom, K.B.: Active deep learning for classification of hyperspectral images. IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens. **10**(2), 712–724 (2016)
24. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
25. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, vol. 2011, p. 7 (2011)
26. Pandey, D.S., Yu, Q.: Multidimensional belief quantification for label-efficient meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14391–14400 (2022)
27. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: International Conference on Learning Representations (2018)
28. Sensoy, M., Kaplan, L., Kandemir, M.: Evidential deep learning to quantify classification uncertainty. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
29. Shi, W., Zhao, X., Chen, F., Yu, Q.: Multifaceted uncertainty estimation for label-efficient deep learning. In: Advances in Neural Information Processing Systems, vol. 33, pp. 17247–17257 (2020)
30. Sourati, J., Gholipour, A., Dy, J.G., Tomas-Fernandez, X., Kurugol, S., Warfield, S.K.: Intelligent labeling based on fisher information for medical image segmentation using deep learning. IEEE Trans. Med. Imaging **38**(11), 2642–2653 (2019)

31. Xie, M., Li, S., Zhang, R., Liu, C.H.: Dirichlet-based uncertainty calibration for active domain adaptation. arXiv preprint arXiv:2302.13824 (2023)
32. Yang, Y., Zhou, D.W., Zhan, D.C., Xiong, H., Jiang, Y., Yang, J.: Cost-effective incremental deep model: matching model capacity with the least sampling. IEEE Trans. Knowl. Data Eng. (2021)
33. Yang, Y., Ma, Z., Nie, F., Chang, X., Hauptmann, A.G.: Multi-class active learning by uncertainty sampling with diversity maximization. Int. J. Comput. Vision **113**, 113–127 (2015)
34. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 93–102 (2019)
35. Yuan, J., Hou, X., Xiao, Y., Cao, D., Guan, W., Nie, L.: Multi-criteria active deep learning for image classification. Knowl.-Based Syst. **172**, 86–94 (2019)
36. Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active learning from data streams. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 757–762. IEEE (2007)

Knowledge Distillation in Deep Networks Under a Constrained Query Budget

Ankita Singh and Shayok Chakraborty(✉)

Department of Computer Science, Florida State University, Tallahassee, USA
shayok@cs.fsu.edu

Abstract. Knowledge distillation addresses the problem of training a lightweight model (student) from a deeper, more complex model (teacher) so as to mimic its performance. Existing techniques mostly utilize the predictions furnished by the teacher on a given training set to perform the distillation and train the student. However, querying the teacher model for labels can be an expensive process in terms of computational/ financial overhead. In this paper, we tackle the problem of distilling knowledge from a blackbox teacher model into a student deep neural network, in a cost-efficient manner. Active learning algorithms automatically identify the salient and exemplar samples from large amounts of unlabeled data and are instrumental in reducing human annotation effort in inducing a machine learning model. We propose a novel active learning algorithm using which the student model can identify the most informative samples from a large amount of unlabeled data, which need to be queried from the teacher. We exploit the geometry of the unlabeled data to identify a batch of representative samples which can reconstruct the data with minimal error. We pose the sample selection as an NP-hard optimization problem and solve it efficiently using an iterative algorithm, with global convergence. Such an algorithm can be effective in distilling relevant knowledge from the teacher to the student under a constrained query budget. Our extensive empirical studies on five challenging datasets from two application domains (computer vision and text mining) corroborate the efficacy of our active sampling framework over competing baselines.

Keywords: Knowledge distillation · Deep learning · Subset selection

1 Introduction

Knowledge distillation (KD) is a method for transferring complex mapping functions learned by a high-capacity model or an ensemble of multiple models (the teacher) to a relatively simpler, lightweight model (the student) [9,11]. Generally, the teacher models deliver good generalization performance; however, they have a high memory footprint and are computationally expensive. The student

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_3.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 33–48, 2025.
https://doi.org/10.1007/978-3-031-78107-0_3

models, on the other hand, require much less memory and computation and are thus more suitable for real-time applications. KD has been used in a variety of applications, such as pose estimation [25], object detection [6] and video representation [7] among others.

In any KD application, the objective is to train the student network to imitate the teacher network, using a given training set. The transfer of knowledge from the teacher to the student is typically facilitated using a variety of methods, such as matching the soft-label probabilities, the l_2 norm between the feature representations or the attention maps, and the maximum mean discrepancy (MMD) between the distributions of the neuron selectivity patterns learned by the teacher and the student networks [11,15,29] among others. In all these methods, all the training examples are required to be passed to the teacher, and its outputs are used to compute a distillation loss and train the student. However, in certain applications, querying the teacher model can be expensive computationally and/or financially. For instance, the teachers are often models that are trained and hosted by companies on the cloud, commonly referred to as Machine Learning as a Service (MLaaS) platforms. Third-party developers access these models through Application Programming Interfaces (APIs). Each access to the API incurs a cost, which means that a price needs to be paid each time the teacher model is queried. In such applications, obtaining the teacher's output on all the training samples in order to train the student, can be prohibitive. This necessitates the development of an algorithm to distill knowledge from the teacher to the student network, when the number of label queries to the teacher cannot exceed a pre-specified budget.

We formally pose the research question as follows: *We are given a blackbox teacher model (deep neural network) trained on a given application of interest. The data used to train the teacher is not available; the specific architecture and trained parameters of the teacher are also not known. We are interested to train a student deep neural network using a knowledge distillation algorithm to imitate the teacher. For this purpose, we are given a small amount of labeled data L, and a large amount of unlabeled data U, with $|L| \ll |U|$. However, we are not allowed to query the labels of all the $|U|$ samples from the teacher due to computational and/or financial cost constraints. We are further given a query budget k which denotes the number of unlabeled samples whose labels can be queried from the teacher. Which k samples should we select for query so that the student model's generalization accuracy gets closest to that of the teacher?*

Active learning (AL) is a machine learning paradigm to automatically identify the most informative samples from large amounts of unlabeled data [34]. This tremendously reduces human annotation effort in inducing a model, as only the few samples that are selected by the algorithm, need to be labeled manually. Further, since the model gets trained on the exemplar data samples, it typically depicts better generalization accuracy than a passive learner, where the training data is sampled at random. AL has been successfully used in a variety of applications, such as computer vision [43], text analytics [38], computational biology [26], email classification [32] etc.

In this paper, we propose a novel AL algorithm to address the aforementioned research question. We pose the sample selection as a constrained NP-hard optimization problem (based on the data reconstruction error) and derive an iterative algorithm, with global convergence, to solve it. Our framework is easy to implement and independent of the underlying KD algorithm, as well as the architectures of the teacher and student networks. The proposed algorithm is generic and can be used in any application to select an informative subset of samples from large amounts of data; we validate it on the KD application in this paper, as research in this area is still in a nascent stage.

2 Related Work

Active Learning: With the advent and popularity of deep neural networks, deep active learning (DAL) has attracted significant research attention, where the objective is to identify the salient unlabeled samples for manual annotation and simultaneously learn discriminating feature representations using a deep neural network [28]. Recently proposed DAL strategies include learning a task-agnostic loss function to identify the informative unlabeled samples [43], finding a core set of samples such that the deep model trained on this subset is competitive over the whole dataset [33], a method based on diverse gradient embeddings (BADGE) which combines uncertainty and diversity for active sample selection [2], a discriminative algorithm that selects samples such that the labeled and the unlabeled sets are maximally similar [8] and methods based on adversarial learning [36,48]. Other related research in AL includes active learning in the presence of noisy annotators [14], actively completing an incomplete data matrix [30], combining active learning with transfer learning [37], actively selecting the informative features and samples [20] and AL with novel annotation mechanisms [12] among others.

Knowledge Distillation: Knowledge distillation has received increasing attention from the research community in recent years; please refer [9] for a comprehensive survey. A simple and effective idea to transfer knowledge is to match the responses [11], learned feature representations [29] or relationships between different layers [19] between the student and the teacher models. Metrics such as the KL-divergence, Maximum Mean Discrepancy (MMD), l_2 and l_1 norm distance are commonly used to compute the similarity and formulate the distillation loss terms to train the student network. Several distillation techniques have been explored to improve the transfer of knowledge from the teacher to the student in more complex settings, including adversarial distillation [22], multi-teacher distillation [45] and graph-based distillation [5] among others. Recently, a body of research has focused on reducing the amount of training data required foreffective transfer of knowledge. Few-shot KD has been proposed to retain the

teacher model's performance with pseudo samplers which are generated in an adversarial manner [16]. Zero-shot KD has also been explored by generating data using the gradient information of the teacher network [23]. However, these methods require the gradient information of the teacher network, which is difficult to obtain in real-world applications.

Active Learning for Knowledge Distillation: Even though both AL and KD have been extensively studied, AL for KD is in its nascent stage and has only been explored in recent years. As in conventional AL, uncertainty sampling has been exploited to actively select unlabeled samples to train the student model [3,27,46]. Very recently, researchers have begun to study the performance of deep active learning algorithms such as BADGE and Coreset for KD [13,18]. Wang *et al.* [39] proposed mixup together with active learning to augment the unlabeled pool with synthetic data samples, and then query the labels of the hard examples from the teacher to train the student. However, as noted by the authors, mixup may produce data samples that are semantically meaningless, and the knowledge gained by the student from such fake (sample-label) pairs may not be substantial. In contrast, we propose a method to identify the informative unlabeled samples to train the student without generating synthetic/fake data samples. We now describe our framework.

3 Proposed Framework

3.1 Problem Setup

In our problem setup, the student model is given a labeled set L and an unlabeled set U, where $|L| \ll |U|$. Let n be the number of unlabeled samples, where each sample is represented using a vector of d dimensions. Let $X \in \Re^{d \times n}$ denote the unlabeled data matrix, where each column represents a sample and each row represents a feature. Our objective is to select k samples from U to distill knowledge from the teacher and train the student model. Our method is motivated by research in transductive experimental design [35,44], which attempts to select a representative subset such that the whole dataset can be approximated by a linear combination of the selected samples. We formulate the active sampling problem based on data geometry and attempt to select k samples using which the unlabeled data can be reconstructed with minimal error.

3.2 Active Sample Selection

Let $z \in \{0,1\}^{n \times 1}$ be a binary selection vector where $z_i = 1$ if unlabeled sample x_i is selected in the batch and $z_i = 0$ otherwise; let $diag(z)$ be a diagonal matrix with z along the main diagonal. We pose the sample selection as minimizing the following residual:

$$\min_{z,\widehat{C}} \left\| X - X diag(z)\widehat{C} \right\|_F^2 \tag{1}$$

$$\text{s.t.:} \quad z \in \{0,1\}^{n\times 1}, \qquad \sum_{i=1}^{n} z_i = k$$

where $||.||_F$ denotes the matrix Frobenius norm. The term $Xdiag(z)$ attempts to retain k columns (data samples) in the matrix X and set the remaining $(n-k)$ columns to 0; these k samples therefore denote the most representative samples to reconstruct the unlabeled data matrix X. $\widehat{C} \in \Re^{n\times n}$ is a matrix of reconstruction coefficients. We decompose $\widehat{C} = CX$ where $C \in \Re^{n\times d}$ and express the problem as:

$$\min_{z,C} \|X - X diag(z) CX\|_F^2 \tag{2}$$

$$\text{s.t.:} \quad z \in \{0,1\}^{n\times 1}, \qquad \sum_{i=1}^{n} z_i = k$$

This can be written equivalently as:

$$\min_{Q} \|X - XQX\|_F^2 \tag{3}$$

$$\text{s.t.:} \quad \|Q\|_{2,0} = k$$

where $Q = diag(z).C$ and $\|Q\|_{2,0}$ denotes the $l_{2,0}$ norm of a matrix, that is, the number of non-zero rows in the matrix Q. To see the equivalence between Eqs. (2) and (3), we note that, if a particular row of $diag(z)$ has all 0 s, that row in Q will also have all 0 s. Hence, the number of non-zero rows in Q is equal to the number of non-zero entries in $diag(z)$, that is, $\|Q\|_{2,0} = k$. Based on this, we propose to optimize the following objective function:

$$\min_{Q} \|X - XQX\|_F^2 + \alpha \|Q\|_{2,0} \tag{4}$$

where $\alpha \geq 0$ is a regularization parameter. Once we solve for Q, we can compute the l_2 norm of each row of Q and select the k unlabeled samples corresponding to the k highest l_2 norm values. However, this is an NP-hard problem due to the matrix $l_{2,0}$ norm. Nie *et al.* [24] established that the $l_{2,1}$ norm of a matrix is the minimum convex hull of the $l_{2,0}$ norm, and minimizing the $l_{2,1}$ norm is equivalent to minimizing the $l_{2,0}$ norm, as long as the matrix is row-sparse. With this assumption, we can relax (4) into the following convex optimization problem:

$$\min_{Q} \|X - XQX\|_F^2 + \alpha \|Q\|_{2,1} \tag{5}$$

where $\|Q\|_{2,1}$ is the matrix $l_{2,1}$ norm, which is the sum of the l_2 norm of each row of a matrix. Our objective function contains the non-smooth term $\alpha \|Q\|_{2,1}$, which makes it challenging to guarantee an optimal solution by directly differentiating the objective. We employ the alternating direction method of multipliers

(ADMM) to solve this problem [4]. We introduce a new variable $\widehat{Q}$ and express the problem as:

$$\min_{Q,\widehat{Q}} \|X - XQX\|_F^2 + \alpha \left\|\widehat{Q}\right\|_{2,1}$$
$$\text{s.t.: } Q = \widehat{Q} \tag{6}$$

The augmented Lagrangian function can be written as:

$$L(Q, \widehat{Q}, \lambda, \theta) = \|X - XQX\|_F^2 + \alpha \left\|\widehat{Q}\right\|_{2,1} + \langle \lambda, Q - \widehat{Q} \rangle + \frac{\theta}{2} \left\|Q - \widehat{Q}\right\|_F^2 \tag{7}$$

where $\lambda \in \Re^{n \times d}$ is the matrix of Lagrangian multipliers, $\theta \in \Re^{1 \times 1}$ is a constraint violation penalty parameter and $\langle ., . \rangle$ denotes the matrix inner product operator.

Updating Q: Considering the terms with Q in Eq. (7), we have the objective function to be minimized as:

$$L_Q = \|X - XQX\|_F^2 + \frac{\theta}{2} \left\|Q - \widehat{Q} + \frac{\lambda}{\theta}\right\|_F^2 \tag{8}$$

Setting $\frac{\partial L_Q}{\partial Q} = 0$, we get

$$2X^\top XQXX^\top + \theta Q = 2X^\top XX^\top + \theta \left(\widehat{Q} - \frac{\lambda}{\theta}\right) \tag{9}$$

Let $A = 2X^\top X$ and $B = 2X^\top XX^\top + \theta \left(\widehat{Q} - \frac{\lambda}{\theta}\right)$. Plugging back in Eq. (9) we get:

$$AQXX^\top + \theta Q = B \tag{10}$$

Note that both A and $XX^\top$ are symmetric and positive-semidefinite matrices. We can therefore perform an eigen decomposition of both these matrices as follows:

$$A = U\Sigma_1 U^\top, \quad XX^\top = V\Sigma_2 V^\top \tag{11}$$

where U and V are orthogonal matrices, Σ_1, Σ_2 are diagonal matrices. Plugging this back in Eq. (10) we get:

$$U\Sigma_1 U^\top QV\Sigma_2 V^\top + \theta Q = B \tag{12}$$

Multiplying both sides by $U^\top$ from left and V from right, we get:

$$\Sigma_1 U^\top QV\Sigma_2 + \theta U^\top QV = U^\top BV \tag{13}$$

Let $D = U^\top QV$. Plugging this back in Eq. (13) we get:

$$\Sigma_1 D\Sigma_2 + \theta D = U^\top BV \tag{14}$$

Equating both sides element by element, we get:

$$D_{ij} = \frac{(U^\top BV)_{ij}}{(\Sigma_1)_{ii}.(\Sigma_2)_{jj} + \theta}, \quad i = 1 \ldots n, j = 1 \ldots d \tag{15}$$

Now, $D = U^\top QV$. Thus, we can solve Q as:

$$Q = UDV^\top \tag{16}$$

Updating $\widehat{Q}$: From Eq. (7), considering the terms with $\widehat{Q}$, we have the objective function to be minimized as:

$$L_{\widehat{Q}} = \alpha \left\| \widehat{Q} \right\|_{2,1} + \frac{\theta}{2} \left\| Q - \widehat{Q} + \frac{\lambda}{\theta} \right\|_F^2 \tag{17}$$

Since the $l_{2,1}$ norm is the sum of the l_2 norms of each row of a matrix, we can decouple the minimization problem and solve for the matrix $\widehat{Q}$ row by row. The following lemma can be used to solve the above optimization problem [42].

Lemma 1. *For any $\kappa, \mu > 0$ and $g \in \Re^{n \times 1}$, the minimizer of*

$$\min_{t \in \Re^{n \times 1}} \kappa ||t||_2 + \frac{\mu}{2} ||t - g||_2^2$$

is given by

$$t = \begin{cases} \left(1 - \frac{\kappa}{\mu ||g||_2}\right) g & \text{if } ||g||_2 > \frac{\kappa}{\mu} \\ 0 & \text{if } ||g||_2 \leq \frac{\kappa}{\mu} \end{cases}$$

The solution to (17) is thus obtained as:

$$\widehat{Q}^i = \begin{cases} \left(1 - \frac{\alpha}{\theta ||s||_2}\right) s & \text{if } ||s||_2 > \frac{\alpha}{\theta} \\ 0 & \text{if } ||s||_2 \leq \frac{\alpha}{\theta} \end{cases} \tag{18}$$

where $s = \left(Q + \frac{\lambda}{\theta}\right)^i$, for $i = 1 \ldots n$ and M^i denotes the i^{th} row of matrix M.

Updating λ: The matrix λ can be updated using the following equation [4]:

$$\lambda \leftarrow \lambda + \theta(Q - \widehat{Q}) \tag{19}$$

The pseudo-code of our framework is outlined in Algorithm 1. As evident from the pseudo-code, our algorithm is independent of the underlying KD algorithm and the teacher-student network architectures, and can thus be seamlessly integrated across different teacher-student architectures and different applications. It is also very easy to implement.

3.3 Convergence Analysis

As evident from Algorithm 1, the sub-problems corresponding to $Q, \widehat{Q}$ and λ have closed form solutions. The convergence of Algorithm 1 can be obtained from the ADMM convergence results established in [4,10], which is formalized in the following theorem:

Theorem 1. *For given parameters α and θ, the iterates $(Q^i, \widehat{Q^i}, \lambda^i)$ converge to the solution $(Q^\star, \widehat{Q^\star}, \lambda^\star)$ where $(Q^\star, \widehat{Q^\star})$ is the global optimal solution of Problem (6).*

Please refer [4,10] for detailed proof.

Algorithm 1. The proposed active sample selection algorithm

Require: Unlabeled data matrix $X \in \Re^{d \times n}$, AL batch size k, parameters α, θ

Initialize: $Q = \widehat{Q} = \lambda = \{0\}^{n \times d}$
repeat
  Compute the matrices A and B, as shown in Equation (10)
  Perform eigen-decomposition and compute the matrices U, V, Σ_1 and Σ_2, as shown in Equation (11)
  Compute the matrix D element by element, as shown in Equation (15)
  Update the matrix Q, as shown in Equation (16)
  Update the matrix $\widehat{Q}$ row by row using Equation (18)
  Update the matrix λ using Equation (19)
until Convergence
Compute the l_2 norm of each row of the matrix Q. Identify the k rows with the highest l_2 norms and select the corresponding k unlabeled samples in the batch

3.4 Using Labeled Data for Active Sampling

Depending on the size of the initial training set L, it maybe desirable to use the uncertainty of the student model trained on L to select unlabeled samples from U, together with the method proposed in Algorithm 1. To this end, we compute an uncertainty vector $e \in \Re^{n \times 1}$ containing the prediction entropy of the student on all the unlabeled samples. Also, let q be the vector containing the l_2 norm of each row of the matrix Q, as detailed in Algorithm 1. We compute a weighted summation of these two vectors as follows:

$$v = \beta.q + (1 - \beta).e \tag{20}$$

where $0 \leq \beta \leq 1$ is a weight parameter governing the relative importance of the two terms. The k largest entries in the vector v are used to select the unlabeled samples in the batch. Note that our active sampling algorithm still remains independent of the network architecture and the KD algorithm, since the entropy vector can be computed merely from the probability values furnished by the student on the samples in U.

4 Experiments and Results

Datasets: Since knowledge distillation has been most extensively used in computer vision, we used three challenging and widely used computer vision datasets to study the performance of our framework: (i) **Fashion-MNIST (FMNIST)** [41]; (ii) **CIFAR-10** [17]; (iii) **CIFAR-100** [17]. We also studied the performance of our framework on two text mining datasets (detailed below).

Experimental Setup: Each dataset was divided into 4 subsets. The first subset was used to train the blackbox teacher model. The other subsets were used

as the initial labeled set L, unlabeled set U and test set to actively train the student. The number of samples in each subset for each dataset, together with the accuracy of the teacher model, are detailed in Table 1. Each algorithm selected a batch of k unlabeled samples in each AL iteration (where k is the query budget/batch size). The labels of the selected samples were obtained from the teacher and the newly labeled samples were added to the labeled training set. The student network was trained on the updated labeled set and its accuracy was computed on the test set. The process was continued for 15 iterations (taken as the stopping criterion in this work). All the results were averaged over 3 runs to rule out the effects of randomness. The vanilla knowledge distillation algorithm proposed by Hinton *et al.* [11] was used as the underlying KD algorithm for knowledge transfer.

The batch size k was taken as 300; the weight parameter β in Eq. (20) was taken as 0.5, the parameters α and θ in Eq. (7) were taken as 10^{-6} and 10^{-5} respectively. The matrices Q, $\widehat{Q}$ and λ were all initialized to 0. Following the convention in knowledge distillation research [9,39], the teacher was considered the oracle in our empirical studies; that is, the labels furnished by the teacher in response to the sample queries were considered the ground-truth and were used to train the student network. The labels of the samples in the initial training set L were also obtained from the teacher model.

Teacher Student Network Architectures: The architectures of the teacher and student networks for each dataset are also shown in Table 1. Such architectures have been used with these datasets in previous KD research [31,40].

Table 1. Details of our experimental setup. The columns respectively denote the dataset, number of samples used to train the teacher model, the generalization accuracy of the teacher, the number of samples in the initial training set L, the unlabeled set U, the test set, the network architecture of the teacher model and the student model.

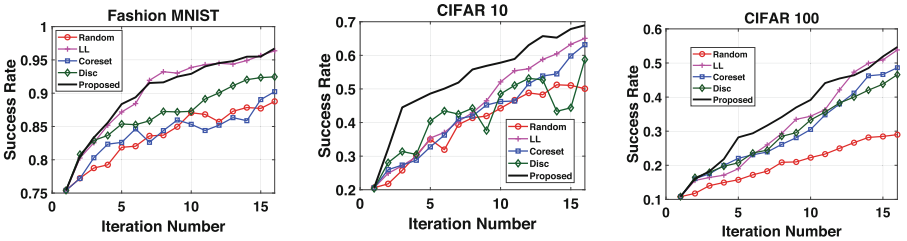
	Teacher Train	Teacher Acc.(%)	Initial Train	Unlabeled	Test	Teacher Arch	Student Arch
FMNIST	30,000	88.27	500	5,000	10,000	LeNet-5	LeNet-5-Half
CIFAR 10	30,000	75.34	500	5,000	10,000	AlexNet	AlexNet-Half
CIFAR 100	30,000	68.63	500	5,000	10,000	ResNet-34	ResNet-18
IMDB	25,000	84.28	500	5,000	6,500	BERT	DistilBERT
Tripadvisor	10,000	86.71	400	5,000	5,000	BERT	DistilBERT

Implementation Details: Please refer to the Supplemental File regarding the implementation details for training the teacher and student models.

Evaluation Metric: We used the *distillation success rate* [39] as the evaluation metric in this research. It computes the amount of knowledge the student network

distills from the teacher and is computed as the ratio between the student's classification accuracy and the teacher's accuracy on the test set. A high value of this metric denotes better performance.

Comparison Baselines: The following AL algorithms were used as comparison baselines in our work: (*i*) **Random Sampling**, where a batch of unlabeled samples was selected at random; (*ii*) **Learning Loss for Active Learning (LL)** [43]; (*iii*) **Coreset** [33]; and (*iv*) **Discriminative AL (Disc)** [8]. *LL* and *Disc* are widely used techniques in recent active learning research [28]; *Coreset* has been used in the context of AL for knowledge distillation [18] and was hence selected as a comparison baseline.

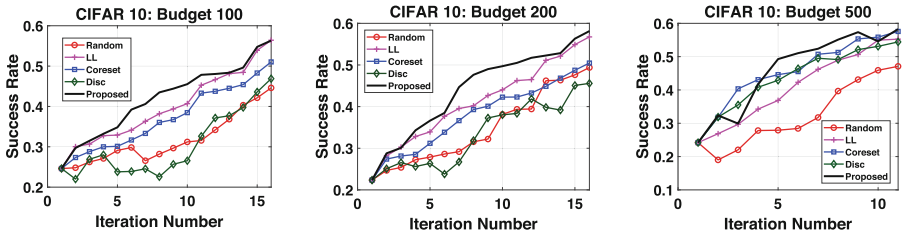


(a) Fashion MNIST (b) CIFAR 10 (c) CIFAR 100

Fig. 1. Active Learning performance comparison. The x-axis denotes the iteration number and the y-axis denotes the distillation success rate on the test set. Best viewed in color.

4.1 Active Learning Performance

The AL performance results are depicted in Fig. 1. In each figure, the x-axis denotes the iteration number, and the y-axis denotes the distillation success rate. *Random Sampling* does not produce good performance and achieves low distillation success rates with increasing size of the training set. The *Coreset* and *Disc* methods perform better than *Random Sampling* for the CIFAR-100 dataset. However, for CIFAR-10 and FMNIST datasets, the performance of *Coreset* is almost similar to *Random Sampling* and is sometimes inferior to *Random Sampling*; the *Disc* method mostly outperforms *Coreset* in the initial AL iterations. Both these observations are consistent with [8]. The *Learning Loss* method depicts the best performance among the baselines. The proposed method consistently depicts impressive performance and shows a steady growth in the distillation success rate with increasing label queries. It depicts the highest success rate for most of the AL iterations across all three datasets; it also attains the highest success rate at the end of 15 AL iterations, for all three datasets. Thus, by minimizing the reconstruction error, the proposed method is able to identify a



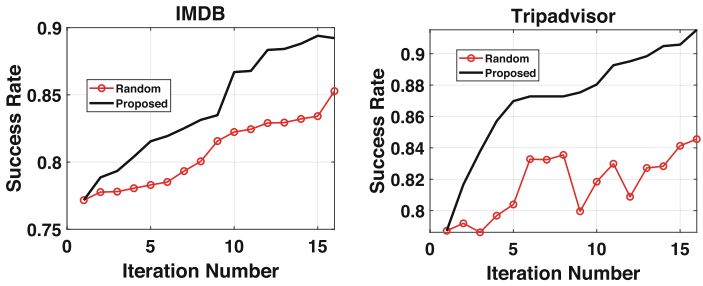
(a) Budget 100 (b) Budget 200 (c) Budget 500

Fig. 2. Study of query budget on the CIFAR 10 dataset. The results with budget 300 are presented in Fig. 1(b) and are not included here. Best viewed in color.

batch of exemplar samples which well-represent the unlabeled data. These results unanimously depict the potential of the proposed AL technique to actively distill knowledge from the teacher to the student network, when the number of label queries to the teacher is constrained by a given budget.

4.2 Study of Query Budget

The goal of this experiment was to study the effect of query budget on the AL performance. The results on the CIFAR-10 dataset for query budgets 100, 200 and 500 are presented in Fig. 2. The results depict a similar trend as Fig. 1. However, the performance of *Learning Loss* is not consistent across different budgets; it sometimes depicts marginally worse performance than *Coreset* and *Disc* (Fig. 2(c)). Our framework outperforms the baselines consistently across all query budgets. This shows the practical usefulness of our algorithm, as the query budget is often application specific and is dependent on the resources available for a given application.



(a) IMDB (b) Tripadvisor

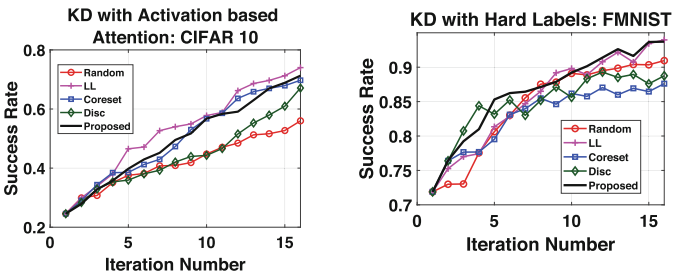
Fig. 3. Performance comparison on text mining datasets. Best viewed in color.

4.3 Performance on Text Mining

One of the useful features of our algorithm is its ability to generalize across multiple network architectures and hence, multiple applications. To demonstrate this, we studied its performance on text mining. We used the IMDB [21] and Tripadvisor [1] datasets for this experiment; the number of samples used are detailed in Table 1. The *BERT* model (based on a Transformer architecture) was used as the teacher and *DistilBERT* (a sub-network of BERT with half the number of layers) was used as the student for this experiment, similar to [31]. The KD algorithm was kept the same [11]. The baseline methods have largely been studied with CNN architectures for computer vision applications. The *Learning Loss* method, for instance, has mostly been applied with CNNs and its integration with transformer based architectures is not straightforward. We therefore compared our framework against *Random Sampling* in this study. The results are presented in Fig. 3. Our algorithm comprehensively outperforms *Random Sampling* and attains a much better success rate at the end of the AL iterations. For Tripadvisor, the improvement in the final success rate is about 7%. This corroborates the ability of our algorithm to seamlessly integrate across multiple teacher-student network architectures, and its ease of applicability in different domains.

4.4 Study of the Underlying KD Algorithm

Another useful feature of our framework is its independence of the underlying KD algorithm. In this experiment, we studied the performance of our framework in conjunction with the KD algorithm that uses activation based spatial attention as a mechanism of transferring knowledge from the teacher to the student network [47]. As in Table 1, we used AlexNet as the teacher and AlexNet-half as the student in this experiment. The results on the CIFAR-10 dataset are shown in Fig. 4(a). We also analyzed our framework with a KD algorithm where the teacher furnishes only hard labels (instead of the soft-label probabilities) in



(a) KD with Activation based Attention: CIFAR 10

(b) KD with Hard Labels: FMNIST

Fig. 4. Study of the underlying knowledge distillation (KD) algorithm on the CIFAR 10 and FMNIST datasets. Best viewed in color.

response to each sample query. We used LeNet-5 as the teacher and LeNet-5-half as the student, as in Table 1. The results on the FMNIST dataset are depicted in Fig. 4(b).

Our framework once again depicts competitive performance, demonstrating its generalizability across different KD algorithms (even when the teacher produces only hard labels). The *Learning Loss* method also depicts good performance.

We also conducted experiments to study the effect of the weight parameter β in Eq. (20) and the computation time of all the algorithms. These results are included in the Supplemental File due to space constraints.

5 Conclusion and Future Work

In this paper, we proposed a novel active learning algorithm for knowledge distillation applications. Such an algorithm can be immensely useful in training a lightweight student model to imitate a more complex teacher model, when the number of queries to the teacher cannot exceed a pre-specified budget. We posed the selection of exemplar training samples (to distill knowledge from the teacher to the student) as an NP-hard optimization problem and solved it using an iterative algorithm with global convergence. Our framework is independent of the underlying KD algorithm, as well as the architectures of the teacher and student networks, and can thus be seamlessly integrated across different KD applications. Our extensive empirical analyses verified the effectiveness of our framework for cost-effective blackbox knowledge distillation. Although validated on the KD application in this paper (as this is an under-explored research area), the proposed method is generic and can be used in any application to select an informative subset of samples from large amounts of data. As part of future research, we plan to study the performance of our framework on applications beyond computer vision and text mining, and also for regression and multi-label knowledge distillation.

Acknowledgment. This research was supported in part by the National Science Foundation under Grant Number: IIS-2143424 (NSF CAREER Award).

References

1. Alam, H., Ryu, W., Lee, S.: Joint multi-grain topic sentiment: modeling semantic aspects for online reviews. Inf. Sci. **339**, 206–223 (2016)
2. Ash, J., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: International Conference on Learning Representations (ICLR) (2020)
3. Boreshban, Y., Mirbostani, S., Ghassem-Sani, G., Mirroshandel, S., Amiriparian, S.: Improving question answering performance using knowledge distillation and active learning. arXiv:2109.12662 (2021)

4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **1**, 1–122 (2011)
5. Chen, H., Wang, Y., Xu, C., Xu, C., Tao, D.: Learning student networks via feature embedding. IEEE Trans. Neural Netw. Learn. Syst. (TNNLS) **32**(1), 25–35 (2021)
6. Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T.: Relation distillation networks for video object detection. In: IEEE International Conference on Computer Vision (ICCV) (2019)
7. Gan, C., Gong, B., Liu, K., Su, H., Guibas, L.: Geometry guided convolutional neural networks for self-supervised video representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. Gissin, D., Shalev-Shwartz, S.: Discriminative active learning. arXiv:1907.06347 (2019)
9. Gou, J., Yu, B., Maybank, S., Tao, D.: Knowledge distillation: a survey. Int. J. Comput. Vision (IJCV) **129**, 1789–1819 (2021)
10. He, B., Liao, L., Han, D., Hai, Y.: A new inexact alternating directions method for monotone variational inequalities. Math. Program. **92**(1), 103–118 (2002)
11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
12. Hu, P., Lipton, Z., Anandkumar, A., Ramanan, D.: Active learning with partial feedback. In: International Conference on Learning Representations (ICLR) (2019)
13. Hu, Z., Hou, W., Liu, X.: Deep batch active learning and knowledge distillation for person re-identification. IEEE Sens. J. **22**(14) (2022)
14. Huang, S., Chen, J., Mu, X., Zhou, Z.: Cost-effective active learning from diverse labelers. In: International Joint Conference on Artificial Intelligence (IJCAI) (2017)
15. Huang, Z., Wang, N.: Like what you like: knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
16. Kimura, A., Ghahramani, Z., Takeuchi, K., Iwata, T., Ueda, N.: Few-shot learning of neural networks from scratch by pseudo example optimization. In: British Machine Vision Conference (BMVC) (2018)
17. Krizhevsky, A.: Learning multiple layers of features from tiny images. In: Technical Report, University of Toronto (2009)
18. Kwak, B., Kim, Y., Kim, Y., Hwang, S., Yeo, J.: Trustal: trustworthy active learning using knowledge distillation. In: AAAI Conference on Artificial Intelligence (2022)
19. Lee, S., Song, B.: Graph-based knowledge distillation by multi-head attention network. In: British Machine Vision Conference (BMVC) (2019)
20. Li, C., Wang, X., Dong, W., Yan, J., Liu, Q., Zha, H.: Joint active learning with feature selection via cur matrix decomposition. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **41**(6), 1382–1396 (2019)
21. Maas, A., Daly, R., Pham, P., Huang, D., Ng, A., Potts, C.: Learning word vectors for sentiment analysis. In: Association for Computational Linguistics (ACL) (2011)
22. Micaelli, P., Storkey, A.: Zero-shot knowledge transfer via adversarial belief matching. In: Neural Information Processing Systems (NeurIPS) (2019)
23. Nayak, G., Mopuri, K., Shaj, V., Babu, R., Chakraborty, A.: Zero-shot knowledge distillation in deep networks. In: International Conference on Machine Learning (ICML) (2019)
24. Nie, F., Wang, H., Huang, H., Ding, C.: Early active learning via robust representation and structured sparsity. In: International Joint Conference on Artificial Intelligence (IJCAI) (2013)

25. Nie, X., Li, Y., Luo, L., Zhang, N., Feng, J.: Dynamic kernel distillation for efficient pose estimation in videos. In: IEEE International Conference on Computer Vision (ICCV) (2019)
26. Osmanbeyoglu, H., Wehner, J., Carbonell, J., Ganapathiraju, M.: Active machine learning for transmembrane helix prediction. BMC Bioinf. **11**(1) (2010)
27. Peng, F., Wang, C., Liu, J., Yang, Z.: Active learning for lane detection: a knowledge distillation approach. In: IEEE International Conference on Computer Vision (ICCV) (2021)
28. Ren, P., et al.: A survey of deep active learning. ACM Comput. Surv. **54**(9), 1–40 (2021)
29. Romero, A., Ballas, N., Kahou, S., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: hints for thin deep nets. In: International Conference on Learning Representations (ICLR) (2015)
30. Ruchansky, N., Crovella, M., Terzi, E.: Matrix completion with queries. In: ACM Conference on Knowledge Discovery and Data Mining (KDD) (2015)
31. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In: Neural Information Processing Systems (NeurIPS) Workshop (2019)
32. Sculley, D.: Online active learning methods for fast Label-Efficient spam filtering. In: Fourth Conference on Email and AntiSpam (2007)
33. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: International Conference on Learning Representations (ICLR) (2018)
34. Settles, B.: Active learning literature survey. In: Technical Report 1648, University of Wisconsin-Madison (2010)
35. Shi, L., Shen, Y.: Diversifying convex transductive experimental design for active learning. In: International Joint Conference on Artificial Intelligence (IJCAI) (2016)
36. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: IEEE International Conference on Computer Vision (ICCV) (2019)
37. Su, J., Tsai, Y., Sohn, K., Liu, B., Maji, S., Chandraker, M.: Active adversarial domain adaptation. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2020)
38. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. (JMLR) **2**, 45–66 (2001)
39. Wang, D., Li, Y., Wang, L., Gong, B.: Neural networks are more productive teachers than human raters: active mixup for data-efficient knowledge distillation from a blackbox model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
40. Wang, Z.: Zero-shot knowledge distillation from a decision-based black-box model. In: International Conference on Machine Learning (ICML) (2021)
41. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
42. Yang, J., Yin, W., Zhang, Y., Wang, Y.: A fast algorithm for edge preserving variational multichannel image restoration. SIAM J. Imag. Sci. **2**(2), 569–592 (2009)
43. Yoo, D., Kweon, I.: Learning loss for active learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
44. Yu, K., Bi, J., Tresp, V.: Active learning via transductive experimental design. In: International Conference on Machine Learning (ICML) (2006)
45. Yuan, F., et al.: Reinforced multi-teacher selection for knowledge distillation. In: AAAI Conference on Artificial Intelligence (2021)

46. Yun, J., Kim, B., Kim, J.: Weight decay scheduling and knowledge distillation for active learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12371, pp. 431–447. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58574-7_26
47. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations (ICLR) (2017)
48. Zhu, J., Bento, J.: Generative adversarial active learning. In: Workshop at Neural Information Processing Systems (NeurIPS-W) (2017)

Adabot: An Adaptive Trading Bot Using an Ensemble of Phase-Specific Few-Shot Learners to Adapt to the Changing Market Dynamics

Vishvajeet Upadhyay(✉) and Angshuman Paul

Indian Institute of Technology Jodhpur, Jodhpur, India
vishvajeet.pro@gmail.com

Abstract. In today's fast-paced and interconnected financial markets, the use of machine learning (ML) has become a game-changer in the realm of algorithmic trading. However, designing a consistently profitable algorithmic trading system (ATS) is challenging because of the dynamic and stochastic nature of the stock market. Often, stock markets undergo phase transitions where a market may suddenly change from a bullish trend to a bearish trend or vice versa. Most ML models fail to capture these phase changes. Consequently, inferior performance is observed from ATS during phase transitions. Moreover, the recent market data is usually small in quantity and often not sufficient to train an ATS model. We propose Adabot, an ensemble of phase-specific few-shot learners that can adapt to the changing market dynamics. Our models exploit synthetic data alongside real market data to train the ensemble. Adabot can adapt to a completely different market without any redesigning or training with extensive historical data thus reducing the deployment cycle, and can be used in markets that do not have sufficient historical data. When tested in four diverse markets, Adabot generated profits that were 30 to 90% greater than the respective benchmark returns over the test period. At the same time, Adabot significantly reduced the overall risk and did not degrade even after price shocks.

Keywords: Few-shot Learning · Synthetic data · Algorithmic Trading Systems · Ensemble

1 Introduction

An automated or algorithmic trading system (ATS), is a computer program that follows a defined set of instructions (an algorithm) to generate and place a trade in the Stock Exchange. The stock market is a dynamic stochastic environment

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_4.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 49–66, 2025.
https://doi.org/10.1007/978-3-031-78107-0_4

that constantly changes from one phase to another. A stock that was trending up (bullish phase) may suddenly crash and start trending downwards (bearish phase) or may start consolidating (sideways moving phase) due to various micro and macroeconomic factors. Designing an ATS that is consistently profitable in all the market phases is an extremely difficult task even with AI technology. Moreover, the ML models that are being used in ATS are trained conventionally and then used over a period of time without changing the parameters. Any phase change in the market may cause these models to degrade and produce very poor results.

Trading or investing in the stock market has many inherent difficulties. As per the Random Walk Hypothesis [10] the stock market exhibits a significant degree of randomness and noise, which can make short-term price movements unpredictable. The Efficient Market Hypothesis [9] suggests that stock prices already incorporate all available information. So, it is challenging to consistently find undervalued or overvalued stocks because market participants quickly react to news and information. Some of the other common factors contributing to the failure of ML models in stock trading are due to heavy reliance on historical data, overfitting, and rare "black swan" events which may not be present in the training data. Over time, AI models can experience a phenomenon known as "model drift," where their predictive accuracy diminishes as market conditions change.

This paper proposes Adabot, an adaptive Deep Trading Bot that overcomes these limitations. Adabot is an innovative model that adapts to the changing market dynamics with a small amount of real-time data by using few-shot learning techniques of synthetic data generation and Meta-Learning in an ensemble framework. The model was trained using only S&P 500 (USA) data and gave good results in that market. Using transfer learning, the model was then tested in three other diverse markets (Gold, Shanghai Composite (China), and MOEX (Russia)) without changing any parameters, the only change was that the test data was from different markets. Even in these markets, Adabot generated profits that were much greater than the respective benchmark returns over the test period while reducing the overall risk and did not degrade even after price shocks.

The main contributions of this paper are as follows:

- We propose an ATS model Adabot that can adapt to the phase transitions in the market.
- Our design consists of an ensemble of three expert models. While one model caters to the bullish phase of the market, the other two models handle the bearish, and consolidating phases of the market, respectively.
- Our model uses a small amount of real data alongside synthetic data to adapt to the changing market scenario.
- We propose a novel strategy for synthetic data generation.
- Our model significantly outperforms the benchmarks for various stock markets in the world in profitability as well as risk metrics.

2 Related Works

The most popular classical trading strategies are the trend following and mean reversion strategies, which are covered in detail in [6,19]. The focus of the majority of works applying machine learning techniques for algorithmic trading is on forecasting. The computation of optimal trading decisions is easy if the market behavior is known in advance with a high level of probability. Many DL techniques have already been investigated following this approach with good results [1]. In [3] a novel deep learning framework where wavelet transforms, stacked autoencoders, and LSTM are combined for stock price forecasting is presented. In this paper, the use of hierarchically extracted deep features using stacked autoencoders for stock price forecasting is introduced. Fischer and Krauss used LSTM networks for predicting directional movements for the constituent stocks of S&P 500 from 1992 until 2015 and concluded that the LSTM network could effectively extract meaningful information from the financial time series data [11]. A comparative study of Backpropagation, Kalman filter, SVM, and LSTM, to predict the stock price of nine selected companies is presented in [15]. LSTM was found to have the best performance in terms of prediction accuracy with low variance. A stock price prediction method that combines the phase-space reconstruction method for time series analysis and the LSTM model is presented in [25]. A comparative study of four models: Multilayer Perceptron, LSTM, and Convolutional Neural Network, and one attention-based neural network for the task of predicting the next day's index price according to the historical data on the dataset consists of the SP500 index, CSI300 index, and Nikkei225 index is presented in [12]. The results showed that the attention-based model had the best performance among the alternative models. A method for the prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system based upon the optimization of membership function by using the GA is proposed in [16].

The use of Reinforcement Learning models is also gaining popularity for algo trading. Moody and Saffell present methods for optimizing portfolios, asset allocations, and trading systems based on direct reinforcement. Investment decision-making is viewed as a stochastic control problem in this approach, and strategies are discovered directly. They introduced a recurrent RL algorithm for discovering new investment policies without the need to build forecasting models [18]. Deng et al. introduced the fuzzy recurrent deep neural network structure to obtain a technical-indicator-free trading system taking advantage of fuzzy learning to reduce the time series uncertainty. They also propose a task-aware backpropagation through time method to cope with the gradient vanishing issue in deep training [7]. One can also mention [4] which studied the application of the deep Q-learning algorithm for trading in foreign exchange markets. Théate and Ernst propose a novel DRL trading policy to maximize the resulting Sharpe ratio performance indicator on a broad range of stock markets. The training of the resulting reinforcement learning agent is entirely based on the generation of artificial trajectories from a limited set of stock market historical data [23].

The models described above are mostly static in nature. After their initial training, their parameters remain fixed throughout the trading process. This may lead to poor performance and model degradation when there is a phase change in the market or when they encounter data that is much different from the training data. These models do not address the following research questions:

- The financial markets are dynamic and have definite phases. How can we use this fact to design better ML trading Models? Our model addresses this gap by using an ensemble of phase-specific expert models.
- The most recent market data is vital yet it is not used in most of the existing models. How can we use it to improve our models? To overcome this gap, we use the latest market data to generate synthetic data and then use the combined data to fine-tune the expert models using few-shot learning.
- How can a model quickly adapt when deployed in a completely different Market? Our model uses transfer learning and few-shot learning to quickly adapt to a different market without the need for retraining.

3 Methods

Stock markets often undergo phase transitions. Market phases or regimes are defined as clusters of persistent market conditions. Determining the current regimes and the transition points is important as they have a deep impact on the relevance of investment models and the success of the trading strategies being used. In the Four-regime Markov switching model, the regimes are classified based on volatility into the tranquil, volatile, turbulent, or panic regimes [8]. Others maintain that market regimes are not directly observable and hard to determine and propose ML clustering algorithms for determining the regimes without explicitly classifying them rigidly [26]. For this study, we take a practical approach and define the regimes as in the common investment literature based on regimes defined by price trends: 'Bull market' (persistent uptrend), 'Bear market' (persistent downtrend), and 'Consolidating market' (sideways moving) [2]. To analyze these price trends we use Technical Analysis, which is a form of security analysis and provides a variety of methodologies for analyzing and forecasting the price trends by using past market data. This analysis is done using technical indicators which are, in essence, mathematical (statistical) calculations that analyze historical price and volume data. In this study, we use these technical indicators for categorizing and automatically segregating the historical data into market phases and also for creating additional features for training our models.

3.1 The Adabot

Most ML-based trading models fail to capture the phase transition in the markets. We propose Adabot which is designed to adapt dynamically to the changing market environment. Our model consists of two modules: the generation module

and the prediction module. To ensure that our model adapts to the changing market scenario, the generation module generates synthetic data using the present and past data. This data, along with real data is used for fine-tuning the prediction module using a meta-learning framework. It is difficult to have a single model that predicts well in all market phases. So, while designing the prediction module, we consider the fact that three major phases are usually observed in a stock market. These are bullish, bearish, and consolidating or flattish phases.

We design an ensemble of three models each catering to one of these three phases. This is done by training the basic models with data segregated for the three different market phases from the overall historical data. This segregation of data is done automatically using technical indicators. From these trained models, we choose three best-performing models that work well during the bull, bear, and consolidating phases of the markets respectively. All three models are deployed in the market. However, the capital allocation to these models is done in an ensemble way based on the weights of the models. The weights are calculated based on the characteristics of the present data. A block diagram of the proposed model is presented in Fig. 1. Next, we discuss the detailed architecture of our Adabot model.

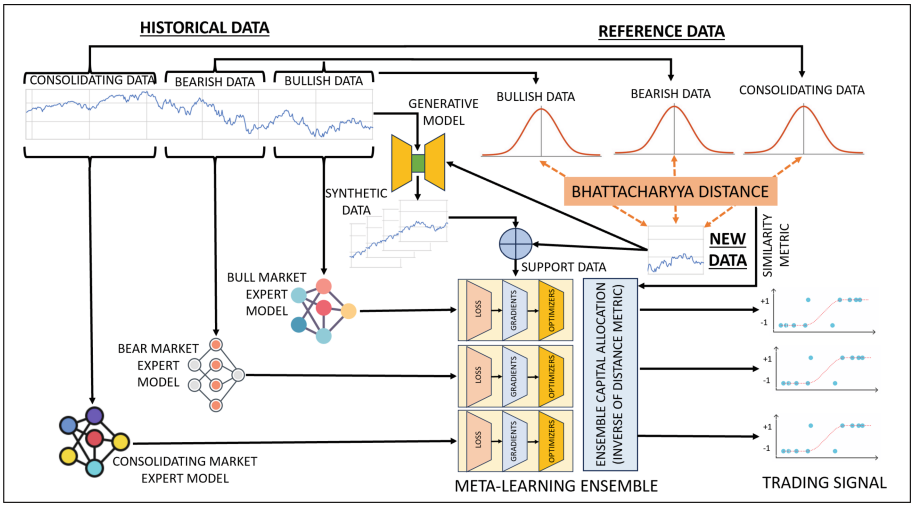


Fig. 1. The design and workflow of the AdaBot Model. We have three expert models for bullish, bearish, and consolidating phases of the market, respectively. These models are periodically fine-tuned using synthetic data combined with recent market data. We calculate weights for individual models. The weights are used for capital allocation.

3.2 The Generative Module

The generative module is meant to produce synthetic data to train the prediction module. The generative module should produce synthetic data that mimics the

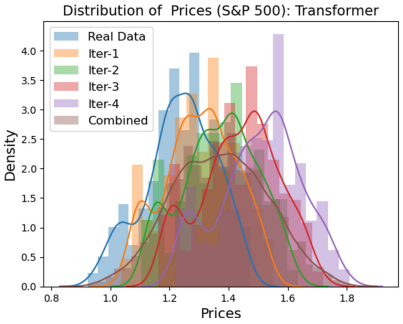


Fig. 2. Generated data distribution

Table 1. Effect of quantity of synthetic data used for meta-learning of Adabot (15-days cycle). The best results are obtained when we use 60 days of synthetic data. MDD is the Maximum Drawdown.

Quantity of Syn. Data: WGAN (Days)				
	30	45	60	75
Return:	18.6%	15.7%	28.7%	23.2%
MDD:	–28.5%	–29.2%	–18.4%	–17.6%
Quantity of Syn. Data: Transformer				
Return:	8.6%	12.2%	22.3%	18.8%
MDD:	–23.8%	–17.5%	–15.2%	–20.9%

present market scenario. To that end, we try two different generation approaches. In particular, we separately use a Wasserstein GAN (WGAN) based approach [17] and a transformer-based approach [22] and show the results for both cases.

WGAN-Based Approach. The WGAN is trained using stock market data consisting of the closing prices and various technical indicators to reduce the market noise and provide additional training features. We first train a variational autoencoder (VAE) to extract the distribution parameters of the stock market data. Subsequently, these distribution parameters are appended to the original stock market data to provide a distributional context to the dataset. Subsequently, this appended stock market dataset is used to train the WGAN. Once trained, the WGAN may produce realistic synthetic data for $(t+1)^{st}$ day, given previous t days' data as input. Using the synthetic data of $(t+1)^{st}$ day and the data from previous $(t-1)$ days, we get the synthetic data of $(t+2)^{nd}$ day. Thus, in an autoregressive manner, we use the WGAN to generate synthetic data.

Transformer-Based Approach. A transformer model is a neural network that learns context and thus meaning by tracking relationships in sequential data like the words in this sentence. First described in a 2017 paper by [24], transformers apply an evolving set of mathematical techniques, called attention or self-attention, to detect subtle ways even distant data elements in a series influence and depend on each other. We have used a model as proposed by [22] consisting of the first layer for time2vec conversion followed by three multi-head attention layers and two dense layers. The synthetic data is generated in an autoregressive manner similar to one described for WGAN.

Synthetic Data. We want to generate synthetic data that mimics the recent real data. However, the synthetic data should be diverse. It can be seen in Fig. 2 that in each iteration the generated data slightly diverges from the recent real

data and empirically we found that using four iterations (60 days of generated data) for few-shot fine-tuning of the models gave the best results (see Table 1).

3.3 The Prediction Module

The prediction module is composed of an ensemble of deep neural network models. We use two small architectures for the prediction models to minimize the system latency so that there is minimum lag during high-frequency trading. The first one is a stacked LSTM model with two layers of 50 units each. The second architecture is a hybrid model with three 1D convolution layers and a bidirectional GRU layer with 50 units (these models were empirically determined; for details see Supplementary Material). The ensemble consists of three models, each of which is trained to handle a specific market phase. Thus, we have a model each for bearish, bullish, and consolidating markets respectively. Given the t^{th} day's closing price of a stock, each model in the ensemble predicts whether to hold a long position (indicated by +1 signal) or a short position (indicated by –1 signal) for that stock on $(t+1)^{st}$ day.

For each of the bullish, bearish, and consolidating market scenarios, we create a reference distribution of closing prices (DCP) based on historical data. We use the same data that we had segregated using technical indicators and used previously for training the expert models in different market phases, as the reference DCP for the three phases (see sub-section 'Pre-training with Past Data'for details). We also calculate a distribution of the closing price for the present market using the last d days of data (empirically we found that using 15 days of latest data gave the best results). The present phase of the market for the stock being traded is identified by computing the Bhattacharyya distance between the DCP of the present market and that of the reference distributions. For a particular stock i, let the distance between the DCP of the present market and the bullish reference distribution is $r_{bu}(i)$. The distance of the DCP of this stock with respect to the bearish and the consolidating markets are $r_{be}(i)$ and $r_{co}(i)$, respectively. The weight assigned to the bullish model is

$$W_{bu} = S/r_{bu}(i), \tag{1}$$

where

$$S = 1/r_{bu}(i) + 1/r_{be}(i) + 1/r_{co}(i). \tag{2}$$

Similarly, we can calculate the weights for the bearish and consolidating models. Based on the weights, the trading capital for that particular stock is divided into bullish, bearish, and consolidating segments. Let these segments be n_{bu}, n_{be}, and n_{co}, respectively for bullish, bearish, and consolidating segments. The total capital $N = n_{bu} + n_{be} + n_{co}$. Also, assume that the signals generated by the bullish, bearish, and consolidating models are u_{bu}, u_{be}, and u_{co}, respectively. Then, the net capital to be traded on the next day in the direction of the dominant signal is:

$$C = n_{bu}u_{bu} + n_{be}u_{be} + n_{co}u_{co}. \tag{3}$$

It can be noticed that we are not using the full trading capital unless all three expert models agree on their prediction for the next trading day. This ensembling procedure helps Adabot to minimize the losses during price shocks in the market thus, the distribution of returns for Adabot is thin-tailed when compared to that of the benchmark (refer to Fig. 5). The prediction module is trained through two stages: Pre-training with past data, and few-shot fine-tuning with composite recent data using the Reptile algorithm [20] when deployed in the market.

Pre-training with Past Data. Each model in the ensemble is pre-trained separately with past data. We segregate the entire past training data into bullish, bearish, and consolidating data based on their trends automatically by using Moving Average (5,20) Cross-over technical indicator along with Relative Strength Indicator (RSI) set to 14 days. When the fast average (5 days) crosses over the slow average (20 days) from below and RSI > 45, we consider the subsequent days as bullish. If the fast average crosses the slow average from above and RSI > 45, we consider the subsequent days as bearish. For any cross-over, if RSI < 45, we consider the subsequent days as consolidating. Thereafter, we train the basic models with different hyperparameters using this segregated data. Out of these trained models, we choose the best-performing model in each of the bullish, bearish, and consolidating phases. These selected models are the expert models shown in Fig. 1 and are used for making the predictions.

Few-Shot Fine Tuning with Composite Recent Data. To quickly adapt to the current market scenario, we perform a few-shot fine-tuning of each model in the ensemble. For that, we use real stock data for the past 15 days (this period for the fine-tuning cycle was empirically determined, refer to Supplementary Material for details) and generate synthetic data for the next 60 days using the generative module. The fine-tuning is performed using the composite data consisting of the above real and synthetic data. Once the model is fine-tuned, it can make predictions for the next 15 days. Once the models are fine-tuned, the prediction module takes past t days' data as input and makes the prediction for $(t+1)^{st}$ day. This few-shot adaptation helps the model to adapt even to phase transitions in the market. Every time we need to fine-tune, we start with the original pre-trained model. The fine-tuning is performed using Reptile [20].

Inference. During inference to detect the current market phase, we feed in past 15 days' data to our Adabot model. Based on the DCP of this data, our model calculates the distances of the DCP of present data w.r.t. the DCPs of bullish, bearish, and consolidating markets' reference DCP as mentioned in Sect. 3.3. We use the Bhattacharyya distance to calculate the above distances between the distributions. The Bhattacharyya Distance measures the divergences between two distributions P and Q in the same domain χ as:

$$B_D(P, Q) = -\ln(B_C(P, Q)), \tag{4}$$

where B_C is the Bhattacharyya coefficient given by

$$B_C(P, Q) = \sum_{x \in \chi} \sqrt{P(x)Q(x)}. \tag{5}$$

Based on these distances, each of the bullish, bearish, and consolidating models is assigned a weight as per (1). Each of these models also provides a prediction on whether to hold a long position (+1) or a short position (–1) on the next day for this particular stock. Based on the weights, we allocate the trading funds to each of the models. Subsequently, using the predictions and the allocated fund, the net capital to be traded is obtained from (3).

3.4 Data Preprocessing and Financial Parameters

In this study, we use financial time series price data of the S&P 500 stock market index which is the most traded index in the US. It is a market capitalization-weighted price index that tracks the prices of the 500 largest listed companies in the US stock market. There is a lot of noise in the data hence training the models only on closing price data gives poor prediction accuracy. To overcome this, apart from the daily closing price we create and use additional technical indicators as features. We use Exponential Moving Average (EMA), Relative Strength Indicator (RSI), Moving Average Convergence Divergence (MACD), and Moving Standard Deviation (MSD) to create the additional features with various parameter settings. We have used 20 years of historical data with the test period from 04 Sep 2020 to 16 June 2023 (700 trading days). For the same test period, Adabot is also tested on three diverse market assets: Gold, Shanghai Composite Index (China), and MOEX (Russia). This testing is done without retraining the model to show the effectiveness of transfer learning and the few-shot training process. The performance of the model in various markets is assessed using the following financial parameters: Cumulative Return, Compounded Annual Growth Rate (CAGR), Sharpe ratio (excess average return earned per unit of volatility), Sortino ratio (excess average return earned per unit of downside volatility), Beta (expected move in an asset relative to movements in the benchmark), Jensen's Alpha (risk-adjusted excess return of an asset over benchmark returns) and Maximum Drawdown. We have taken the risk-free rate as zero for easy comparability of parameters across the markets.

3.5 Adabot: Overall Workflow

For the basic trading models (expert models in Fig. 1) we feed the end-of-the-day closing price and the models predict the closing price for the next day. If the predicted price is greater than the previous closing price(+1 signal) we take a long position- buy at open and sell at the close of Markets on the next day (Profit is made if prices rise). And if the predicted price is less than the previous closing price(-1 signal) we take a short position- sell at open and buy at the close of Markets on the next day (profit is made if prices fall).

The Adabot model is fine-tuned after every 15 days (the chosen period for fine-tuning is an adjustable hyper-parameter). The 15 days of new data is fed to the generating model in an auto-regressive way to generate 60 days of synthetic data. We combine these data points to get 75 days of data which is then used to fine-tune the model using a few-shot learning procedure (Reptile) with two iterations only in each cycle. This 15 days of new data is also used to calculate its similarity with the Bull, Bear, and Consolidating market reference data using Bhattacharyya Distance to measure the divergences between the distributions. The trading capital is allocated to the three trading models in proportion to the inverse of this similarity measure and changes in every cycle (refer to Algorithm 1).

Algorithm 1. Workflow for the Adabot Model

Data: n, N_{close} ▷ Cycle length, closing prices for recent n days
procedure TRAINING(n, N_{close}) ▷ Generate synthetic data and fine-tune each pre-trained expert model
    genr_input=[N_{close}] ▷ input data for generator
    syn_data=[N_{close}] ▷ Synthetic data
    for $iteration = 1, 2, \ldots,$ *auto_reg_cycles* **do**
        genr_output=predictions for genr_input
        genr_input=genr_output ▷ update genr_input
        Append genr_output to syn_data
    end for
    for *Expert_Models* **do** ▷ done in parallel for all expert models for two iterations each using syn_data
        Compute $W = SGD(\mathcal{L}_\tau, \theta, k)$
        Update $\theta \leftarrow \theta + \epsilon(W - \theta)/\alpha$
    end for
end procedure

Data: N_i, N_{close}, C ▷ Closing prices for i^{th} day, closing prices for n previous days, Trading capital
procedure DAILY TRADING(N_i, N_{close}, C)
    while Adabot is deployed in the Market **do**
        1. Get trading signal from each expert model
        2. Calculate Bhattacharya distance between N_{close} and Reference data
        3. Allott capital C to models based on inverse of distance (standardized)
        4. Calculate net trading signal and net capital to be deployed on opening of next trading day
    end while
    return Trading signal, Net Capital to be deployed
end procedure

4 Experiments and Results

4.1 Performances on Different Datasets

The Adabot model was initially trained using only the S&P 500 Historical data and tested on that Index. However, when deployed in other markets it was not again retrained with historical data, but with transfer learning and rapidly adapting with few-shot learning, Adabot gave good performance in other markets also as can be seen from the results.

S&P 500 (USA): Adabot was trained using S&P 500 data, and as can be seen in Fig. 3 and Table 2, against a benchmark (S&P 500 buy and hold) return of 29.9% the model gave a higher return of 57.6% and 36.6% with WGAN and Transformer generated synthetic data respectively for the test data. These returns were achieved with much lower risk compared to the benchmark which had a maximum drawdown of 20% compared to 7.6% and 12.4% for the model respectively. The Sharpe ratio for the model improved to 1.61 and 0.95 as compared to the benchmark Sharpe ratio of 0.58.

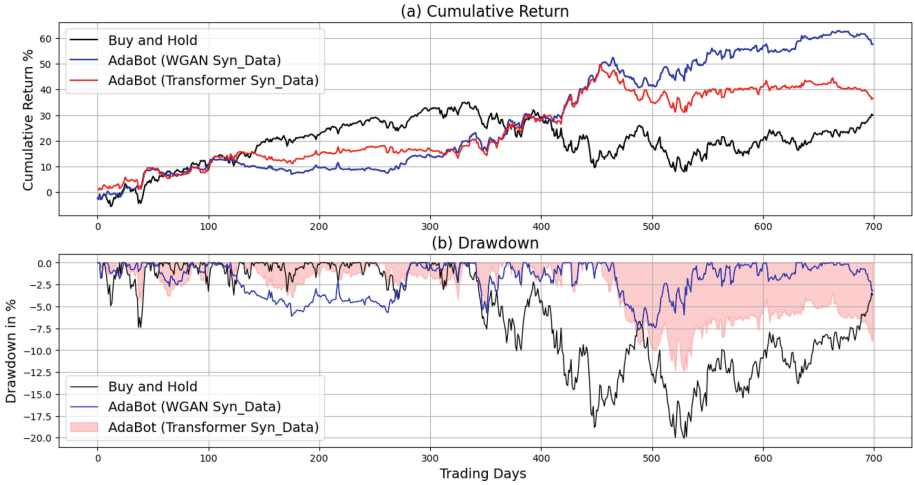


Fig. 3. Performance of AdaBot on S&P 500 (USA): Panel (a) shows that Adabot has more profitability even though there were periods of underperformance. In panel (b) it can be seen that the drawdowns for Adabot are reduced and have low correlation with Benchmark drawdowns thus providing strategic risk diversification.

Gold: During the test period, gold was in a consolidating phase, giving a return of only 5% (see Fig. 4(b) and Table 2). Adabot trained using S&P 500 data, was used without changing any parameters, the only change was that the new data

for fine-tuning and testing was from the Gold futures market. The model gave a similar result of 42.8% and 41.3% of overall cumulative return with WGAN and Transformer generated synthetic data for the test data. The risk profile also improved showing that Adabot has good generalizability using transfer learning.

Shanghai Composite (China): The Shanghai Composite index gave a return of only 2.8% with a maximum drawdown of 21.4% for the test period (see Fig. 4(c) and Table 2). Adabot, used without changing any parameters, gave returns of 51.8% and 46.7% with WGAN and Transformer generated synthetic data respectively for the test data with a maximum drawdown of 7.4% and 6.8%. The Sharpe ratio for the model improved to 1.41 and 1.27 as compared to the benchmark Sharpe ratio of 0.06.

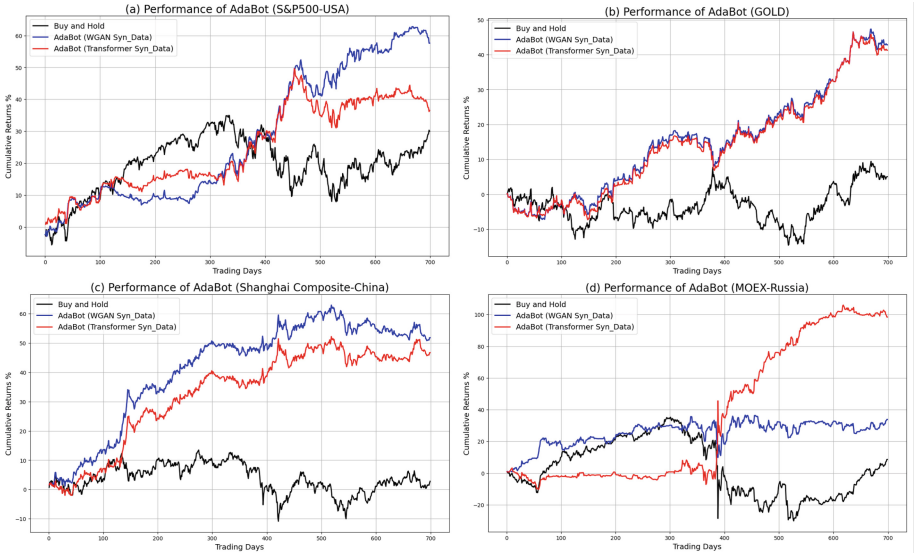


Fig. 4. Performance of Adabot on (a) S&P 500 (USA), (b) Gold, (c) Shanghai Composite (China) and (d) MOEX (Russia): The model was trained using only S&P 500 data yet shows good profitability and stable performance in diverse markets with transfer learning

MOEX (Russia): The Russian market was chosen to check the performance of Adabot during price shocks. During the test period, Russia had entered into a war with Ukraine. Russia's stock index (MOEX) dropped by almost 50% from its peak value and then started recovering slowly in the latter part to give an overall return of 8.6% at the end of the test period (see Fig. 4(d) and Table 2). Adabot gave a higher return of 33.8% and 98.4% with maximum drawdowns of 19% and

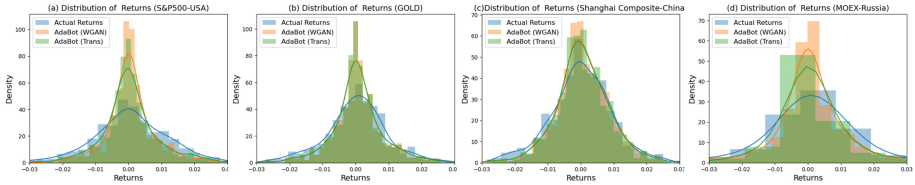


Fig. 5. Distribution of returns in various markets: financial assets are 'fat-tailed' (exhibit large leptokurtosis) which implies that extreme adverse events are more likely than that suggested by a normal distribution [21]. The figure shows that in all the markets, Adabot returns are 'thin-tailed' when compared to the respective Benchmarks and thus have reduced risk.

15.4% with WGAN and transformer-generated synthetic data respectively for the test data with improved risk metric. The results show the stability of the Adabot model during price shocks.

Performance in Other Markets. To generalize the performance, we show the results from four more markets: Bitcoin, DAX (Germany), TAIEX (Taiwan), and NIFTY (India) (see Fig. 6). In these markets, Adabot was tested after being trained on historical data for that specific asset. Better profitability and improved risk measures were observed in all these markets. When trading Bitcoin, Adabot was better on all parameters however, the returns were highly volatile and inconsistent just like the traded asset. When trading the other Indices, Adabot gave consistent results. There were periods of underperformance compared to the Benchmarks from which it recovered (refer to Table 5 in Supplementary Material for experimental results).

Risk Management. In Adabot risk control is inbuilt by ensuring that only part of the total investable capital is deployed in the market unless all the expert models in the ensemble agree to the predicted market direction. Also, all the standard risk management procedures can be integrated with our model. Using a stop loss placed at a distance of twice the 7-day Average True Range (ATR) technical indicator, and using a volatility threshold beyond which the model stops trading showed good results (refer to the section on Risk Management for a detailed discussion and Fig. 5 in Supplementary Material).

4.2 Comparision with SOTA Models

The most popular conventional models being used in ATS are SVM, XGBoost, and LSTM models whose comparative performance is shown in Fig. 7(a). These models are easy to configure and train so they are not only used as standalone trading models but also for feature extraction/selection [14]. We also show the comparative performance with three State of the Art Models viz: Informer [27] which uses Probabilistic Attention mechanism to select the "active" queries

Table 2. Performance of Adabot on S&P 500 (USA), Gold, Shanghai Composite (China), and MOEX (Russia). AdaBot1 shows results with WGAN generated data and AdaBot2 shows results with Transformer generated data.

	S&P 500			(USA) GOLD		
	Benchmark	AdaBot1	AdaBot2	Benchmark	AdaBot1	AdaBot2
Cumul. Return:	29.994%	57.594%	36.545%	5.060%	42.797%	41.256%
CAGR%:	10.794%	18.021%	12.016%	1.815%	13.858%	13.408%
Sharpe:	0.583	1.610	0.950	0.119	1.232	1.184
Sortino:	0.844	2.300	1.38	0.159	1.680	1.603
Beta:	1.000	0.077	0.147	1.000	0.069	0.071
Alpha:	0.000%	17.196%	10.439%	0.000%	13.732%	13.279%
MaxDrawdown:	–20.051%	–7.679%	–12.409%	–20.692%	–8.589%	–8.446%
	SHCOMP (China)			MOEX (Russia)		
	Benchmark	AdaBot1	AdaBot2	Benchmark	AdaBot1	AdaBot2
Cumul. Return:	2.805 %	51.796 %	46.674 %	8.600 %	33.826 %	98.408 %
CAGR%:	1.013 %	16.421 %	14.974 %	3.051 %	11.198 %	28.350 %
Sharpe:	0.066	1.413	1.271	0.092	0.588	1.162
Sortino:	0.092	2.193	1.986	0.088	0.670	1.635
Beta:	1.000	–0.041	–0.067	1.000	0.203	–0.484
Alpha:	0.000 %	16.462 %	15.042 %	0.000 %	10.578 %	29.828 %
MaxDrawdown:	–21.433 %	–7.461 %	–6.784 %	–48.390 %	–19.050 %	–15.378 %

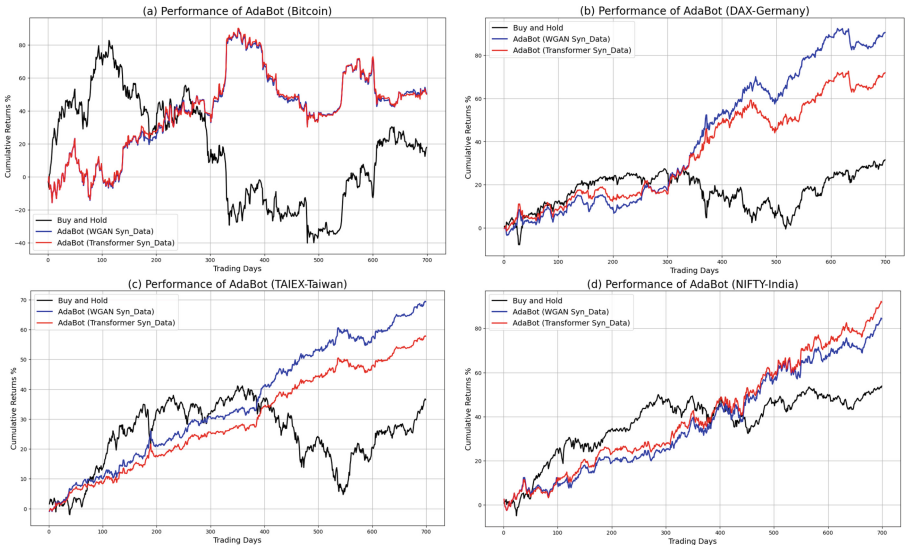


Fig. 6. Performance of Adabot in Other Markets: The Model was tested in four diverse markets after being trained with historical data for the respective asset

rather than the "lazy" queries and provides a sparse Transformer; NHITS [5] which uses Multi-rate input pooling, hierarchical interpolation, and backcast residual connections together to make forecasts; and TimeGPT-1 [13] which is a foundational generative pre-trained transformer based forecasting model, in Fig. 7(b) and Table 3. The hyperparameters for these models were tuned on validation data and then these models were used for uni-variate prediction with a forecast horizon of one day on the same test data for S&P 500 as before. As can be seen from the results, Adabot outperformed all these models (Fig. 7 and Table 3).

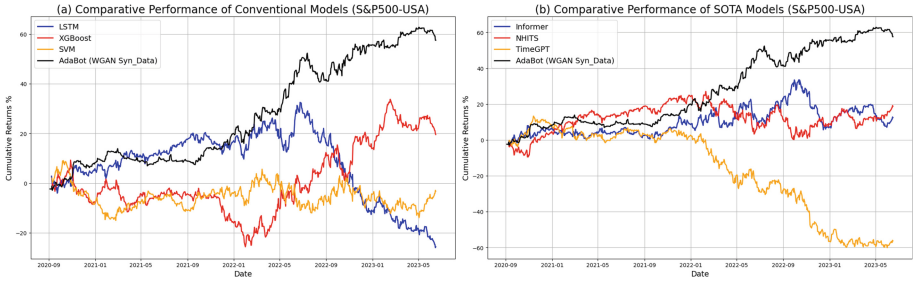


Fig. 7. Comparative Performance of Adabot vs (a) Conventional Models, (b) State of the Art Models

Table 3. Performance of Adabot compared to Conventional and SOTA models.

	AdaBot	SVM	XGboost	LSTM	Informer	NHITS	TimeGPT
Cumul. Return:	57.594%	–3.267%	19.554%	–8.337%	12.456%	18.844%	–56.274%
CAGR%:	18.021%	–1.203%	6.722%	–3.121%	4.369%	6.491%	–26.018%
Sharpe:	1.610	–0.064	0.381	–0.163	0.243	0.368	–1.100
Sortino:	2.300	–0.100	0.639	–0.243	0.365	0.543	–1.569
Beta:	0.077	0.050	–0.322	0.054	–0.295	0.846	–0.141
Alpha:	17.196%	–1.743%	10.198%	–3.704%	7.553%	–2.641%	–24.496%
MaxDrawdown:	–7.679%	–22.117%	–31.990%	–28.985%	–20.906%	–21.151%	–64.749%

4.3 Ablation Studies

Ablation studies were done to select the best architectures for the basic trading models, retraining cycle period, amount of synthetic data, etc. (refer to Supplementary Material).

5 Discussion and Future Work

The Adabot model offers many advantages over similar static models. Its design prevents model degradation and prolongs the useful life of the model. Adabot is more profitable and less risky due to reduced volatility and losses. It provides strategy diversification in the portfolio due to decreased correlation with the benchmark. Adabot is more reliable and has better explainability with the use of classical ensemble and meta-learning adaptability thus increasing user confidence in the model.

No model works in all the markets at all times. Even though our model has been well-tested in diverse markets, we cannot guarantee profitability in any market. Our model has many limitations and should be used cautiously after proper backtesting. In volatile markets (for example Bitcoin) our model may give inconsistent returns which may be as volatile as the traded asset. When there is a sharp bullish move our model may underperform the traded benchmark, the underperformance period may be as long as a year during which the investor may lose confidence in our model (for example NIFTY, TAIEX).

The model was tested in live trading however, a detailed analysis of the same is beyond the scope of this paper. Simulations were done to create various scenarios like low liquidity, execution slippages, and transaction costs of 0.5% (which is on the higher side). There was a drop in performance of 1 to 3% in the CAGR across all the markets, which is to be expected.

Adabot model not only adapts to the changing market dynamics in the same market but, as has been shown, it also adapts to completely different markets using Transfer learning. This can be very helpful for trading assets that do not have much historical data like Initial Public Offers of new companies or new crypto-currencies and is a step towards the generalizability of trading models. This model was constructed using bull, bear, and consolidation data classification. It can also be constructed using Markov regimes. We used a similarity metric to provide the ensemble weights, other criteria like previous performance of expert models or other portfolio management procedures like mean-variance optimization or their combinations can also be used as weights and are left for future work. The results in this paper are with unleveraged positions. The use of financial leverage for enhancing profits and the use of Adabot in a portfolio of multiple assets needs to be studied.

6 Conclusion

In this paper, we propose AdaBot, a deep Learning Model that adapts quickly to the changing market dynamics with a small amount of real-time data by using few-shot learning techniques of synthetic data generation and Meta-Learning in an ensemble framework. The model was trained only using S&P 500 data. The trained model showed excellent results in various other markets only with fine-tuning showing its generalizability and ability to adapt. Thus, overall, the proposed model can be helpful for increased profitability and reduced risk. In

the future, we will explore the possibility to use various potentially impactful conditions as input to our model. We will also apply our model for various other markets.

References

1. Arévalo, A., Niño, J., Hernández, G., Sandoval, J.: High-frequency trading strategy based on deep neural networks. In: Huang, D.-S., Han, K., Hussain, A. (eds.) ICIC 2016. LNCS (LNAI), vol. 9773, pp. 424–436. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42297-8_40
2. Arora, V., Das, S.: Day of the week effects in nse stock returns: an empirical study. Available at SSRN 1113332 (2007)
3. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE **12**(7), e0180944 (2017)
4. Carapuço, J., Neves, R., Horta, N.: Reinforcement learning applied to forex trading. Appl. Soft Comput. **73**, 783–794 (2018)
5. Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: Nhits: neural hierarchical interpolation for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 6989–6997 (2023)
6. Chan, E.P.: Quantitative Trading: How to Build Your Own Algorithmic Trading Business. John Wiley & Sons, Hoboken (2021)
7. Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q.: Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans. Neural Netw. Learn. Syst. **28**(3), 653–664 (2016)
8. Endovitsky, D.A., Korotkikh, V.V., Khripushin, D.A.: Equity risk and return across hidden market regimes. Risks **9**(11), 188 (2021)
9. Fama, E.F.: Efficient capital markets: a review of theory and empirical work. J. Financ. **25**(2), 383–417 (1970)
10. Fama, E.F.: Random walks in stock market prices. Financ. Anal. J. **51**(1), 75–80 (1995)
11. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. **270**(2), 654–669 (2018)
12. Gao, P., Zhang, R., Yang, X.: The application of stock index price prediction with neural network. Math. Comput. Appl. **25**(3), 53 (2020)
13. Garza, A., Mergenthaler-Canseco, M.: Timegpt-1. arXiv preprint arXiv:2310.03589 (2023)
14. Htun, H.H., Biehl, M., Petkov, N.: Survey of feature selection and extraction techniques for stock market prediction. Financ. Innov. **9**(1), 26 (2023)
15. Karmiani, D., Kazi, R., Nambisan, A., Shah, A., Kamble, V.: Comparison of predictive algorithms: backpropagation, svm, lstm and kalman filter for stock market. In: 2019 Amity International Conference on Artificial Intelligence (AICAI), pp. 228–234. IEEE (2019)
16. Kishikawa, Y., Tokinaga, S.: Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system optimized by the ga. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **83**(2), 357–366 (2000)
17. Lin, H., Chen, C., Huang, G., Jafari, A.: Stock price prediction using generative adversarial networks. J. Comp. Sci., 17–188 (2021)

18. Moody, J., Saffell, M.: Learning to trade via direct reinforcement. IEEE Trans. Neural Netw. **12**(4), 875–889 (2001)
19. Narang, R.K.: Inside the Black Box: A Simple Guide to Quantitative and High Frequency Trading, vol. 846. John Wiley & Sons, Hoboken (2013)
20. Nichol, A., Schulman, J.: Reptile: a scalable metalearning algorithm, **2**(3), 4 (2018). arXiv preprint arXiv:1803.02999
21. Rachev, Z., Rocheva-Iotovo, B., Stoyanov, S.: Capturing fat tails. Risk **23**(5), 72 (2010)
22. Schmidt, J.: Stock predictions with state-of-the-art transformer and time embeddings. Juillet (2020)
23. Théate, T., Ernst, D.: An application of deep reinforcement learning to algorithmic trading. Expert Syst. Appl. **173**, 114632 (2021)
24. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
25. Yu, P., Yan, X.: Stock price prediction based on deep neural networks. Neural Comput. Appl. **32**, 1609–1628 (2020)
26. Yuan, Y., Mitra, G.: Market regime identification using hidden markov models. Available at SSRN 3406068 (2016)
27. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)

Uncertainty in Ambiguity of Data

Hoang-Hiep Trinh(✉)

Hanoi University of Science and Technology, Hanoi, Vietnam
hiep.th212248m@sis.hust.edu.vn

Abstract. The current deep learning models have to deal with overconfidence issues due to overfitting and over-parameterization. Therefore, uncertainty estimation is an essential step in understanding and explaining the model's predictions to provide warnings and help humans trust the predictions of the AI model, especially in precise high-risk tasks, low regime data, and ambiguous datasets. This paper introduces an ambiguous dataset namely the YSH dataset and proposes a novel architecture, named the Diversity Ensemble of ResNet (DEResNet), to address the challenging uncertainty and diversity in the YSH dataset. Specifically, we use the Stein Variational Gradient Descent (SVGD) algorithm for uncertainty estimation. Then we utilize a diversity mutual information (MI) loss on target data to attack the diversity problem. DEResNet can address comprehensive uncertainty both in parameter space and output space. Extensive experiments demonstrate that our model outperforms the baseline method in the YSH dataset. Furthermore, our proposal also has the potential to learn the set of solutions for an ambiguous task and help find different solutions for optimization problems. Our code is available here https://github.com/trinh-hoang-hiep/DEResNet.

Keywords: uncertainty · distribution shifts · deep ensemble learning

1 Introduction

Current deep neural network (DNN) models face the problem of data scarcity when they require a large amount of diverse data with many contexts to train in order to give good performance. Recently, meta-learning in machine learning is also formulated as few-shot learning that can learn meaningful performance from such a few data points by leveraging meta-knowledge from past tasks. However, learning from a few samples inherently induces significant uncertainty and missing diversity. This is also a common phenomenon when we train a complex model such as a large and deep neural network in low-regime data by using only a few examples. Because there is a lack of diversity in the training data, it is not possible to fully describe the task's information, leading to having to a variety of solutions for an ambiguous task.

It is quite helpful to have an estimation of confidence in the prediction. One effective way to achieve this is through uncertainty estimation. Uncertainty estimation is related to the calibration quality or measurement of a prediction,

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 67–83, 2025.
https://doi.org/10.1007/978-3-031-78107-0_5

which involves repeatability and predictability. It should be noted that error, on the other hand, refers to the difference between a prediction and the ground truth. A well-calibrated model is capable of estimating both in-distribution (ID) and out-of-distribution (OOD) data. For ID data, the confidence level should be high in the correct predictions and low in the incorrect predictions and hard samples, meaning that the confidence level should be approximately equal to the true likelihood. For OOD data, the model should be able to detect that data is OOD and produce a high level of uncertainty in its prediction. This means that the confidence level should be approximately equal to a random guess. OOD detection is helpful for multitasking learning, semi-supervised learning, and continual learning. This uncertainty-driven learning can improve the model's efficiency, robustness, and fairness. In the classification task, a calibrated model is trained to classify data with confidence equal to the likelihood per class. This can be expressed as:

$$P(\hat{Y} = Y|\hat{P} = P) = p \forall p \in [0, 1] \tag{1}$$

where $\hat{Y}$ is the predicted class and $\hat{P}$ is the predicted probability. For instance, if 100 predictions have an average confidence of $\hat{P} = 0.7$, then 70 or 71 predictions should be accurate $P(\hat{Y} = Y) = 0.7$ or 0.71.

For ID, according to Kendall et al. [Kendall and Gal (2017)], uncertainty can be of two types: epistemic and aleatoric. Epistemic uncertainty is related to the model weights and can be decreased by collecting more data or can be estimated through Bayesian nets or MC dropout. On the other hand, aleatoric uncertainty is related to the data itself and cannot be decreased by collecting more data. Aleatoric uncertainty can be either homoscedastic or heteroscedastic and can be kept by teacher-student knowledge distillation or adversarial perturbation. Our solution uses Bayesian Stein Variational Gradient Descent as an ensemble architecture to capture both types of uncertainty. Aleatoric uncertainty is also known as the ambiguity problem in the data and the bias problem in the dataset, where the collected data is not diverse enough or the data obtained fluctuates greatly (e.g. objects far away from the camera). For better understanding, we present an example below.

The First Example: The Example: Figure 1 depicts the problem of uncertainty in a few-shot learning when the positive label includes *young, smiling, and hat-wearing* people, and the negative label includes *elderly, unhappy, and hatless* people. Models learn very well when training and can easily distinguish between two labels. But if, in the test set, there is a picture of an *elderly person smiling* and *wearing a hat*, which label will the model classify? This is the motivation for solving the aleatoric uncertainty problem, especially when we are not exploring data enough, considering attributes like feeling (smile and not smile), costume (hat-wearing or hatless), and age (young or elderly) as hidden attributes. We also can't know how many hidden attributes and their meaning are in the data. In addition, this problem is also related to the diversity in the data; for example, in the classification of cars and ships, the model can learn to classify water and road environments. Therefore, it is necessary to have images of cars in the water

or ships on land. Back to the problem of "old people laughing and wearing a hat", it is often difficult for models to capture the "hat" attribute because this attribute is present at times in negative labels; hence disentangling this hidden attribute will be more difficult than the other two attributes. In addition, the smile attribute is also very vague and can be confused with the "open mouth" attribute if the training data is not diverse. Moreover, pixels of a human mouth account for a relatively small region, the model will likely not be able to capture this attribute, leading to overconfidence when classifying images of *young people wearing hats but not smiling* as the positive label.

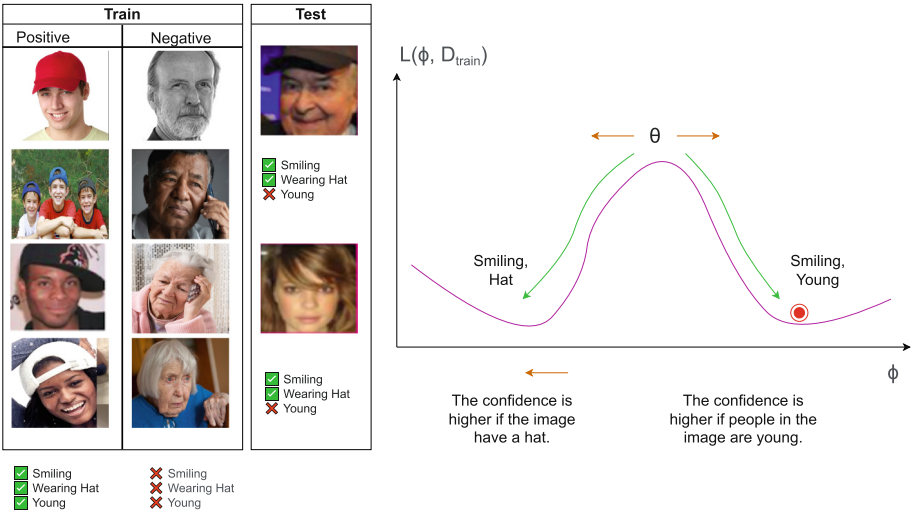


Fig. 1. The figure depicts the aleatoric uncertainty and disambiguation in the model. A particle will learn "smiling" and "hat" if its parameter θ moves to the left. A particle will learn "smiling" and "young" if the θ moves to the right. The red point indicates that a particle is approaching the target parameter and satisfies the requirement to classify the "smiling", and "young" attributes and ignore the "wearing hat" attribute. source: adapted from: http://cs330.stanford.edu/fall2019/slides/cs330_bayesian_metalearning.pdf (Color figure online)

To address this problem, techniques such as invariant feature learning and orthogonal feature learning with many headers in representation and output space can be used. Besides processing by those techniques, bias in the model can be processed by initializing an ensemble in weight space. Currently, DEResNet uses SVGD [Wang et al. (2019)], which is a Bayesian-based method. This method maintains an ensemble with diverse and same architecture members, also known as particles. Particles will have repulsive force on each other in parameter space, so SVGD can capture a wider range of attributes and make diversity in the prediction in parameter space. By applying the SVGD method, we expect the particles to fall into optimal regions on this non-convex function. A particle will

learn "smiling" and "hat" if its parameter θ moves to the left. A particle will learn "smiling" and "young" if the θ moves to the right. By using the constraint that the weights must be different in the weight space, we want to bring about the high possibility that the model will have particles that push each other away from the local optimum, and even if a particle falls to the saddle point, also means other particles will be avoided and gives the best prediction performance with multi-view. This method is different from the representation space methods; it reduces the effect of initialization on the model weights.

In summary, the paper proposes a new dataset and new problem to deal with the ambiguity in the task that occurs when the collected data is little. The ambiguity problem will disentangle the features and try to learn the variety using particles, each of which will capture a hidden attribute or a view of the data. We then remove some particles and evaluate whether our ensemble's prediction is still sensitive to that hidden attribute. The new dataset is formulated as the example by a dataset called the YSH dataset. The main contributions of this paper are as follows:

- We propose a new dataset to evaluate the ability to handle the ambiguity in the task.
- We propose the DEResNet utilize SVGD and a training method to find sub-models to provide a set of solutions for the ambiguous dataset. We use a target set containing ambiguous data to prove that there will be some sub-models that prefer one attribute over another and then find a set of sub-models that gives the best results on the target set. This paper is the first to combine Bayesian SVGD in parameter space and a diversity mutual information loss in output space for handling uncertainty and solving the diversity problem. This is also the first study to replace the mutual information loss in the headers of DivDis with particles with the same architecture of SVGD to diversify predictions.
- Extensive experiments on the YSH datasets demonstrate that our DEResNet outperforms the baseline method. It can overcome the saddle point to reach the global optimal and produce a set of solutions with multiple interpretations of an ambiguous task.

2 Related Work

2.1 Uncertainty Estimation

DNNs have great success in the scientific and engineering fields of computer vision, natural language processing, and generating precise predictions. However, DNNs can be incorrect and overconfident, when fine-tuning a large and deep model on a small dataset due to over-parameterization and overfitting. Therefore, uncertainty estimation is receiving more attention in the deep learning community. Naeini et al. [Naeini et al. (2015)] suggest using temperature scaling as a post-processing technique to improve prediction smoothing. However, it has a limitation as it is unable to detect the OOD data. In contrast, Pereyra

et al. [Pereyra et al. (2017)] introduced an entropy regularization method that maximizes the entropy of false predictions, but it can negatively affect the high valid confidence of correct predictions. Blundell et al. [Blundell et al. (2015)] proposed Bayes by Backprop (BBB) to train Bayesian neural networks. However, BBB's prior distribution may be difficult to find, and its uncertainty estimate can be unreliable due to over-parametrization. Similarly, MCDropout [Gal and Ghahramani (2015)] approximates Bayesian nets by generating a Monte Carlo distribution of predictions during test time, estimating the prediction's epistemic uncertainty.

For aleatoric uncertainty, Zhang et al. [Zhang et al. (2023)] analyzed the risk bounds on aleatoric of estimators that use Mean squared error (MSE) loss and Negative log likelihood (NLL) loss. They found that a method based on moment matching such as the MSE empirical risk minimizer has an advantage in a high variance regime. Another study by Fan et al. [Fan et al. (2021)] provided necessary and satisfying conditions for the Wasserstein gradient flow (WGF) system to be a moment-matching property. Additionally, Wang et al. [Wang et al. (2022)] demonstrated SVGD as a method to address the WGF problem.

2.2 Diversity and Disambiguation

Studies on diversity in the prediction of DNNs and disambiguation in ambiguous tasks are still relatively few, with limited results, and are still open to many challenges. Luca Scimeca et al. [Scimeca et al. (2022)] found that DNNs frequently depend on simple-to-learn distinguishing features known as shortcut learning. They observed that solutions that prioritize easy-to-learn attributes tend to reach flat minima on the loss surface, and these solutions are more common in the parameter space. They have proposed a way to evaluate the abundance of certain attributes via Kolmogorov complexity: "Solutions corresponding to Kolmogorov-simple cues are abundant in the parameter space and are thus preferred by DNNs". Specifically, models are often biased to learn simple cues, such as color and ethnicity that may cause negative societal impacts.

Yoonho Lee et al. [Lee et al. (2022)] researched shifts between source and target distributions, where the source data falls short of conveying the desired behavior on target inputs. They proposed DivDis with two stages for resolving ambiguity in data and addressing the distribution shifts problem. First, DivDis learns diverse hypotheses with low source loss but different predictions on target inputs to form a set of solutions. In the second stage, they disambiguated by selecting one of the discovered functions.

Our model addresses both problems of uncertainty and diversity. Specifically, DEResNet uses SVGD which is a Bayesian-based ensemble method. For epistemic uncertainty, with samples that are difficult to model, our DEResNet uses a deep ensemble to measure the disagreement among particles. To calculate the epistemic uncertainty, we need to find the entropy of the probability vector generated by particles. As for aleatoric uncertainty, this type of uncertainty cannot be reduced by more data as it is inherent to the data, for example, an object far from the camera or object boundaries. DERestNet uses SVGD to

learn the attributes of multi-view data, so it can de-bias and learn depth information to increase uncertainty for objects far from the camera. In the theory of analysis, SVGD is a moment matching-based method so DEResNet uses SVGD to determine uncertainty in high-variance regimes in the data. Additionally, to instruct particles that learn different attributes, our model takes advantage of mutual information loss in DivDis and performs an exhaustive search algorithm to disambiguate and select the best solution in the set of solutions.

Fig. 2. The figure depicts the YSH dataset.

3 Creating a Dataset for Ambiguity Tasks

The emergence and success of ChatGPT and foundation models have led to the need to refine models with only a few samples by a few shot prompt engineering. However, with little data, it is not possible to fully describe the task, which leads to ambiguity in the task. To address this problem, it is often necessary to use additional information in order to supplement the task. Another way is for the model to provide a set of solutions. Our proposed dataset contains a target set that supports the model to capture multi-views of data. Ambiguous tasks will lead to overconfidence and a lack of robustness in classification. With this in mind, our goals for developing a dataset for ambiguity tasks are: to provide a new challenging problem and to spark novel ideas.

We formulated the example by a dataset called the YSH dataset, with positive labels including only pictures of *young people smiling and wearing hats* and negative labels only including images of *old people not smiling and not wearing hats* as the training set and testing set of the source distribution. There are 3 additional training and testing sets in the target distribution as follows:

- No smile: The positive label includes *young people wearing hats* but *not smiling*, and negative labels include *old people not wearing hats* but *smiling*. We'd like to be able to have a particle that predicts a "smile" attribute that belongs to the positive label. To evaluate it, we remove that module from our ensemble and hope the resulting sub-model will achieve the greatest performance on this set.
- No hat: The positive label includes *young people smiling* but *not wearing hats*, and the negative label includes *old people not smiling* but *wearing a hat*. We'd like to be able to remove the positive-label hat distinguishing module from our ensemble. So that the sub-model with the remaining paricles obtained will achieve the greatest performance on this set.
- Not young: The positive label includes the *elderly wearing a hat and smiling*, and the negative label includes the *young person not wearing a hat and not smiling*. We would like to be able to remove the module that distinguishes young attributes as positive labels from our ensemble.

Exemplars of YSH are shown in Fig. 2. The dataset contains 1041 images collected from the Internet via the Google search engine with some keywords such as "young smiled man wearing a hat", "unhappy man old wearing a hat", "smiled child wearing a hat", "happy woman old wearing a hat",... It contains 316 images in source distribution and 98 images in target distribution. The source distribution is split into 250 images (129 negative labels and 121 positive labels) for training and 66 images (31 negative labels and 35 positive labels) for testing. The target distribution is split into the target training set and the target testing set. It has three sub-testing sets: 53 images for no smile training and 205 images for no smile testing, 27 images for no hat training, 207 images for no hat testing, and 18 images for not young training and 215 images for not young testing.

4 Proposed Method

4.1 General Framework

Our framework is built on diversity training for training the DivDis architecture [Lee et al. (2022)], which is a domain adaptation framework for training the DEResNet architecture. DeResNet is built on the ResNet backbone and SVGD algorithm for diversifying the ensemble. Our proposal improves uncertainty and ambiguity estimation with a flexible training strategy by combining SVGD, and MI loss to diversify and disambiguate for any task. Our approach with SVGD and MI loss can be easily applied to any DNNs and is easy to plug and play.

In general, we approach the problem in a distribution shift with the source distribution being labeled images, and the target distribution known as OOD data being unlabeled images. Our training strategy consists of 2 stages:

- Stage 1 - Uncertainty estimation and Diversity: The model will train on the source and target distribution in parallel. Specifically, after the model learns a batch of (x_s, y_s) on the source distribution, it tries to maximize the uncertainty of the target distribution's predictions by making the model's predictions close to the true data distribution of labels (with the YSH dataset the true distribution of positive and negative labels is (0.5, 0.5)). At the end of stage 1, we obtain a well-calibrated model, accurate predictions on the source distribution, and low confidence in predicting images belonging to the target distribution.
- Stage 2 - Disambiguate in target distribution prediction: We use MI loss and regularizations to force headers also known as particles to make different predictions. At the end of phase 2, we have a set of solutions, and hopefully, each particle will learn a view of the data. Finally, we gradually remove particles from the ensemble using an exhaustive search to find the sub-ensemble that best fits each data view.

Figure 3 shows the overall framework of our proposal, where x_s, y_s are the source image and source label. x_t is the image in the target distribution. DEResNet is our model trained in two stages. L_{s_stage1}, L_{s_stage2}, L_{t_stage1}, L_{t_stage2} are loss functions using at stage 1 and stage 2 on the source data, and on the target data respectively. We will present these loss functions in more detail in the "Training Strategies" section.

4.2 DEResNet Architecture

To address uncertainty estimation, we employ the well-known SVGD architecture [Wang et al. (2019)] on the ResNet backbone to form DEResNet. Figure 4 shows the DEResNet architecture.

ResNet Baseline. He et al. [He et al. (2016)] introduced the ResNet architecture. ResNet models consist of residual blocks, which include stacked convolutional layers and a skip connection from the first to the last layer. Skip connections can train very deep neural networks by mitigating the vanishing gradient problem. The authors proposed five ResNet variants of different depths: ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. For our DEResNet model, we chose ResNet-50 as the backbone. The ResNet model belongs to the family of deterministic models, in addition, due to the large number of parameters, this model is overconfident in the predictions of the target distribution.

SVGD in DEResNet. Unlike the original ResNet baseline suggested here [1], we use both an ensemble and a Bayesian, namely the SVGD method. It uses particles to approximate and minimize KL divergence to fit the true posterior

[1] https://colab.research.google.com/github/bentrevett/pytorch-image-classification/blob/master/5_resnet.ipynb.

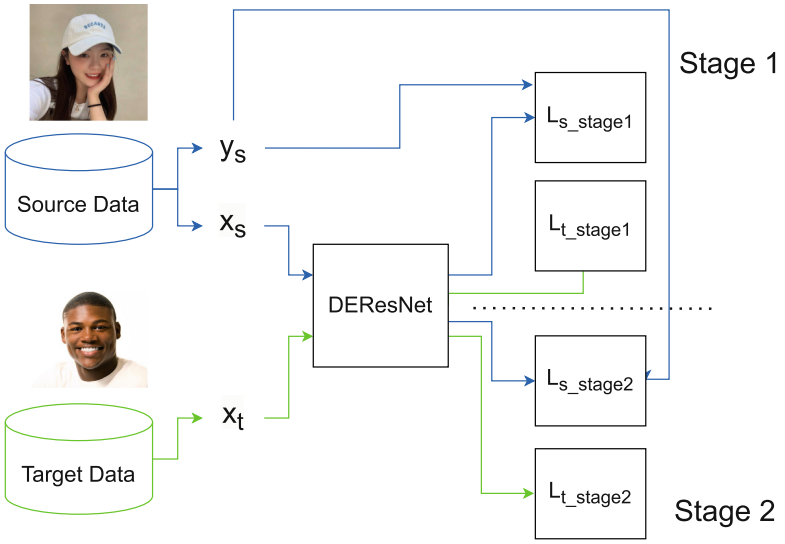


Fig. 3. The overall framework for training the DEResNet architecture.

distribution $P(w|D)$ with w as the weight of the model, and D as the data. Since $P(w|D)$ is intractable, we estimate a maximum a posteriori (MAP), but instead of using MCMC, we use SVGD due to the assumption of the non-convexity of the MAP estimation to generate and independently train M-models samples from the target distribution. Using only one particle, it reduces to MAP. With more particles, it becomes a fully Bayesian approach, similar to many MAPs in the Bayes inference step. SVGD repels particles from high-density regions of the target distribution and uses them as regularizers to prevent ensemble collapse. Therefore, it effectively has the potential in low regime data.

In this paper, we apply the SVGD algorithm in the ResNet-50 model. First, we generate an ensemble of M particles from a Gaussian distribution with mean μ as the parameter of the ResNet-50 model and standard deviation σ. We initialize an SVGD pre-conditioner that uses a kernel RBF function. This pre-conditioner will help to learn the second term in Eq. 3. The final prediction is the average of the particle predictions in the ensemble:

$$P(y|x) = E_{P(w|D)}[P(y|x,w)] = \frac{1}{M}\sum_{i=1}^{i=M}[P(y|x,w_i)] \tag{2}$$

When training the model, we compute the gradient for each weight of the ensemble and then use the SVGD pre-conditioner to calculate the second term in Eq. 3, which is added to the gradient. This phase is shown in Fig. 4 in a purple ellipse denoted SVGD.step(). The pre-conditioner will update the weight following the

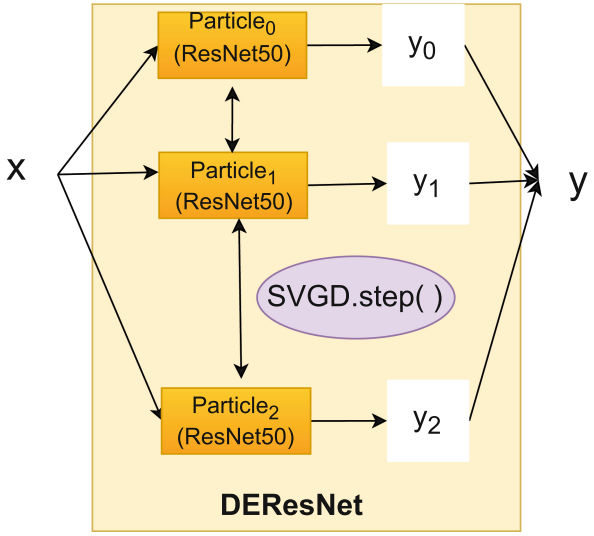


Fig. 4. The DEResNet architecture.

formula:

$$\phi(w_t) = \frac{1}{M}\sum_{j=1}^{M}[k(w_t{}^j, w_t) \bigtriangledown_{w_t{}^j} logp(w_t{}^j) + \bigtriangledown_{w_t{}^j} k(w_t{}^j, w_t)] \tag{3}$$

corresponding to the first term is the likelihood term and the second term is the repulsive term, where M is the number of particles, k is RBF kernel, t is iteration, and ϕ is the updated rule $w_{t+1} = w_t + \epsilon_t \phi(w_t)$ with ϵ is learning rate.

4.3 Training Strategies

Training Strategy in Stage 1. In stage 1, we train on the source distribution, then estimate uncertainty and calibrate on the target distribution. On the source distribution, we use L_{s_stage1} as cross-entropy loss for learning and give the best performance on the source dataset. Because the data on the source distribution are the same in 3 hidden attributes: "smile", "hat" and "young", we want each particle prediction along with the ensemble's final prediction to fit the training data. So we have the following formula Eq. 4:

$$L_{s\_stage1} = CE(\hat{y_s}, y_s) + \sum_{i=1}^{i=M}[CE(\hat{y_s}, y_{s,i})] \tag{4}$$

where CE is cross-entropy loss, x_s is the image of source set, y_s is the ensemble's final prediction and $y_{s,i}$ is the prediction of $particle_i$, $\hat{y_s}$ is groundtruth.

For the target distribution, DEResNet will minimize the confidence of the predictions, that is, maximize the uncertainty using L_{t_stage1}.

$$L_{t\_stage1} = D(p(y_t)||p(y)) = (y_t - 0.5)^2 \tag{5}$$

where $p(y)$ is the label distribution in the source dataset, we chose squared Euclid distance for distance function D and y_t is the final prediction of image x_t in the target set.

This loss function has the effect of encouraging particles to learn a wider range of hidden attributes. Because if a model does not learn diversely, it cannot capture all the hidden attributes in the target set. For example, in the YSH dataset, if the model fits too well with the source set and cannot capture the "smile" attribute, then when evaluated on the No Smile test set, it will still have high accuracy but be overconfident. Because the model only exploits the "hat" and "young" attributes, when an image of a "young person without a hat and sad" appears, the model ignores the "smiling" attribute and predicts that the image is correct with too much confidence. DEResNet uses L_{t_stage1} to calibrate the predictions of the target distribution to avoid overconfidence, and thus give the model a second chance to learn more diverse attributes in stage 2.

Training Strategy in Stage 2. In stage 2, we ask particles to make different predictions to disambiguate the task and hope each particle will learn a view of the data while maintaining good source accuracy. So we continue to use the source set and train on it with the same loss function as in stage 1. The formula of L_{s_stage2} is the same as Eq. 4.

For the target distribution, we use mutual information loss known as MI loss and the regularization of the first stage to force the particles to come up with a set of different solutions. MI loss gets ideas from the DivDis architecture applied to headers of the same model. However, these headers have a perfect similarity to the particles in SVGD. So we tweak the MI loss to minimize the mutual information between each pair of particle's predictions and the label distribution of the source data:

$$L_{MI}(y_{t,i}, y_{t,j}) = D_{KL}(p(y_{t,i}, y_{t,j})||p(y_{t,i}) \otimes p(y_{t,j})) \tag{6}$$

where $D_{KL}(.||.)$ is the KL divergence and $y_{t,i}$ is the label distribution in the source data $p(y) = (0.5, 0.5)$ or the prediction of particle i for x_t as the target data.

The final loss function of the target set in stage 2 is as follows:

$$L_{t\_stage2} = \lambda L_{MI} + \beta||B^T B||_1 - \gamma y_t[\underset{t}{\text{argmax}}(||y_t||_2)] \tag{7}$$

where λ, β, γ is hyperparameter, $||B^T B||_1$ is the L1-norm of the dot product between two features of two particles, we use it to minimize the covariance matrix between the features of the particles. Then $-\gamma y_t[\text{argmax}_t(||y_t||_2)]$ is the term to force the diversity particle's predictions by maximizing the prediction

of the most confident particle and we want the samples to be free to "cluster" into one of the particles that are most likely to maximize the confidence of the prediction. It makes each prediction of particles at the target data avoid random guesses.

5 Experiments

5.1 Scenario of Experiments, Datasets and Metrics

Datasets and Scenario of Experiments. We use the YSH dataset for our experiments. In addition, we perform image data augmentation through random rotation, random flip, center crop, and normalization operators. We compare our method with the ResNet baseline [He et al. (2016)] without the target set and the ResNet50_DivDis model [Lee et al. (2022)] which is trained with multi-head classification. In the ResNet50_DivDis model, we keep the hyperparameters unchanged and use cross-entropy for regularization as in the source code provided by the authors. For a fair comparison with our method, we set the number of headers to 3 equal to the number of seeds, meaning the output of layer"fc" is 3 times the number of classes. The training method remains the same, i.e. supervised training on the source set and applying MI loss to the target set.

Evaluation Metrics. For the classification task, we utilize accuracy metrics to evaluate the performance and denoted as Acc in Eq. 8. To evaluate the uncertainty estimation, we use the ratio of the number of over-confidence predictions to the number of total predictions in the target set $OverConf$. Over-confidence predictions are defined as incorrect predictions with a confidence level greater than a threshold τ ($\tau = 0.6$ in our experiments) ($P(\hat{Y} = Y) > \tau$) in Eq. 9

$$Acc = \frac{number\_of\_correct\_predictions}{number\_of\_total\_predictions} \tag{8}$$

$$OverConf = \frac{number\_of\_overconfidence\_incorrect}{number\_of\_total\_predictions} \tag{9}$$

We want Acc to be as high as possible, with $Overconf$ as low as possible.

5.2 Experimental Results

Diversity and Uncertainty. Table 1 is the quantitative results of the accuracy metric and over-confidence metric of ***DEResNet-two stage***, DEResNet-stage1, and ResNet50. Because the SVGD model is a Bayesian-based method and the training data is small, the results in this table are the average of 10 times of training the model. Detailed results of 10 times of model training are presented in the Appendix. Regarding uncertainty, it can be seen that ResNet50 has a more severe over-confidence problem than our method. That's because our method uses Bayesian SVGD and regularizations to reasonably diversify and estimate

Table 1. The quantitative results of accuracy metric and over-confidence metric of DEResNet

	No Hat		Not Young		No Smile		Total Acc (↑)
Model	Acc (↑)	Over-Conf (↓)	Acc (↑)	Over-Conf (↓)	Acc (↑)	Over-Conf (↓)	
ResNet50	48,15	44,44	55,56	38,89	**96,23**	3,77	75,5
DEResNet-stage1	58,9 ± 6,2	**8,9**	68,3 ± 9.5	**2,8**	74,3 ± 4,7	**1,5**	69,0
DEResNet-two stage	**78,1 ± 13,9**	10,7	**76,7 ± 3,5**	7,2	79,1 ± 3,6	7,2	**78,4 ± 5,5**
ResNet-50_DivDis	51,9	0	55,6	0	84,49	0	70,4

Table 2. The quantitative results of the accuracy metric in the source testing set and the target testing set

	Source test	Target test			
Model	TestAcc	No Hat	Not Young	No Smile	Total Acc
ResNet50	97,5	61,8	57,7	**90,2**	69,7
ResNet-50_DivDis	**99,2 ± 0.7**	65,7 ± 6	**71,3 ± 7,6**	88,8 ± 0.5	75,2 ± 2,5
DEResNet-stage1	92,4 ± 5,2	71,0 ± 1,2	62,8 ± 2,9	85,7 ± 2,3	73,0
DEResNet-two stage	95,9 ± 2,5	**74,9 ± 10,9**	69,9 ± 7,9	86,7 ± 3.2	**77,1 ± 6,9**

uncertainty. In the case of the "No Smile" test set, because the accuracy of ResNet50 is good above this test set, the number of samples with overconfidence is also less. However, it is possible that ResNet50 cannot exploit information about the "Smile" hidden attribute, which is why it achieves such good accuracy. It can only capture other attributes, and when the "young person wearing a hat but not smiling" image appears, it correctly classifies it with high certainty. Proving the model can capture the attribute is very difficult and is being studied in the field of disentangled learning. We will leave the explanation in the AI model to future work.

In terms of classification performance, our DEResNet-two stage achieves the best performance on the "No Hat" set and "Not Young" set. That's because, in stage 2, we implemented SVGD and MI loss to help particles learn diverse views of the data. This will help some particles to be optimized for learning the "Hat" attribute, while some other particles are optimal for learning the "Young" attribute. For the "No Smile" set, as explained above, DEResNet in stage 1 tried to minimize the confidence when predicting the "No Smile" set. This increases the chances of DEResNet being able to capture the "Smile" attribute. This can be seen in Table 5 in the Appendix section. In rows test 1,2,3,4,5 and test 10; the optimal sub-model for the "No Smile" set is different from the sub-model of the other two hidden attributes. This shows that there was a particle that could keep the "Smile" attribute and preferred to learn "Smile" over "Young" and "Hat".

Model Performance and Generality. Table 2 is the quantitative results of the accuracy metric in the source testing set and the target testing set. Compared

Fig. 5. The incorrect predictions of ResNet50 and DEResNet.

to the results of the baseline model, our method has slightly reduced results on the source testing set. However, the results on the target testing set are much superior, especially for the DEResNet-two stage. This shows that after the ambiguity reduction stage, the model can specialize well for each data view. Compared with Lee's method using the ResNet50 backbone with the number of headers equal to the number of particles, our model is at the target set "No Hat" and has better total accuracy. This shows that it is more effective to divide ambiguity samples into particles rather than into headers. This is understandable since our method generates much more diversity in the predictions of the particles than the predictions of the headers, which is due to the fact that we take into account diversity from the initialization stage of the weights.

Qualitative Analysis. Figure 5 shows the wrong prediction cases of ResNet50 and our method in the No hat target set. It can be seen that our model is less wrong and less overconfident in prediction than the baseline model. Specifically, in the same image of a white girl with blond hair, ResNet-50 encountered the overconfidence phenomenon when thinking that the photo was a 96% negative label. Our model is better calibrated with only false predictions with only 61% confidence. That's because the particles in our model capture the "hat" attribute, and when the smiling young person in the positive image is no longer wearing a hat, some particles will be uncertain, thus helping incorrect predictions have lower confidence.

Experiment of the Variance of the Results. Because the model is trained on a very small dataset, and SVGD has randomness coming from initializing parameters from particles, the tests have very large variance. To resolve this, we

have conducted additional runs with more consistent initial weights and added standard deviation across the experiments. Specifically, we randomly initialize 2 initial weights. Then for each initial weight, we will conduct 10 experiments with different random seeds of the same set of initial weights. The results are shown in 3 and 4, we also provide the average and standard deviation of each column. We can see that with the same initial weight, the results vary very little providing better insight into the robustness and reliability of our method.

Table 3. The target testing results of our 10 training experiments of the first initial weight of DEResNet.

	Stage1				Stage2			Disambiguate				
Test	No Hat	Not Young	No Smile	Total Acc	No Hat	Not Young	No Smile	No Hat	Not Young	No Smile	Total Acc	SEED
Test 1	73.9	61.4	85.4	73.4	74.4	60.5	87.3	85.5	68.4	91.2	81.5	9876
Test 2	73.9	63.7	86.8	74.6	70.5	64.7	88.8	79.2	71.2	87.3	79.1	1234
Test 3	75.4	60.9	85.9	73.8	68.1	55.8	88.8	75.4	63.7	92.2	76.9	9877
Test 4	74.4	62.8	86.3	74.3	67.6	59.5	87.3	79.7	69.8	86.8	78.6	1235
Test 5	73.4	57.2	88.3	72.7	70.0	61.4	81.5	76.3	73.5	94.1	81.2	1
Test 6	71.5	66.0	84.4	73.8	73.4	59.5	82.0	87.9	69.3	86.8	81.2	10
Test 7	77.8	60.9	87.8	75.3	68.6	67.0	91.2	76.3	69.3	94.1	79.7	100
Test 8	74.4	59.1	88.8	73.8	68.1	65.6	87.8	74.9	71.2	89.8	78.5	1000
Test 9	68.1	63.7	86.3	72.6	46.4	75.3	93.7	71.0	84.2	94.1	83.1	10000
Test 10	74.4	58.1	83.4	71.8	73.9	51.2	85.4	84.5	62.8	88.8	78.5	100000
AVG	73.7	61.4	86.3	73.6	68.1	62.0	87.4	**79.1**	**70.3**	**90.5**	**79.8**	
STD	2.5	2.8	1.7	1.0	8.1	6.6	3.8	5.4	5.9	3.1	1.9	

Table 4. The target testing results of our 10 training experiments of the second initial weight of DEResNet.

	Stage1				Stage2			Disambiguate				
Test	No Hat	Not Young	No Smile	Total Acc	No Hat	Not Young	No Smile	No Hat	Not Young	No Smile	Total Acc	SEED
Test 1	81.2	58.1	80.0	72.9	87.9	47.0	85.4	91.3	61.4	85.4	79.1	9876
Test 2	81.6	56.3	79.0	72.1	83.1	54.4	80.5	88.9	63.3	83.4	78.3	1234
Test 3	79.7	59.1	83.4	73.8	80.7	55.3	87.3	84.1	61.4	88.3	77.7	1
Test 4	80.2	57.2	81.5	72.7	84.5	54.0	77.1	93.7	64.2	82.0	79.7	10
Test 5	79.2	55.8	82.0	72.1	84.1	58.6	82.9	88.9	65.1	82.4	78.6	100
Test 6	83.1	50.7	85.4	72.7	87.0	46.0	82.9	91.3	63.3	87.3	80.4	1000
Test 7	81.6	58.6	80.0	73.2	76.8	61.4	82.4	89.4	69.8	84.9	81.2	1235
Test 8	82.1	55.3	81.0	72.6	95.2	50.2	85.4	83.1	60.9	90.2	77.8	9877
Test 9	80.7	56.3	81.5	72.6	75.8	57.7	79.5	79.2	63.7	82.9	75.1	10000
Test 10	82.1	55.3	83.4	73.4	82.6	50.7	84.4	89.4	61.4	89.3	79.7	100000
AVG	81.2	56.3	81.7	72.8	83.8	53.5	82.8	**87.9**	**63.4**	**85.6**	**78.8**	
STD	1.2	2.4	1.9	0.5	5.6	5.0	3.1	4.4	2.6	3.0	1.7	

6 Conclusion

This paper introduced a new dataset and a novel architecture (DEResNet) to handle the problem of ambiguity in tasks, useful for handling uncertainty in few-shot learning and prompting foundation models. Our proposed model has demonstrated its ability to capture hidden attributes and gives a set of diverse solutions.

7 Future Work

In the future, it would be interesting to study the risk bound and ambiguity bound of particles in the parameter space and research the effectiveness of the disentangled method in the YSH dataset. In additional, this article is inspired by the active learning problem in [Lee et al. (2022)]. In the future, we will assume a part of the trained target distribution will be given, meaning that the label y_t will still mix all 3 sub-datasets to test the ability to divide the learning sub-datasets into particles so that each particle will achieve the best performance on one of the 3 sub-datasets. We also hope to be able to apply Scimeca's work [Scimeca et al. (2022)] to analyze the Kolmogorov complexity of 3 ambiguous classes in the future.

8 Appendix

Because the SVGD model is a probabilistic model and the training data is small, the results between tests may vary due to different parameter initialization. So we tested DEResNet 10 times and averaged the results for Table 1.

Table 5 is the results of our 10 training runs of DEResNet. Table 6 is the target testing results of our 10 training experiments of DEResNet. Table 2 is the result when calculating the average of the 10 rows of the Table 6. In which, f_i represents the i-th particle, $f_i + f_j$ represents the sub-ensemble of the particle i and particle j.

Table 5. The results of our 10 training experiments of DEResNet.

Stage 1								Stage 2						Disambiguate						
	No Hat		Not Young		No Smile		Total_stage1	TestAcc	No Hat		Not Young		No Smile		No Hat		Not Young		No Smile		Total Acc	TestAcc
Test	Acc	Over-Conf	Acc	Over-Conf	Acc	Over-Conf			Acc	Over-Conf	Acc	Over-Conf	Acc	Over-Conf	sub_model	Acc	sub_model	Acc	sub_model	Acc		
Test 1	55,6	7,4	55,6	11,1	75,5	0,0	66,3	87,5	59,3	11,1	61,1	11,1	79,2	16,7	f2	74,1	f2+f0	83,3	f1	79,2	78,6	93,75
Test 2	63,0	7,4	55,6	5,6	73,6	1,9	67,3	86,3	63,0	22,2	66,7	11,1	75,5	11,1	f1+f2	63,0	f0+f2	77,8	f1+f2	75,5	72,4	93,75
Test 3	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	59,3	18,5	72,2	5,6	75,5	11,1	f2	74,1	f0+f2	72,2	f1+f2	77,4	75,5	93,7
Test 4	70,4	7,4	55,6	11,1	77,4	1,9	71,4	87,3	55,6	14,8	61,1	5,6	54,7	16,7	f1	59,3	f1+f2	72,2	f1+f2	79,2	72,4	95
Test 5	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	40,7	14,8	61,1	11,1	77,4	5,6	f1+f2	70,4	f0+f1	77,8	f1	75,5	74,5	93,7
Test 6	63,0	3,7	72,2	0,0	79,2	0,0	73,5	98,0	55,6	3,7	72,2	5,6	81,1	0,0	f0	96,3	f1+f2	77,8	f1	84,9	86,7	98,75
Test 7	59,3	3,7	72,2	0,0	77,4	0,0	71,4	98,8	59,3	3,7	66,7	5,6	84,9	0,0	f0+f1	96,3	f1+f2	77,8	f0+f1	81,1	84,7	98,75
Test 8	59,3	3,7	72,2	0,0	77,4	0,0	71,4	98,8	59,3	3,7	66,7	5,6	84,9	0,0	f0+f1	96,3	f1	77,8	f1	81,1	84,7	98,75
Test 9	63,0	0,0	66,7	0,0	79,2	0,0	72,4	97,5	63,0	0,0	66,7	0,0	77,4	5,6	f0	81,5	f1+f2	72,2	f0	83,0	80,6	98,75
Test 10	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	40,7	14,8	61,1	11,1	77,4	5,6	f1+f2	70,4	f0+f1	77,8	f1+f2	73,6	73,5	93,75
AVG	58,9	8,9	68,3	2,8	74,3	1,5	69,0	92,4	55,6	10,7	65,6	7,2	76,8	7,2		78,1		76,7		79,1	78,4	95,9
STD	6,2	7,0	9,5	4,7	4,7	1,7	3,3	5,2	8,2	7,5	4,4	3,7	8,5	6,4		13,9		3,5		3,6	5,5	2,5

Table 6. The target testing results of our 10 training experiments of DEResNet.

Stage 1								Stage 2						Disambiguate							
	No Hat		Not Young		No Smile		Total_stage1	TestAcc	No Hat		Not Young		No Smile		No Hat		Not Young		No Smile		Total Acc	TestAcc
Test	Acc	Over-Conf	Acc	Over-Conf	Acc	Over-Conf			Acc	Over-Conf	Acc	Over-Conf	Acc	Over-Conf	sub_model	Acc	sub_model	Acc	sub_model	Acc		
Test 1	55,6	7,4	55,6	11,1	75,5	0,0	66,3	87,5	59,3	11,1	61,1	11,1	79,2	16,7	f2	74,1	f2+f0	83,3	f1	79,2	78,6	93,75
Test 2	63,0	7,4	55,6	5,6	73,6	1,9	67,3	86,3	63,0	22,2	66,7	11,1	75,5	11,1	f1	63,0	f0+f2	77,8	f0+f1	75,5	72,4	93,75
Test 3	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	59,3	18,5	72,2	5,6	75,5	11,1	f2	74,1	f0+f2	72,2	f0+f1	77,4	75,5	93,7
Test 4	70,4	7,4	55,6	11,1	77,4	1,9	71,4	87,3	55,6	14,8	61,1	5,6	54,7	16,7	f0+f2	59,3	f0+f2	72,2	f0+f1	79,2	72,4	95
Test 5	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	40,7	14,8	61,1	11,1	77,4	5,6	f0+f2	70,4	f0+f2	77,8	f1	75,5	74,5	93,7
Test 6	63,0	3,7	72,2	0,0	79,2	0,0	73,5	98,0	55,6	3,7	72,2	5,6	81,1	0,0	f0	96,3	f2	77,8	f2+f0	84,9	86,7	98,75
Test 7	59,3	3,7	72,2	0,0	77,4	0,0	71,4	98,8	59,3	3,7	66,7	5,6	84,9	0,0	f0	96,3	f2	77,8	f2+f0	81,1	84,7	98,75
Test 8	59,3	3,7	72,2	0,0	77,4	0,0	71,4	98,8	59,3	3,7	66,7	5,6	84,9	0,0	f0	96,3	f2	77,8	f0+f2	81,1	84,7	98,75
Test 9	63,0	0,0	66,7	0,0	79,2	0,0	72,4	97,5	63,0	0,0	66,7	0,0	77,4	5,6	f0	81,5	f2	72,2	f0	83,0	80,6	98,75
Test 10	51,9	18,5	77,8	0,0	67,9	3,8	65,3	90,0	40,7	14,8	61,1	11,1	77,4	5,6	f0+f2	70,4	f0+f2	77,8	f0+f1	73,6	73,5	93,75
AVG	58,9	8,9	68,3	2,8	74,3	1,5	69,0	92,4	55,6	10,7	65,6	7,2	76,8	7,2		78,1		76,7		79,1	78,4	95,9
STD	6,2	7,0	9,5	4,7	4,7	1,7	3,3	5,2	8,2	7,5	4,4	3,7	8,5	6,4		13,9		3,5		3,6	5,5	2,5

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International Conference on Machine Learning, pp. 1613–1622. PMLR (2015)

Fan, J., Zhang, Q., Taghvaei, A., Chen, Y.: Variational Wasserstein gradient flow. arXiv preprint arXiv:2112.02424 (2021)

Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Insights and applications. In: Deep Learning Workshop, vol. 1, pp. 2, ICML (2015)

He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, vol. 30 (2017)

Lee, Y., Yao, H., Finn, C.: Diversify and disambiguate: Out-of-distribution robustness via disagreement. In: The Eleventh International Conference on Learning Representations (2022)

Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29 (2015)

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)

Scimeca, L., Joon Oh, S., Chun, S., Poli, M., Yun, S.: Which shortcut cues will DNNs choose? A study from the parameter-space perspective. ICLR (2022)

Wang, D., Tang, Z., Bajaj, C., Liu, Q.: Stein variational gradient descent with matrix-valued kernels. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

Wang, Z., Shen, L., Fang, L., Suo, Q., Duan, T., Gao, M.: Improving task-free continual learning by distributionally robust memory evolution. In: International Conference on Machine Learning, pp. 22985–22998. PMLR (2022)

Zhang, Y., et al.: Risk bounds on aleatoric uncertainty recovery. In: International Conference on Artificial Intelligence and Statistics, pp. 6015–6036. PMLR (2023)

When Uncertainty-Based Active Learning May Fail?

Amir Hossein Rahmati[1(✉)], Mingzhou Fan[1], Ruida Zhou[2], Nathan M. Urban[3], Byung-Jun Yoon[1,3], and Xiaoning Qian[1,3]

[1] Texas A&M University, College Station, TX 77843, USA
{amir_hossein_rahmati,mzfan,bjyoon,xqian}@tamu.edu
[2] University of California, Los Angeles, CA 90095, USA
ruida@g.ucla.edu
[3] Brookhaven National Laboratory, Upton, NY 11973, USA
{nurban,byoon,xqian1}@bnl.gov

Abstract. Instead of randomly acquiring training data, Uncertainty-based Active Learning (UAL) selects pivotal samples from an unlabeled dataset based on the prediction uncertainty and queries their labels so that the labeling cost for model training can be minimized. As a result, the efficacy of UAL depends on the model capacity as well as the adopted uncertainty-based acquisition function. In this study, our analytical focus is directed toward comprehending how the capacity of the machine learning model may affect UAL efficacy. Through theoretical analysis and comprehensive simulation and empirical studies, we demonstrate that UAL can lead to worse performance compared to random sampling when the machine learning model class has low capacity and is unable to cover the underlying ground-truth.

Keywords: Active Learning (AL) · Bayesian Active Learning (BAL) · Uncertainty-based Active Learning (UAL)

1 Introduction

With the advent of novel profiling and database technologies, the ever-increasing volume of available data gives rise to extensive unlabeled datasets that may help further advance the AI and machine learning (AI/ML) development [27]. However, the associated cost of labeling is exorbitantly prohibitive [4,13,15,26], to which Active Learning (AL) can be a potential solution.

AL aims to reduce the quantity of labeled training data while achieving the desired prediction performance in ML [20]. Based on a selected acquisition function, AL iteratively queries the labels of the most informative samples, hoping to learn with better sample efficiency utilizing much fewer samples than what

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_6.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 84–100, 2025.
https://doi.org/10.1007/978-3-031-78107-0_6

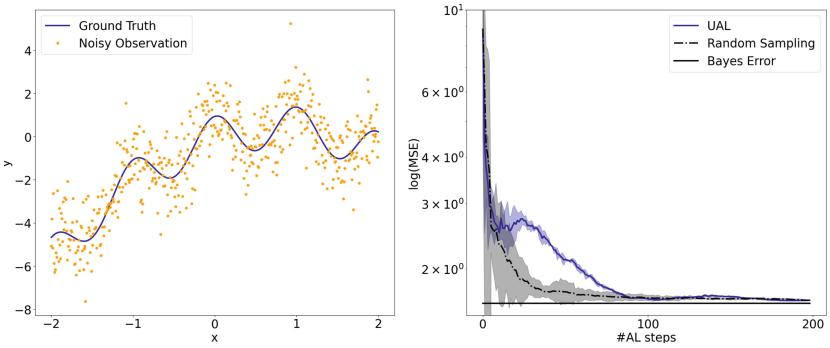


Fig. 1. Motivating example with a ground-truth target that is more complex than the prediction model class. In this example, the noisy data is generated by the ground-truth function that is the summation of a quadratic function and a cosine function ($y = f(x) + \epsilon$ with $\epsilon \sim N(0, 1)$, where $f(x) = \langle \mathbf{x}, \mathbf{w} \rangle + cos(2\pi x)$, and $\mathbf{x} = [1, x, x^2]$). The left plot shows the ground-truth target function and its corresponding noisy observed data. The right plot shows the comparison of UAL and random sampling performance to learn the quadratic predictor based on Bayesian polynomial regression (BPR).

is available in the initial unlabeled dataset [14]. This iterative AL procedure persists until either the model achieves the desired prediction performance or the designated labeling budget is exhausted. These methods broadly can be divided into *membership query synthesis* [1,14,20], *stream-based selective sampling* [3,14,20], and *pool-based active learning* [9,14,20] based on the problem setup. There are two main categories of sample selection strategies considered in pool-based active learning methods: *Uncertainty-based AL* (UAL) [8,16,20,23–25] and *Diversity-based AL* (DAL) [5,14,18–21]. Some studies also focused on using both categories leading to hybrid methods [2,28]. For UAL, samples are selected based on the model's prediction confidence/uncertainty reflecting their significance in improving the model's performance. Variance-based or information-theoretic methods (estimated by entropy or mutual information), such as Bayesian Active Learning by Disagreement (BALD) [7], are considered under this category. DAL strives to identify the subset encapsulating the underlying data distribution, including core-set methods [18]. This study centers its attention on UAL, which identifies and queries labels for the most uncertain samples, implying their importance for the current model training [20].

Despite the popularity of UAL methods, some previous works have demonstrated situations where they may not outperform random sampling [6,11,17,22], which motivates this study. To better illustrate the need to understand the behaviour of UAL performance, we first present such an example where we compare the regression performance of UAL and random sampling with a quadratic model on data generated from a more complex ground-truth function in Fig. 1. Instead of converging faster than random sampling as typically expected, UAL fails to effectively select the most informative samples to better guide the model

training with the prediction performance of learned regressors worse than the ones trained with random sampling.

These observations have triggered our curiosity to investigate the potential effects of the model capacity on the efficiency of UAL. We seek to find the necessary settings from the model's perspective, under which UAL can outperform random sampling. In particular, we establish that given a model of at least adequate complexity covering the underlying ground-truth target function to learn, UAL is able to have superior performance than random sampling.

The subsequent sections are organized as follows: Sect. 2 provides the necessary background, definitions, and UAL settings. In Sect. 3, we analyze the UAL efficacy under different model settings. We present experimental results supporting the findings of the analysis in Sect. 4. We then conclude the study in Sect. 5.

2 Problem Setting

In this work, we focus on the performance of pool-based UAL for regression under the metric of Mean Squared Error (MSE) for illustrative theoretical analysis. The regression problem is to assign $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ to its corresponding output y with the assumption that the input random vector $\mathbf{X}$ and output random variable Y are jointly distributed following $P(\mathbf{X}, Y)$. The common setting is that, at an arbitrary $\mathbf{X} = \mathbf{x}$, the output is $y = f(\mathbf{x}) + \epsilon$, where $f : \mathcal{X} \to \mathbb{R}$ is the underlying ground-truth function and $\epsilon \sim N(0, \sigma^2)$ is the white noise residual error determining the conditional distribution $P(Y = y|\mathbf{X} = \mathbf{x})$. Given an observed set of data points, $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$, $\hat{y} = \hat{f}_\theta(\mathbf{x}) + \epsilon$ serves as an estimator of y in the regression task, where a regressor $\hat{f}_\theta : \mathcal{X} \to \mathbb{R}$, is a parameterized predictive model with θ being the model parameters.

Bayesian Learning. Bayesian Learning aims to maintain the posterior distribution of the parameters θ, $P^*(\theta|\mathbf{D}) \propto P(\theta)P(\mathbf{D}|\theta)$, based on the Bayes rule with the observed dataset $\mathbf{D}$, where $P(\theta)$ is the prior distribution reserving prior knowledge, and $P(\mathbf{D}|\theta)$ is the likelihood function representing the probability of generating the observed data with specific parameter θ. Via the parameter posterior, the predictive posterior at an arbitrary $\mathbf{x}$ can be calculated as follows:

$$\pi^*(\hat{y}|\mathbf{x}) = \int P(\hat{y}|\theta, \mathbf{X} = \mathbf{x}) P^*(\theta|\mathbf{D}) d\theta. \tag{1}$$

Bayesian Active Learning (BAL). In BAL, we consider the training dataset to be composed of labeled and unlabeled data, i.e. $\mathbf{D} = \mathbf{D_L} \bigcup \mathbf{D_U}$, where $\mathbf{D_L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_\mathbf{L}}$ is the labeled dataset and $\mathbf{D_U} = \{\mathbf{x}_j\}_{j=1}^{n_\mathbf{U}}$ is the unlabeled dataset, $n_\mathbf{L}$ and $n_\mathbf{U}$ are the number of labeled and unlabeled samples respectively. The BAL procedure aims at selecting the most informative sample in the unlabeled dataset by optimizing the acquisition function $\mathcal{A}$. The label

of selected $\mathbf{x}^* \in \arg\max_{\mathbf{x_U} \in \mathbf{D_U}} \mathcal{A}(\pi^*, \mathbf{x_U})$ would be queried and added to the labeled dataset, while $\mathbf{x}^*$ will be removed from the unlabeled dataset. This procedure ends when the performance is satisfactory or the budget is exhausted, $\mathbf{D_U} = \varnothing$. In the context of Bayesian UAL, equipped with learned posterior distributions, $\mathcal{A}$ is designed based on the prediction uncertainty that hopefully captures both uncertainties coming from data and model (the higher the score is, the higher the uncertainty is). With the same procedure as BAL, the most uncertain sample will be added iteratively to the labeled dataset at each UAL step. Figure 2 schematically illustrates this active learning process.

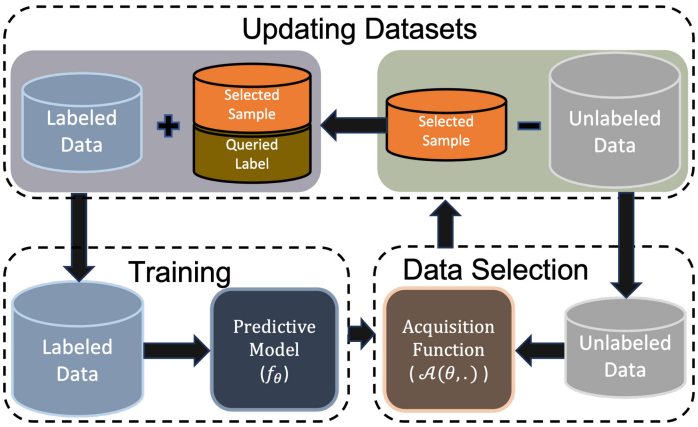


Fig. 2. Schematic illustration of the UAL procedure

Learning Objective. Our UAL performance analysis will be based on the MSE of prediction with respect to observed output, i.e., MSE_{obs}, which can be written as:

$$\begin{aligned}\text{MSE}_{\text{obs}} &= E_{P(Y|\mathbf{X})}[E_{P^*(\theta|\mathbf{D_L})}[(Y - \hat{f}_\theta)^2]] \\ &= \int P(Y|\mathbf{X})(\int P^*(\theta|\mathbf{D_L})(Y - \hat{f}_\theta)^2 d\theta) dY,\end{aligned}$$

where P denotes the ground-truth distribution, and P^* the derived posterior distribution given training data.

3 UAL Efficacy Analysis

Following the problem setup in Sect. 2, we first decompose MSE_{obs} for latter analysis of UAL performance with different prediction model classes:

$$\begin{aligned}\text{MSE}_{\text{obs}} &= E_{P(Y|\mathbf{X})}[E_{P^*(\theta|\mathbf{D_L})}[(Y - E_{P(Y|\mathbf{X})}[Y] + E_{P(Y|\mathbf{X})}[Y] - \hat{f}_\theta)^2]] \\ &= E_{P(Y|\mathbf{X})}[(Y - E_{P(Y|\mathbf{X})}[Y])^2] + E_{P(Y|\mathbf{X})}[E_{P^*(\theta|\mathbf{D_L})}[(E_{P(Y|\mathbf{X})}[Y] - \hat{f}_\theta)^2]]\end{aligned}$$

In the above MSE_{obs} decomposition, the first term comes from the observation noise and the second term is related to the error arising from the predictive model settings. Since we aim to investigate UAL's performance from the model's perspective, our focus is on the analysis of the second term, i.e., MSE of predicted output from the ground-truth f, denoted as:

$$\begin{aligned}\text{MSE} &= E_{P(Y|\mathbf{X})}\left[E_{P^*(\theta|\mathbf{D_L})}[(\hat{f}_\theta - E_{P(Y|\mathbf{X})}[Y])^2]\right] \\ &= E_{P(Y|\mathbf{X})}\Big[\underbrace{E_{P^*(\theta|\mathbf{D_L})}[(\hat{f}_\theta - E_{P^*(\theta|\mathbf{D_L})}[\hat{f}_\theta])^2]}_{\textbf{Variance}} \\ &\quad + \underbrace{(E_{P^*(\theta|\mathbf{D_L})}[\hat{f}_\theta] - E_{P(Y|\mathbf{X})}[Y])^2}_{\textbf{Bias}}\Big]\end{aligned}$$

The above equation gives the well-known bias-variance decomposition. When the derived posterior P^* is close enough to the ground-truth, the **Bias** term vanishes so that the **Variance** term will faithfully capture MSE and represents the model's prediction performance; and therefore, it can be an effective acquisition function leading to better UAL performance (lower MSE and thereafter MSE_{obs}).

Proposition 1. *$(E_{P^*(\theta|\mathbf{D_L})}[\hat{f}_\theta] - E_{P(Y|\mathbf{X})}[Y])^2 < \varepsilon^2 C^2$ if $E_{P(Y|\mathbf{X})}[|Y|] < C$ and $\left|\frac{\pi^*}{P(Y|\mathbf{X})} - 1\right| < \varepsilon$, where $C > 0$ and $\varepsilon > 0$ are constants.*[1]

The Proposition 1 indicates that as the ε goes to 0, **Bias** goes to 0 as well.

Having the MSE decomposed in the bias and variance terms, the next step is to provide illustrative UAL performance analyses for the corresponding predictive model, $\hat{f}_\theta$. We choose to concentrate on Bayesian Regression with polynomial regressors as predictive models. This choice of prediction model classes leads to closed-form MSE analysis and also enables straightforward model class capacity manipulation by increasing the polynomial order.

Bayesian Polynomial Regression (BPR) is a widely utilized parametric model class with polynomial functions. Let $\hat{f}_\theta(\mathbf{x}) = \langle \phi(\mathbf{x}, p), \theta \rangle$, where $\phi(\mathbf{x}, p)$ is a non-linear operator that maps $\mathbf{x}$ to a p^{th}-order polynomial expansion and θ is the corresponding weight vector [12].

With the typical white noise residual error assumption, the observed output Y follows the Gaussian distribution,

$$P(Y|\mathbf{X} = \mathbf{x}) = N(f(\mathbf{x}), \sigma^2). \tag{2}$$

We can derive the predictive posterior given $\mathbf{x}$ in BPR with the conjugate prior. Specifically, with the conjugate prior $\theta \sim N(\hat{\mu}, \hat{\Sigma})$ where $\hat{\mu}$ and $\hat{\Sigma}$ are the prior mean and covariance, the posterior of the model's parameters can be derived by

[1] Proof is provided in the Appendix.

Bayes' rule:

$$
\begin{aligned}
P^*(\theta|\hat{\mathbf{\Phi}}, \mathbf{y}, \sigma^2) &\propto N(\theta|\hat{\mu}, \hat{\Sigma}) N(\mathbf{y}|\hat{\mathbf{\Phi}}\theta, \sigma^2) = N(\theta|\hat{\mu}_p, \hat{\Sigma}_p) \\
\hat{\mu}_p &= \hat{\Sigma}_p \hat{\Sigma}^{-1} \hat{\mu} + \frac{1}{\sigma^2} \hat{\Sigma}_p \hat{\mathbf{\Phi}}^\top \mathbf{y}; \\
\hat{\Sigma}_p^{-1} &= \hat{\Sigma}^{-1} + \frac{1}{\sigma^2} \hat{\mathbf{\Phi}}^\top \hat{\mathbf{\Phi}},
\end{aligned}
\tag{3}
$$

where $\hat{\mathbf{\Phi}} = [\phi_1^p, \ldots, \phi_n^p]^\top$ is the transformed input matrix with n transformed p^{th}-order polynomial terms $\phi_i^p = \phi(\mathbf{x}_i, p)$, $\mathbf{y} = [y_1, \ldots, y_n] \in \mathbb{R}^n$ is the corresponding training output vector with $y_i \sim P(Y|\mathbf{X} = \mathbf{x}_i)$, and $\hat{\mu}_p, \hat{\Sigma}_p$ denote the posterior mean and covariance respectively.

By substituting the parameter posterior to Eq. (1) it can be shown that the predictive posterior at $\mathbf{x}$ follows a Gaussian distribution:

$$
\pi^*(\hat{y}|\mathbf{x}) \sim N(\langle \phi(\mathbf{x}, p), \hat{\mu}_p \rangle, \sigma_p^2(\mathbf{x})), \tag{4}
$$

where

$$
\sigma_p^2(\mathbf{x}) = \sigma^2 + \phi(\mathbf{x}, p)^\top \hat{\Sigma}_p \phi(\mathbf{x}, p) \tag{5}
$$

is the posterior predictive variance. Since $\sigma_p^2(\mathbf{x})$ captures model prediction uncertainty, it can be used to define a meaningful acquisition function to guide UAL for BPR. As a result, in this study, we consider posterior predictive variance in (5) as the acquisition function, i.e., $\mathcal{A}(\pi^*, \mathbf{x}) = \sigma_p^2(\mathbf{x})$, and the following analysis is based on this choice.

Without loss of generality, to evaluate the reduction of the MSE of actively learned models given acquired labeled training data, we consider BPR with univariate inputs ($d = 1$) with $\phi(\mathbf{x}, p) = [1, \ldots, x^p]$ (p: the polynomial order, considered as the model complexity). We note that the analysis results can be easily generalized to multivariate cases.

As mentioned earlier in this section, when the derived P^* is close enough to the ground-truth model that generates the data, **Variance** can represent the MSE, thus leading to better UAL performance. The closeness of P^* to the ground-truth is affected by the prediction model class capacity. More specifically, when the model class capacity is flexible enough to capture the underlying ground-truth function, the MSE can be estimated based on the prediction **Variance**.

As an example, Fig. 4 compares the UAL performance with random sampling when the underlying function's complexity matches the predictive model class with both being quadratic functions. It is clear that UAL is able to outperform random sampling, implying that the estimated acquisition function based on the predictive variance faithfully represents MSE when the model class covers the ground-truth target, leading to efficient UAL, aligning with the analysis.

While when checking the bias and variance decomposition of the experiment in Sect. 1 as demonstrated in Fig. 3, since the model class complexity is not enough to cover the target function, the bias term can dominate the MSE and the variance term does not fully capture the MSE, making it an uninformative acquisition function unable to guide UAL efficiently.

In what follows, we delve deeper into the analysis of MSE, considering an uncertainty class of the ground-truth functions. Assuming that the uncertainty class of ground-truth functions is the l^{th}-order polynomial family, $f(\mathbf{x}) = \langle \phi(\mathbf{x}, l), \mathbf{w} \rangle \in \mathbb{R}$, with $\mathbf{w} \sim N(\mu, \Sigma)$, the MSE can be decomposed as:

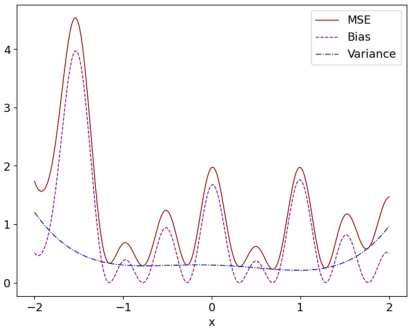


Fig. 3. Bias-variance decomposition for the motivating example with lower complexity prediction model in Sect. 1. In this example the estimated variance cannot capture the learning objective, mean squared error (MSE) for regression.

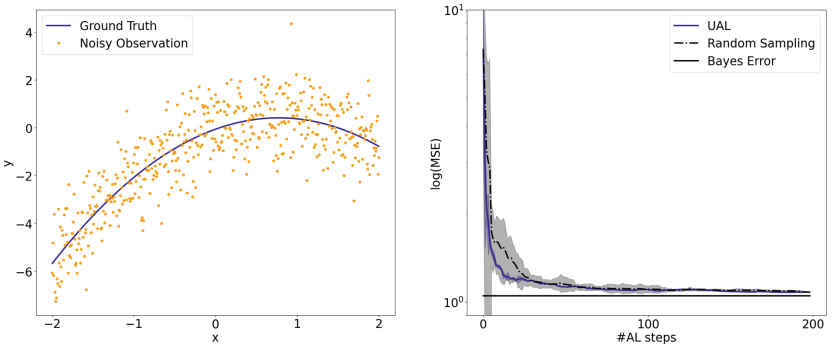


Fig. 4. Motivating example with a ground-truth target as complex as the prediction model class. In this example, the noisy data follows the ground-truth function, which is a quadratic function ($y = f(x) + \epsilon$ with $\epsilon \sim N(0, 1)$, where $f(x) = \langle \mathbf{x}, \mathbf{w} \rangle$, and $\mathbf{x} = [1, x, x^2]$). The left plot shows the ground-truth target function with the same $\mathbf{w}$ as the example in Sect. 1 and its corresponding noisy observed data. The right plot shows the comparison of UAL and random sampling to learn the quadratic predictor by BPR.

$$
\begin{aligned}
\text{MSE} &= E_{P(Y|\mathbf{X})}\left[E_{P^*(\theta|\mathbf{D}_L)}[(f(\mathbf{x}) - \langle \phi(\mathbf{x}, p), \hat{\mu}_p \rangle)^2]\right. \\
&\quad \left. + E_{P^*(\theta|\mathbf{D}_L)}[(\langle \phi(\mathbf{x}, p), \hat{\mu}_p \rangle - \langle \phi(\mathbf{x}, p), \theta \rangle)^2]\right] \\
&= E_{P(Y|\mathbf{X})}[(f(\mathbf{x}) - \langle \phi(\mathbf{x}, p), \hat{\mu}_p \rangle)^2 + \phi(\mathbf{x}, p)^\top \hat{\Sigma}_p \phi(\mathbf{x}, p)] \\
&= \underbrace{E_{P(\mathbf{w})}[E_{P(Y|\mathbf{w},\mathbf{X})}[(f(\mathbf{x}) - \langle \phi(\mathbf{x}, p), \hat{\mu}_p \rangle)^2]]}_{\textbf{Bias}} + \underbrace{\phi(\mathbf{x}, p)^\top \hat{\Sigma}_p \phi(\mathbf{x}, p)}_{\textbf{Variance}}.
\end{aligned}
$$

By replacing posterior mean ($\hat{\mu}_p$) in (3) and taking the expectation, the MSE is:

$$\begin{aligned}\text{MSE} =&(\phi^l)^\top(\mu\mu^\top+\Sigma)(\phi^l)-2(\phi^p)^\top\hat{\Sigma}_p\hat{\Sigma}^{-1}\hat{\mu}\mu^\top\phi^l-\frac{2}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\mathbf{\Phi}(\mu\mu^\top+\Sigma)\phi^l\\&+(\phi^p)^\top\hat{\Sigma}_p\hat{\Sigma}^{-1}\hat{\mu}\hat{\mu}^T\hat{\Sigma}^{-1}\hat{\Sigma}_p(\phi^p)+\frac{1}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\Sigma}^{-1}\hat{\mu}\mu^\top\mathbf{\Phi}^\top\hat{\mathbf{\Phi}}\hat{\Sigma}_p(\phi^p)\\&+\frac{1}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\mathbf{\Phi}\mu\hat{\mu}^\top\hat{\Sigma}^{-1}\hat{\Sigma}_p(\phi^p)\\&+\frac{1}{\sigma^4}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\mathbf{\Phi}(\mu\mu^\top+\Sigma)\mathbf{\Phi}^\top\hat{\mathbf{\Phi}}\hat{\Sigma}_p(\phi^p)+\frac{1}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\hat{\mathbf{\Phi}}\hat{\Sigma}_p(\phi^p)\\&+(\phi^p)^\top\hat{\Sigma}_p(\phi^p),\end{aligned}\tag{6}$$

where $\phi^p=\phi(\mathbf{x},p)$, $\phi^l=\phi(\mathbf{x},l)$, and $\mathbf{\Phi}\in\mathbb{R}^{n\times(l+1)}$.

Assuming the model's complexity matches the target's complexity ($p=l$), we can further simplify the MSE. More specifically, based on this assumption, $\mathbf{\Phi}$ and $\hat{\mathbf{\Phi}}$, ϕ^l and ϕ^p are the same. As a result, $\mathbf{\Phi}\hat{\mathbf{\Phi}}^\top=\hat{\mathbf{\Phi}}\mathbf{\Phi}^\top=\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^\top$. Furthermore, considering knowing the uncertainty class of ground-truth functions, the prior mean and covariance are assumed to be the same as the uncertainty class, i.e., μ and $\hat{\mu}$, Σ and $\hat{\Sigma}$ are equal. Also, considering that $\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^\top=\sigma^2(\hat{\Sigma}_p^{-1}-\hat{\Sigma}^{-1})$, by replacing $\hat{\mathbf{\Phi}}\hat{\mathbf{\Phi}}^\top$ with its equivalent counterpart, the MSE for the matched model is[2]:

$$\text{MSE}=2(\phi^p)^\top\hat{\Sigma}_p(\phi^p)=2(\sigma_p(\mathbf{x})^2-\sigma^2).\tag{7}$$

Equation (7) shows that when the prediction model class matches the target ground-truth, the **Variance** term is proportional to the MSE; i.e., using the posterior predictive variance, one can find what would be the MSE of a sample for the currently trained model, and consequently, which point is the best to be added to the training set. This can effectively guide UAL to outperform the random sampling based training strategy.

With Eq. (7) giving an intuitive explanation to MSE of models with matched complexity as the target, the analysis for the unmatched models would not lead to a similar clean form. It is believed that with reasonable Bayesian inference, higher-order prediction models in BPR can provide a predictive variance that still captures the actual MSE fairly well. However, the lower-order model exhibits a notably pronounced **Bias** with respect to its **Variance** deterring the effectiveness of using the **Variance** as the MSE surrogate. For this reason we further analyze the lower-order model where the model's order is smaller than the target's order ($p<l$).

Before delving into the analysis, we define the following matrices:

$$\mathbf{\Phi}=[\tilde{\mathbf{\Phi}}_c\quad\hat{\mathbf{\Phi}}]\in\mathbb{R}^{n\times(l+1)},\qquad\mu=\begin{bmatrix}\tilde{\mu}_c\\\tilde{\mu}\end{bmatrix}\in\mathbb{R}^{(l+1)},$$

$$\phi^l=\begin{bmatrix}\tilde{\phi}_c\\\phi^p\end{bmatrix}\in\mathbb{R}^{(l+1)},\qquad\Sigma=\begin{bmatrix}\tilde{\Sigma}_c&\Sigma_{12}\\\Sigma_{12}^\top&\tilde{\Sigma}\end{bmatrix}\in\mathbb{R}^{(l+1)\times(l+1)}$$

[2] The detailed derivation for Eq. (6) and (8) is provided in Appendix.

with $\tilde{\phi}_c = [x^{p+1}, \ldots, x^l]$, where

$$\begin{aligned} \tilde{\mathbf{\Phi}}_c \in \mathbb{R}^{n\times(l-p)}, \qquad \hat{\mathbf{\Phi}} \in \mathbb{R}^{n\times(p+1)}, \\ \tilde{\mu}_c \in \mathbb{R}^{(l-p)}, \qquad \tilde{\mu} \in \mathbb{R}^{(p+1)}, \\ \tilde{\phi}_c \in \mathbb{R}^{(l-p)}, \qquad \phi^p \in \mathbb{R}^{(p+1)}, \\ \tilde{\Sigma}_c \in \mathbb{R}^{(l-p)\times(l-p)}, \Sigma_{12} \in \mathbb{R}^{(l-p)\times(p+1)}, \tilde{\Sigma} \in \mathbb{R}^{(p+1)\times(p+1)}. \end{aligned}$$

Assuming $\tilde{\mu} = \hat{\mu}$ and $\tilde{\Sigma} = \hat{\Sigma}$, after plugging in the equivalent matrices to Eq. (6), the MSE will be:

$$\begin{aligned} \text{MSE} &= (\tilde{\phi}_c)^\top(\tilde{\Sigma}_c + \tilde{\mu}_c\tilde{\mu}_c^\top)(\tilde{\phi}_c) - \frac{2}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\tilde{\mathbf{\Phi}}_c(\tilde{\Sigma}_c + \hat{\mu}_c\hat{\mu}_c^\top)(\tilde{\phi}_c) \\ &\quad + 2(\phi^p)^\top\hat{\Sigma}_p\hat{\Sigma}^{-1}\Sigma_{12}^\top(\tilde{\phi}_c) + \frac{1}{\sigma^4}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\tilde{\mathbf{\Phi}}_c(\tilde{\Sigma}_c + \hat{\mu}_c\hat{\mu}_c^\top)\tilde{\mathbf{\Phi}}_c^\top\hat{\mathbf{\Phi}}\hat{\Sigma}_p(\phi^p) \\ &\quad - \frac{2}{\sigma^2}(\phi^p)^\top\hat{\Sigma}_p\hat{\mathbf{\Phi}}^\top\tilde{\mathbf{\Phi}}_c\Sigma_{12}\hat{\Sigma}_0^{-1}\hat{\Sigma}_p(\phi^p) + 2(\phi^p)^\top\hat{\Sigma}_p(\phi^p) \\ &= \mathrm{P}(\mathbf{x}) + 2\times\mathrm{Var}(\mathbf{x}), \end{aligned} \tag{8}$$

where the **Variance** term $\mathrm{Var}(\mathbf{x}) = (\phi^p)^\top\hat{\Sigma}_p(\phi^p)$ is a $(2\times p)^{th}$-order polynomial function of $\mathbf{x}$, and $\mathrm{P}(\mathbf{x})$ consists of the remaining terms in MSE and is a $(2\times l)^{th}$-order polynomial function of $\mathbf{x}$.

Importantly, it means that in the lower-order model case, the complexity (order) of the **Variance** term diverges from the true MSE. This would potentially result in the inability of the estimated uncertainty in accurately capturing the true learning objective, MSE, as we will observe in the latter experiments. Consequently, relying on **Variance** (uncertainty metric) for sample selection yields choices that lack informativeness, which may lead to degraded UAL performance.

4 Experiments

We have theoretically analyzed the performance of UAL algorithms, which relies on whether their adopted acquisition function based on the estimated uncertainty can faithfully portray the ultimate learning objective, MSE, when considering regression in this paper. In Sect. 3, we show that the predictive variance is able to capture regression learning objective when the model's complexity aligns or surpasses that of the underlying ground-truth target function.

In this section, to further investigate the validity of our findings and the effect of model complexity mismatch on UAL performance, we first provide a more comprehensive evaluation of experimental results by utilizing a synthetic dataset tailored to our aims to demonstrate the relationships between the predictive model complexity, acquisition function effectiveness, and UAL performance (measured by MSE). Moreover, to further validate the reliability of our results, we carry out experiments to demonstrate UAL performance on two real-world

open-access datasets comparing the UAL performances with a simple model and a complex model representing low vs. high model complexity classes respectively.

The UAL method in all our experiments and simulations is based on the basic pool-based UAL setting: the prediction model will be trained on the initial labeled dataset, $\mathbf{D_L}$; the most uncertain sample based on the learned model from the unlabeled dataset $\mathbf{D_U}$ will then be queried and added to the labeled dataset to update the model in the next step. This process iterates and ends when all the designated unlabeled training data, $\mathbf{D_U}$, is exploited. In our experiments for UAL with regression, the predictive posterior variance is considered as the measure of uncertainty, and consequently, the acquisition function to query new samples from $\mathbf{D_U}$.

For deriving generalizable conclusions, we conduct experiments on a synthetic dataset that consists of 100 randomly generated 3^{rd}-order polynomial functions as the ground-truth functions to simulate the observed data for UAL performance evaluation. We collect the corresponding performance statistics from these 100 random runs to illustrate the influence of model complexity mismatch on UAL performance. For each run, we first generate an unlabeled training dataset ($\mathbf{D_U}$) of 200 evenly partitioned samples with $x \in [-2, 2]$. We then simulate the output corresponding to each sample x_n during the UAL procedure based on the following noisy observations:

$$y_n = f_n + \epsilon, \qquad \epsilon \sim N(0, \sigma^2), \tag{9}$$

where $f_n = \langle (\phi_n^3)^T, \mathbf{w} \rangle$ with $(\phi_n^3)^T = [1, x_n, x_n^2, x_n^3]$ and the corresponding randomly sampled coefficient vector for the 3^{rd}-order polynomial $\mathbf{w} \in \mathbb{R}^4$, $\mathbf{w} \sim N(\mathbf{0}, \mathbf{I})$ and $\sigma = 1$. For each run, the initial labeled dataset ($\mathbf{D_L}$) consists of one sample x that is randomly drawn from $\mathbf{D_U}$ and its corresponding label y that is acquired from Eq. (9) with the specific coefficient vector $\mathbf{w}$. To evaluate the regression and UAL performance, a holdout test dataset of 500 samples with $x \in [-2, 2]$ and their corresponding observed output y is used to estimate the test MSE. In the following, we perform two sets of experiments on this synthetic dataset to check the performance of UAL when using Bayesian Polynomial and Gaussian Process Regressors as the predictive model, respectively. The initial $\mathbf{D_L}$ and target functions are the same across both sets.

To verify the reliability of our conclusions, we provide additional experimental results on two real-world case studies. In particular, we use the *Concrete Compressive Strength* [29] and *Facebook Metrics* [10] datasets for the regression task. For the *Facebook Metrics* dataset, we use 25% of the dataset as the test dataset and the rest as the training dataset. For the *Concrete Compressive Strength* dataset, we randomly select 600 samples from the dataset, with the same train/test split ratio as in the *Facebook Metrics* dataset. In both case studies, we consider using Gaussian Process Regressors as the predictive model.

Bayesian Polynomial Regression. In this set of experiments, we perform Bayesian polynomial Regression (BPR) for the corresponding model class of polynomial order one (linear) to five and compare the learning performance

based on the UAL and random sampling strategies. In each run, starting from one randomly selected pair of input and output, the performance of each polynomial model in UAL and random sampling procedure is evaluated.

The striking trend in Fig. 5 is that when the model class is linear or quadratic functions, UAL shows significant worse performance than random sampling, while in Fig. 6 when the model class complexity increases over the ground-truth 3^{rd}-order polynomial functions, UAL performs better as typically expected. This can be explained based on the analysis in Sect. 3.

To clearly illustrate the issues of UAL when the model class does not cover the target ground-truth function to learn, we further visualize the **Bias** and **Variance** decomposition patterns. In the second row of Fig. 6 it is clear that when the model class complexity matches or surpasses the target's complexity, the estimated predictive variance captures the actual MSE. However, bias-variance decomposition patterns in Fig. 5 show model classes (linear and quadratic) with low capacity are incapable of providing an accurate estimation of MSE via the variance-based UAL acquisition function; hence, they lead to degraded UAL performance, which can perform even worse than random sampling.

Another observation in this set of BPR experiments is that at the early stages of UAL, a much sharper drop in the error happens compared to random sampling (Fig. 7). Even for models with insufficient complexity, we observe this phenomenon before they get stuck in the uninformative regions. However, after

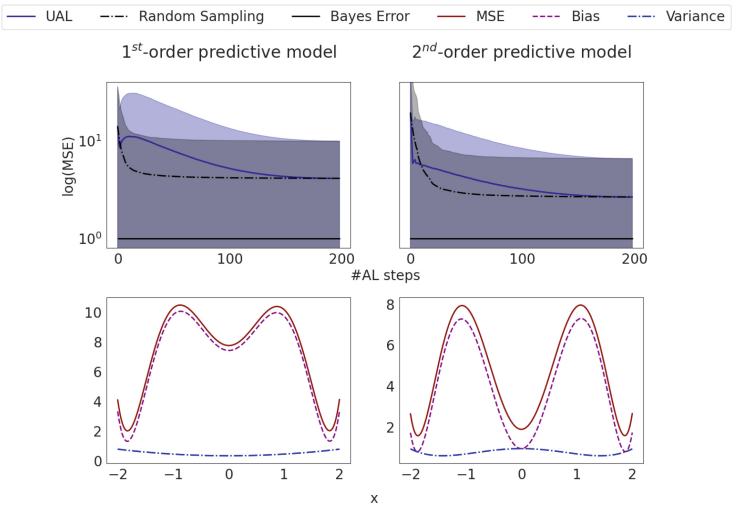


Fig. 5. UAL Performance with the model class of linear and quadratic and their corresponding bias-variance decomposition on BPR over the 3^{rd}-order polynomial family. The first row shows the performance and the second row shows the bias-variance decomposition related to each model class. As both classes are not flexible enough, variance is incapable of capturing the learning objective (MSE); hence UAL performance degrades.

passing the early stages of UAL and observing more samples, random sampling outperforms UAL. Such observed trends suggest in scenarios where the amount of data is extremely small even with model mismatch, UAL in BPR still has short-term efficiency compared to random sampling.

In general, all UAL models perform better than random sampling at the very beginning of UAL for BPR, but as the UAL process progresses, the models whose complexity fall short to match the target's complexity get stuck in the uninformative regions of the input space. This leads to significantly degraded performance of these UAL models compared to random sampling.

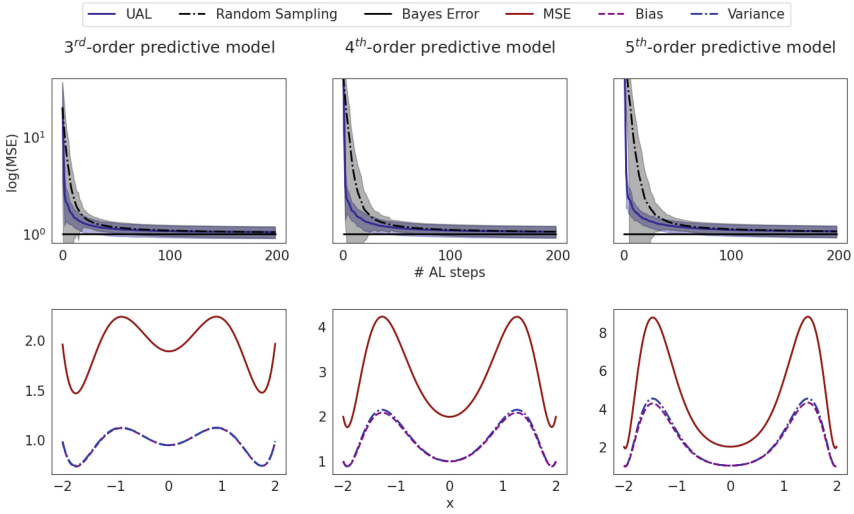


Fig. 6. UAL Performance with the model class of polynomial order three to five and their corresponding bias-variance decomposition on BPR over the 3^{rd}-order polynomial family. The first row shows the performance and the second row shows the bias-variance decomposition related to each model class. As prediction model classes are flexible enough, variance can capture the learning objective (MSE); hence, UAL performance improves.

Gaussian Process Regression. In this set of experiments, we first perform Gaussian Process (GP) regression on the synthetic dataset to further validate the generalizability of our findings with more flexible prediction models. Then we conduct the regression task on two real-world datasets to verify the reliability of our conclusions.

Instead of modeling the correlation of $\mathbf{X}$ and Y with some parameter, GP assumes the outputs are jointly Gaussian distributed with a mean function $m(\mathbf{x})$ and covariance defined by a kernel function $k(\mathbf{x}, \mathbf{x}')$, i.e. $\boldsymbol{f}(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. The conditional distribution $P(\boldsymbol{f}(\mathbf{x})|\mathbf{D_L}) = \mathcal{N}(m_{\mathbf{D_L}}(\mathbf{x}), \sigma^2_{\mathbf{D_L}}(\mathbf{x}))$ is considered to be prediction, where the mean $m_{\mathbf{D_L}}(\mathbf{x}) = m(\mathbf{x}) + \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1}(\mathbf{y} - \mathbf{m})$, the variance $\sigma^2_{\mathbf{D_L}}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*$,

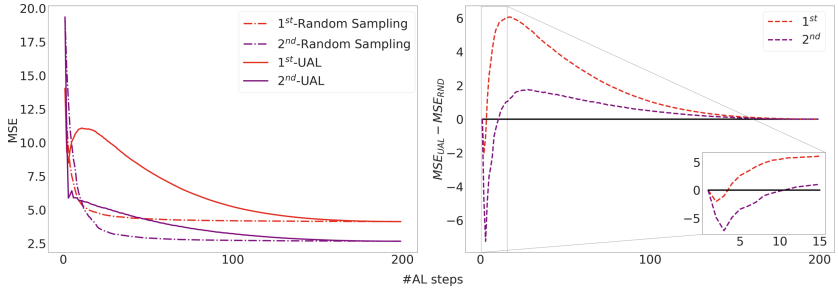


Fig. 7. Performance comparison between UAL and Random sampling for the model class with lower capacity in BPR experiments: the left plot demonstrates when the prediction models' complexity is lower than the ground-truth target function's complexity, UAL gets stuck in querying non-informative regions of the input space after a few early steps; the right plots illustrates the difference between UAL and random sampling MSE (denoted by MSE_{UAL} and MSE_{RND} respectively). In BPR experiments, even models with insufficient complexity have superior performance juxtaposed to random sampling at the early steps of UAL.

$\mathbf{k}_* = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n)]^\top$, $\mathbf{m} = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)]^\top$, the covariance matrix of the observed points $\mathbf{K} = [\mathbf{k}_1, \cdots \mathbf{k}_n]$, and $\mathbf{k}_i = [k(\mathbf{x}_i, \mathbf{x}_1), \ldots, k(\mathbf{x}_i, \mathbf{x}_n)]^\top$.

The selection of the kernel function $k(\mathbf{x}, \mathbf{x})$ plays a critical role in Gaussian Process Regression (GPR) tasks. When choosing linear kernels, GPR is equivalent to BPR with the linear model class. As the kernel becomes more complicated, the model's capacity increases accordingly. Similar to the previous set, for experiments on both synthetic and real-world datasets, GPR with a linear kernel representing low-capacity models is tested. During the experiments on the synthetic dataset, we choose the Matern kernel to account for complex models. For the experiments on real-world datasets, we exploit GPR with a Radial Basis Function (RBF) to represent high-capacity models.

Using the same 100 synthetic 3^{rd}-order ground-truth polynomial functions, as in the BPR experiments, each run starts with one randomly selected pair of input and output corresponding to one 3^{rd}-order polynomial ground-truth function. Subsequently, the performance of GPR with Matern and linear kernel in UAL procedure, as well as the performance of the corresponding random sampling strategy are assessed. Akin to observations in BPR, Fig. 8 illustrates that in GPR with Matern kernels, UAL performs better than naive random sampling which is expected. However, with linear kernels, UAL performs significantly worse than random sampling. Based on the analysis in Sect. 3, when the model's capacity equals or exceeds the ground-truth, the predictive variance captures the actual MSE that leads to the better performance of UAL. However, the limited model class (using linear kernels) with low model capacity results in an inaccurate estimation of MSE. This mismatch leads to a degraded UAL performance.

In real-world case studies, we report the evaluated performance of GPR on the *Facebook Metrics* and *Concrete Compressive Strength* datasets. For *Facebook Metrics*, due to the uncomplicated nature of the dataset, we only report the performance of the simple model in Fig. 9 . Clearly, owing to the simplicity

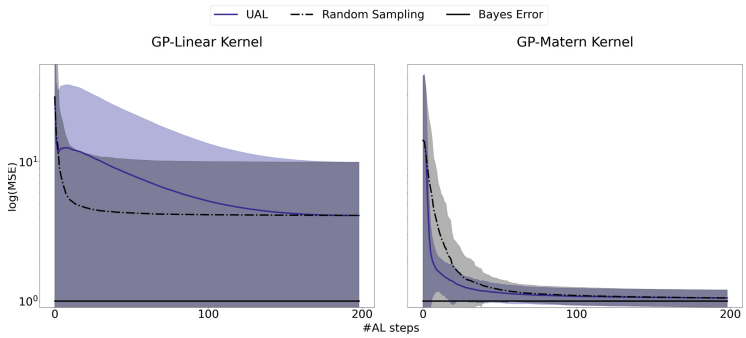


Fig. 8. UAL performance comparison for GP regression with Matern and Linear kernels over the target 3^{rd}-order polynomial family. When using more flexible kernel functions in GP, the ability of variance to faithfully capture the MSE increases; hence, UAL performance improves.

of the regression task on this dataset, the simple model efficiently guides the UAL process and outperforms the naive random sampling strategy. However, for the more challenging regression task on the *Concrete Compressive Strength* dataset, as demonstrated in Fig. 10, the variance-based acquisition function with the simple model fails to direct UAL process, leading to poor performance in comparison to random sampling. However, the variance-based acquisition function with the complex model successfully guides the UAL process and delivers a superior performance than naive random sampling strategy.

Observing UAL performance under model mismatch scenarios and its dependency on the prediction model capacity raises the question of "how to derive efficient UAL in such circumstances". A potential solution is to design acquisition functions approximating the true learning objective, MSE in this study, which opens interesting research opportunities and is the focus of our ongoing work.

5 Conclusions

The performance of UAL, one of the most common pool-based BAL strategies, heavily depends on the adopted acquisition function guiding the sample selection process. When the estimated uncertainty is used to define the acquisition function, the extent to which it can faithfully capture the model prediction performance affects the efficacy of the resulting UAL. We embark on a comprehensive investigation that delves into the analysis of UAL efficacy and the potential mismatch between the prediction model

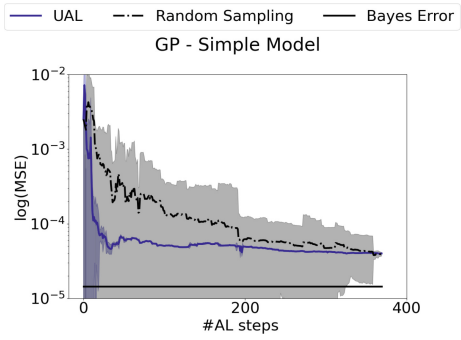


Fig. 9. UAL performance using simple model on the *Facebook Metrics* dataset.

class complexities from the ground-truth target. By analyzing the bias-variance decomposition of prediction error, we showed when the bias due to potential model mismatch dominates the prediction error, UAL may fail, performing worse than training based on random sampling. We conclude that for UAL to perform well, the choice of acquisition function is critical. When the prediction model class complexity aligns with or exceeds the intrinsic complexity of the target, UAL guided by prediction variance can perform better than random sampling. Otherwise, better acquisition functions that directly capture the prediction performance may be needed for effective UAL. We have performed comprehensive empirical evaluation for both BPR and GPR, experimentally validating our theoretical analysis. Additionally, by providing evaluations on two real-world datasets, we further demonstrated the validity of our conclusions. Potential future research directions include theoretical bound analysis on the quantitative relationship between model mismatch and UAL efficacy as well as the design of new acquisition functions based on reliable error estimation for UAL.

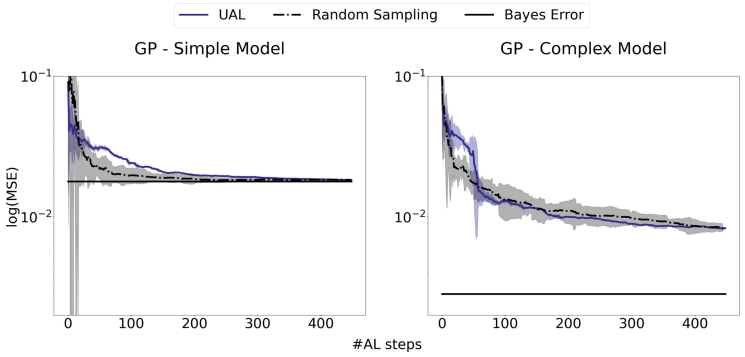


Fig. 10. UAL performance using simple (left) and complex (right) models on the *Concrete Compressive Strength* dataset. It is clear that UAL with the complex model outperforms random sampling while UAL with the simple model is incapable of guiding the model learning procedure efficiently.

Acknowledgement. This work has been supported in part by the U.S. National Science Foundation (NSF) grant IIS-2212419; and by the U.S. Department of Energy (DOE) Office of Science, Advanced Scientific Computing Research (ASCR) M2DT Mathematical Multifaceted Integrated Capability Center (MMICC) under Award B&R# KJ0401010/FWP# CC130, program manager W. Spotz. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing.

References

1. Angluin, D.: Queries and concept learning. Mach. Learn. **2**(4), 319–342 (1988)
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds (2020)
3. Dagan, I., Engelson, S.P.: Committee-based sampling for training probabilistic classifiers. In: Prieditis, A., Russell, S. (eds.) Machine Learning Proceedings 1995, pp. 150–157. Morgan Kaufmann, San Francisco (CA) (1995)
4. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data (2017)
5. Geifman, Y., El-Yaniv, R.: Deep active learning over the long tail (2017)
6. Hacohen, G., Dekel, A., Weinshall, D.: Opposite strategies suit high and low budgets, Active learning on a budget (2022)
7. Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning (2011)
8. Lewis, D.D.: A sequential algorithm for training text classifiers: Corrigendum and additional data. In: ACM SIGIR Forum, vol. 29, pp. 13–19. ACM New York, NY, USA (1995)
9. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Cohen, W.W., Hirsh, H. (eds.) Machine Learning Proceedings 1994, pp. 148–156. Morgan Kaufmann, San Francisco (CA) (1994)
10. Moro, S., Rita, P., Vala, B.: Facebook METRICS. UCI Machine Learning Repository (2016). https://doi.org/10.24432/C5QK55
11. Munjal, P., Hayat, N., Hayat, M., Sourati, J., Khan, S.: Towards robust and reproducible active learning using neural networks (2022)
12. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT Press (2012)
13. Nashaat, M., Ghosh, A., Miller, J., Quader, S., Marston, C., Puget, J.-F.: Hybridization of active learning and data programming for labeling large industrial datasets. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 46–55 (2018)
14. Ren, P., et al.: A survey of deep active learning (2021)
15. Roh, Y., Heo, G., Whang, S.E.: A survey on data collection for machine learning: a big data - AI integration perspective. IEEE Trans. Knowl. Data Eng. **33**(4), 1328–1347 (2021)
16. Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: Proceedings of the 17th European Conference on Machine Learning, ECML'06, page 413-424, Berlin, Heidelberg, Springer-Verlag (2006)
17. Saifullah, S., Agne, S., Dengel, A., Ahmed, S.: Analyzing the potential of active learning for document image classification. Int. J. Doc. Anal. Recogn. (IJDAR), pp. 1–23 (2023)
18. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)
19. Settles, B.: Curious machines: Active learning with structured instances. PhD thesis, University of Wisconsin–Madison (2008)
20. Settles, B.: Active learning literature survey (2009)
21. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 1070–1079 (2008)
22. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5972–5981 (2019)

23. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**, 45–66 (2002)
24. Tür, G., Hakkani-Tür, D.Z., Schapire, R.E.: Combining active and semi-supervised learning for spoken language understanding. Speech Commun. **45**, 171–186 (2005)
25. Wang, D., Shang, Y.: A new active labeling method for deep learning. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 112–119 (2014)
26. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2591–2600 (2017)
27. Wang, W.Y.C., Wang, Y.: Analytics in the era of big data: the digital transformations and value creation in industrial marketing. Ind. Mark. Manage. **86**, 12–15 (2020)
28. Wu, J., Chen, J., Huang, D.: Entropy-based active learning for object detection with progressive diversity constraint (2022)
29. Yeh, I.-C.: Concrete Compressive Strength. UCI Machine Learning Repository (2007). https://doi.org/10.24432/C5PK67

Customizable and Programmable Deep Learning

Ratnabali Pal[1,2], Samarjit Kar[1,3], and Arif Ahmed Sekh[4,5(✉)]

[1] National Institute of Technology Durgapur, Durgapur, WB, India
[2] Brainware University, Kolkata, WB, India
[3] Department of Graphical Systems, Vilnius Gediminas Technical University, Vilnius, Lithuania
[4] XIM University, Bhubaneswar, Odisha, India
[5] UiT The Arctic University of Norway, Tromssa, Norway
skarifahmed@gmail.com

Abstract. In this study, we explore the potential of pre-trained deep learning models, proposing a new approach that emphasizes their reusability and adaptability. Our framework, termed "customizable" deep learning, facilities users to seamlessly integrate diverse pre-trained models for addressing new tasks and enhancing existing solutions. Furthermore, we introduce a "programmable" adapter that enables the flexible combination of different pre-trained modules, expanding the range of applications and customization options. Through empirical experiments, particularly focusing on Visual Question Answering (VQA) for visually impaired (VI) individuals, we demonstrate the practical effectiveness of our methodology. These contributions advance the deep learning field while promoting customization and re-usability across various domains and tasks. The code is available https://github.com/Ratnabali-Pal/CPDA-VQA.

Keywords: Customizable deep-learning · Programmable deep-learning · VQA for VI people

1 Introduction

Researchers continuously propose new deep learning methods [1–4], often building upon existing models with incremental changes. For example, ResNet [5], a widely used model, introduced skip connections to enhance learning. Despite these improvements, the core components of these models, such as perceptions (MLP), convolutional layers, and optimizer, remain consistent. Similarly, attention mechanisms are often fine-tuned to boost accuracy [6] using novel algorithms on existing frameworks.

Post-invention, these deep neural networks are trained on various datasets for validation or deployment. However, many pre-trained models, including large language models (LLMs) and large visual models (LVM), become outdated quickly, leading to wasted resources and environmental concerns due to their high energy consumption [7]. State-of-the-art fusion techniques and transfer learning often rely on pre-trained models [8], but they face challenges such as the lack of suitable baselines for model fusion and pipeline [9]. Such behavior of deep learning methodologies and their incremental advancements poses several challenges:

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 101–116, 2025.
https://doi.org/10.1007/978-3-031-78107-0_7

Rapid Obsolescence of Models: Many pre-trained models become obsolete or are underutilized within months, despite their high environmental costs in energy and carbon emissions [10,11].

Unavailability of Suitable Baselines for Model Fusion: There's a lack of standardized baselines or frameworks to effectively fuse different models or methodologies, which could leverage their complementary strengths for improved performance [12].

Need for Customizable and Programmable Deep Learning: The field lacks methodologies that are easily customizable and programmable to suit specific research needs, making it challenging to adapt existing models to new tasks or datasets efficiently. We argue that if all the available pre-trained models are considered Lego pieces, a particular Lego character (solution) can be built by combining different pieces in many ways. For example, Fig. 1 demonstrates how "a duck" can be built differently using different pieces of Lego in several ways.

To overcome the challenges, we propose a "customizable" and "programmable" deep learning framework that is generalized for solving or improving state-of-the-art tasks. The proposed method consists of three major components **(A)** the pre-trained models for solving different tasks, **(B)** a connection between two or more models, and **(C)** a programmable adapter. The role of the adapter is to mitigate the incompatibility between models or decision-makers.

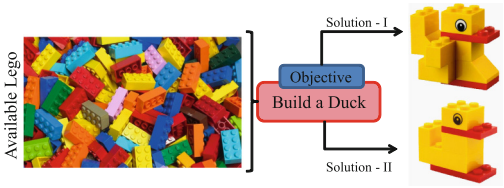


Fig. 1. The Main hypothesis: If we have different colors (purpose) and shapes (compatibility) of Lego pieces (components), it is possible to design new Lego figures (solutions) differently.

This paper introduces a novel deep learning framework designed as programmable and customizable, capable of addressing versatile tasks. The Key contributions of this work include:

Customizability and Programmability: The framework's design allows for the dynamic combination and configuration of deep learning models and adapters, providing adaptability to create customized solutions for diverse applications [13].

Cross-Domain Versatility: Demonstrated through different tasks including a case study on assistive technology for the visually impaired, the framework's adaptability showcases its potential applicability across various fields [14], proving its utility in solving complex, real-world problems.

Superior Performance: By optimizing interactions between models and adapters and leveraging tailored configurations, the framework achieves enhanced accuracy and efficiency, outperforming traditional deep learning approaches in challenging scenarios.

Sustainability: By promoting the reuse and adaptation of pre-trained models and adapters, the framework contributes to the sustainable development of AI technologies, reducing redundancy in training and promoting efficient resource utilization.

These contributions emphasize the framework's potential to revolutionize the approach in different problem-solving within the AI community, offering a customizable, efficient, and sustainable solution for a broad spectrum of tasks.

2 Related Work

In this section, we described the development of "customizable and programmable deep learning". We have discussed **(A)** the different deep learning model fusion methods that promote the customizability and re-usability of deep learning methods, **(B)** the concept of "programmable adapter" to enhance accuracy or solve new tasks, **(C)** Text-based VQA, and **(D)** the problem of VQA for VI people.

Model Fusion: The question of how to fuse representations from different methods and modalities has always attracted different researchers [15]. There exist two major kinds of model fusion, namely decision level (Ensemble) [16], where different methods are combined to make a single decision. On the other side, a feature-level fusion [17] is used to combine different features extracted from different modalities. There exist different modalities of information such as visual, textual, numerical, etc. that can be processed and combined differently to solve a specific task.

Adapter: An adapter is a programmable unit that can attached to a model to solve new tasks or to improve existing tasks. For example, an adapter-based tuning, as demonstrated by the **K-Adapter** [18], offers a versatile approach to fine-tuning pre-trained language models. By introducing lightweight adapter modules, denoted as "K", into existing models, task-specific adjustments can be made without the need for extensive modifications to the original architecture. This adaptability allows for the integration of new knowledge sources while retaining previously learned parameters, enhancing scalability across evolving tasks and data sources.

Similarly, the **VL-Adapter** [19] focuses on efficient transfer learning for vision and language tasks. By incorporating lightweight adapter modules into pre-trained models, VL-Adapter simplifies the adaptation process for specific downstream tasks while minimizing changes to the original model architecture. Notably, VL-Adapter has shown significant performance improvements, particularly in low-shot scenarios, with up to a 10% increase in accuracy. Moreover, automated methods for selecting crucial hyperparameters in VL-Adapter contribute to its accessibility and efficiency across various vision-and-language tasks.

Overall, adapter-based tuning offers a flexible and efficient approach to fine-tuning pre-trained deep models, enabling seamless adaptation to diverse tasks and data sources while preserving the generalization capabilities of the original models. As described in Fig. 2(b), an adapter is used for anomaly detection from CCTV videos by analyzing trajectories extracted by deep neural network [20]. In Fig. 2(b), an adapter is used for semantic segmentation from the binary segmentation [21].

Text-Based VQA: Text-based Visual Question Answering (TextVQA) is a computer vision and natural language processing task that aims to answer accurate questions about images containing textual information. To capture the contextual knowledge of important objects, Li et all. [22] incorporated 3D geometric information into the spatial reasoning process, sequentially. Text-aware pre-training (TAP) [23] has been proposed to integrate scene text to learn a joint representation of text, visual objects, and scene text. TAP specifically includes scene text in pre-training to enhance Text-VQA/Text-Caption. The model is first pre-trained using the scene text-aware pre-training tasks, and it is subsequently fine-tuned for a particular downstream task.

VQA for VI People: Visual Question Answering (VQA) is a helpful tool for VI individuals to understand their surroundings by asking questions and receiving spoken answers. People with visual impairments (VI) rely on technology like smart glasses and phones to navigate the world. These devices capture videos and images that can be converted into spoken answers to questions. Recent advancements in computer vision research have improved VQA systems, making them more accurate and suitable for VI people. The VizWiz dataset is a significant resource tailored specifically for VI individuals, providing a VQA system using the images and questions submitted by them [24,25]. In this area, research efforts have explored the utilization of synthetic data and pre-trained models to enhance VQA performance [26–29]. Synthetic context injection mechanisms have been proposed to improve model performance and facilitate re-usability across different domains. Additionally, the integration of multiple visual data decoding techniques using pre-trained models has been explored to enhance further VQA capabilities. Still, there's space for improvement, especially in developing specialized models specifically designed for VI individuals. More work is needed to ensure that these AI tools effectively meet the needs of VI users.

The concept of model fusion in VQA involves integrating multiple models or techniques within a VQA system to improve its performance or capabilities [30–32]. However, despite these advancements, there remains a lack of specialized VQA system tailored specifically for VI individuals, highlighting a critical gap in current research efforts.

In the following section, we introduce a novel programmable and customizable deep learning framework, designed as a general framework for various applications. We then apply this framework as a case study specifically for Visual Question Answering (VQA) tailored to assist visually impaired (VI) individuals. We validate the effectiveness of this method in addressing the unique needs of VI individuals in accessing visual information through VQA systems.

3 Proposed Method

Let, M be a pre-trained neural network, $M = \{I, O\}$, where I denotes the input and O is the output of the model. The output is either a decision (O_d), or a feature (O_f). Let's say, one image classification neural network is trained on a public dataset. Hence, the input of the model is an image and the output can be the class of the image i.e. (O_d) or extracted feature from intermediate layers (O_f). We, introduce an ADAPTER

(A) is a programmable module. A is defined by $A = \{I, O\}$, similar to a model. Here, $I \in \{O_d, O_f\}$, and O is similar as M. A can be treated as function $A()$, where Input: $\{I\ OR\ O_d\ OR\ O_f\}$, Output: $\{O_d\ OR\ O_f\}$.

Proposed Hypothesis 1: *It is possible to intelligently program an adapter* (A) *such that it can solve a novel problem or can improve a solution.*

Proposed Hypothesis 2: *It is possible to combine different pre-trained models* (M) *, along with some special adapter* (A) *to achieve a solution to a novel problem or to improve a solution.*

Proposed Hypothesis 3: *It is possible to fine-tune individual pre-trained models* (M) *or a group of models in a pipeline without hampering the whole system.*

To support Hypothesis 2, we define four different types of fusion where I, M, O, A: can interact. We restrict in 4 different cases as follows:

Case 1 (Model-to-Adapter): Here, a pre-trained model can be attached with a programmable ADAPTER to solve a new task other than the pre-model is trained to solve. The structure is $I \rightarrow M \rightarrow A \rightarrow O$. In between the input and output, there is one model and an adapter. The information flow between models and ADAPTER is either a decision (O_d) or intermediate feature (O_f). The method is presented in Fig. 2(a). Such a case is proposed in [20], where the authors used the set of object trajectories extracted from videos (pre-trained model) are used to identify anomalies (ADAPTER). The method depicted in Fig. 2(b). Wang et al. [33] proposed similar cases, where deep feature (O_f) passes from models to an ADAPTER. An extracted deep feature using a

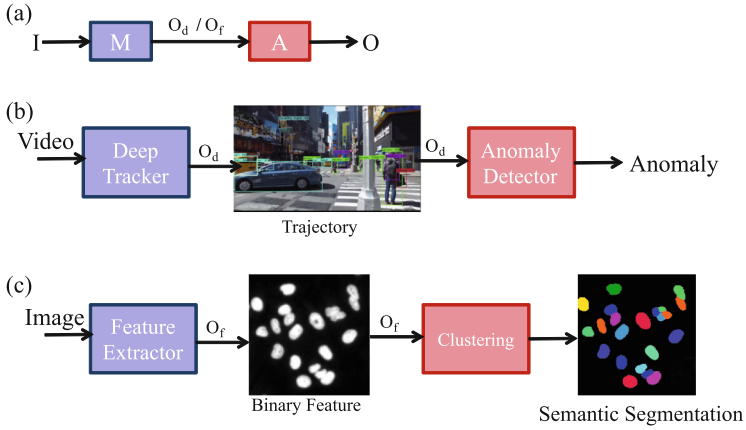


Fig. 2. Case 1: (a) State-of-the-art model-to-adapter pipeline. (b) An example of a case where an object tracking model is used to extract the trajectory and a soft computing ADAPTER is used to detect anomalies in the scene. (c) Example of a case where a pre-trained deep neural network is used to extract the feature for binary segmentation of cell images and a cluster-based ADAPTER is used for semantic segmentation of the cell.

pre-trained model is used in clustering for a cell semantic segmentation as shown in Fig. 2(c).

Case 2 (Adapter-to-Model): Here, a programmable adapter can be attached before a pre-trained model. It can be used to change the input $(I$ to make it more compatible to the model (M). The pipeline is $I \rightarrow A \rightarrow M \rightarrow O$. The constraint is shown in Fig. 3(a). For example, Masoudi et al. [34] demonstrate such methods where an image enhancement module is useful before passing through a neural network as shown in Fig. 3(b).

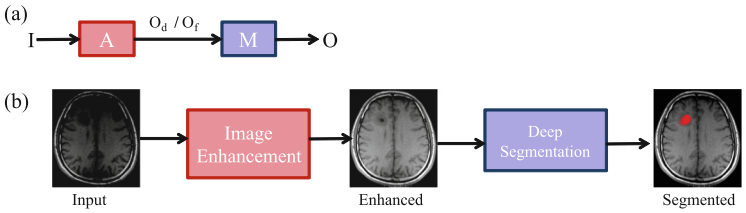


Fig. 3. Case 2: (a) State-of-the-art adapter-to-model pipeline. (b) Example cases when an image enhancer (adapter) is used for an input to a segmentation module (model).

Case 3 (Model-to-Model): A model can directly be attached to another model provided $O_{M_1} = I_{M_2}$. The information flow between models can be O_d or O_f as shown in Fig. 4(a). For example, the problem solved in Case 1 as shown in Fig. 2(a) and 2(b) can

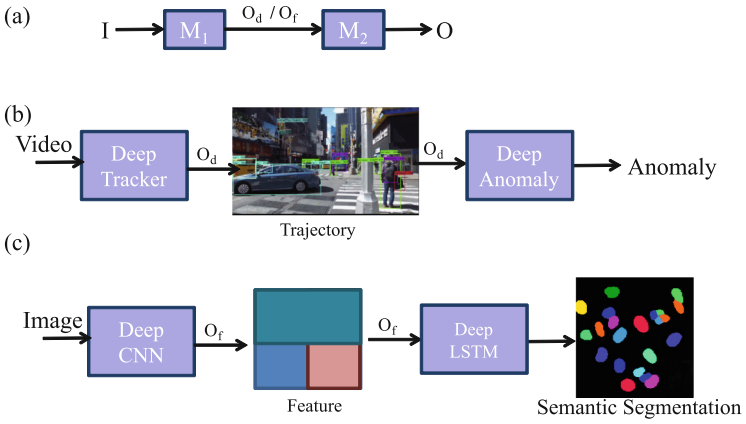


Fig. 4. Case 3: (a) State-of-the-art model-to-model pipeline. (b) Anomaly detection is solved using two deep neural network models. (c) Cell semantic segmentation is solved using CNN and LSTM.

be solved using the model-to-model pipeline. Yu et al. [35] uses a deep neural network for anomaly detection using the trajectory extracted from another deep neural network as shown in Fig. 4(b). Similarly, Saha et al. [21] propose a combination of convolutional neural network (CNN) and long short-term memory network (LSTM) for semantic segmentation as shown in Fig. 4(c).

Case 4 (Models-to-Adapter): In this case, multiple deep neural networks can be added as a pipeline to an adapter. This is $I \rightarrow \{M_1, M_2, M_n\} \rightarrow A \rightarrow O$. Figure 5(a) showcased the method. For example, Islam et al. [36] used two identical deep neural networks (models) and a similarity measurement method (adapter) for image revival. The method depicted in Fig. 5(b). There also exists works [37], where different types of deep neural networks are used to extract different modalities of information and fuse. Another case study [16] uses different decisions taken from different deep neural networks and ensemble them to make a final decision. The authors applied the method on hostile content detection as shown in Fig. 5(c).

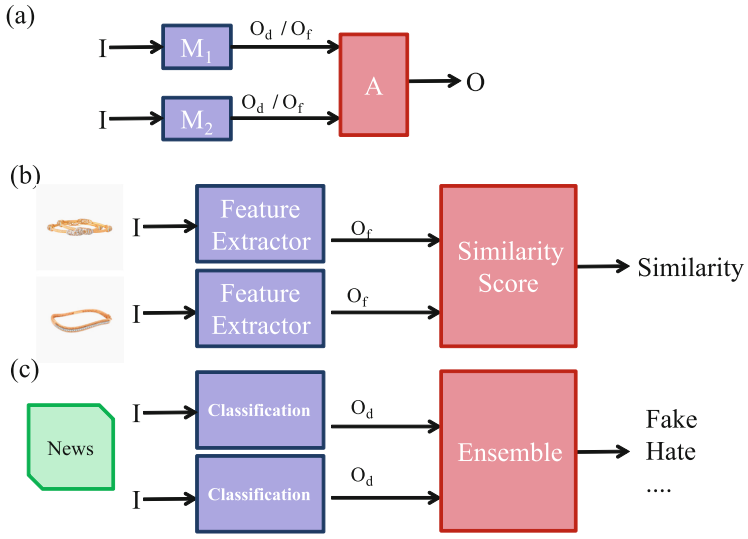


Fig. 5. Case 4: (a) State-of-the-art models-adapter fusion. (b) An example case where two identical neural networks are used for feature extraction and an adapter for fusion. (c) Example of multiple model ensemble.

4 Case Study

Here, we present a case study supporting the hypotheses. We aim to develop an assistive technology for visually impaired (VI) people. We questioned that a given input image or videos (I_{image}/I_{videos}) captured by VI people and a question (Q) generated by the

VI person can be answered. We aimed to develop such a specific VQA for our task. We use the hypothesis presented in the article and consider five pre-trained models having different decision capabilities as presented in Fig. 6(a). Figure 6(b) represents a programmable adapter that can be attached to a model by obeying the constraints defined in the article. Figure 6(c) are the input (image and question text) and (d) represents the expected output (answer). The different pre-trained models participating in the case study are as follows:

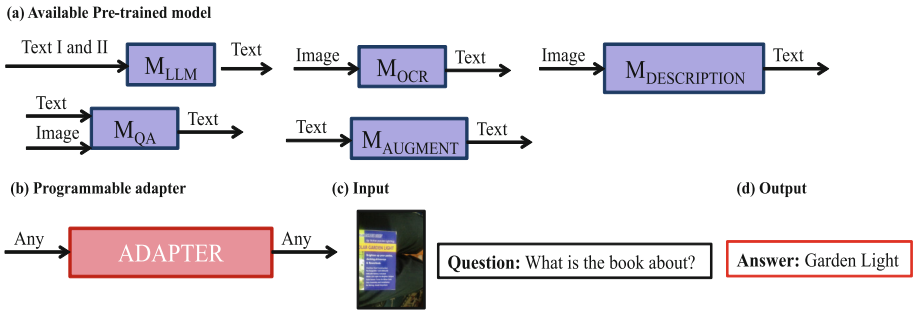


Fig. 6. The components of the proposed hypotheses. (a) is a set of pre-trained models available for the case study, (b) is an adapter that can be programmed, (c) Is the available input, and (d) is the output.

M_{OCR} - This an OCR model capable of extracting text from the image, i.e. $I_{image} \rightarrow O_{text}$. We have used a pre-train lightweight transformer [38].

M_{QA} - is a Vision-and-Language Transformer (ViLT) model pre-trained on GCC+SBU+COCO+VG (200k steps) [39]. It is a kind of $\{I_{image}, I_{text}\} \rightarrow O_{text}$ method used for image question-answering.

$M_{Description}$ - The pre-trained module is used to generate description images, a.k.a. image captioning. we have used CLIP implementation of GPT2 [40].

$M_{Augment}$ - This is a text paraphrasing method used for question augmentation. Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models, or PEGASUS [41] is used as a pre-trained module here. It is a transformer encoder-decoder model.

M_{LLM} - It is a pre-trained language model. The role of the model is to take input a context and question to generate an answer. We use DistilBERT [42] for our experiments.

The **Input:** Images or videos captured by VI people, and the **Output:** Answer asked by the VI people. All the pre-trained models are Open Source and available in HuggingFace[1] platform.

We argue that there are numerous ways to solve the problem using different "configurations" of modules and designing a new "program" for the adapter. To support this, we explain two different solutions to the problem. First, Fig. 7(a) demonstrates a case

[1] https://huggingface.co/.

where a single model can answer the question from the input. In, Fig. 7(b), the identical problem is solved using more complex inputs. We note that the proposed system is permitted to have a different customized solution for the same problem.

A more complex solution can consider all the possible available models. We design an adapter for combining different outputs coming from different models. The adapter plays an important role in transforming incompatible outputs D_{M_1} and inputs I_{M_2} into a compatible format, facilitating their amalgamation through a predefined "template."

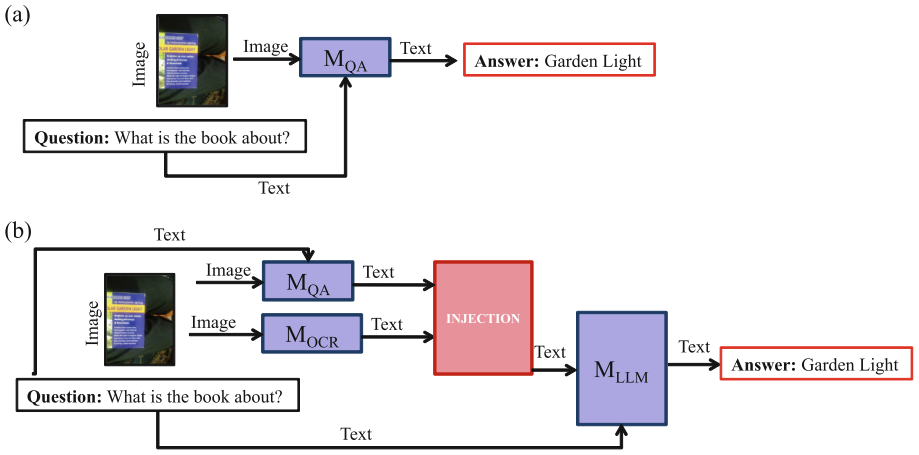


Fig. 7. The different cases of solution of the proposed hypotheses. (a) is a single model solution, (b) is a multimodel solution.

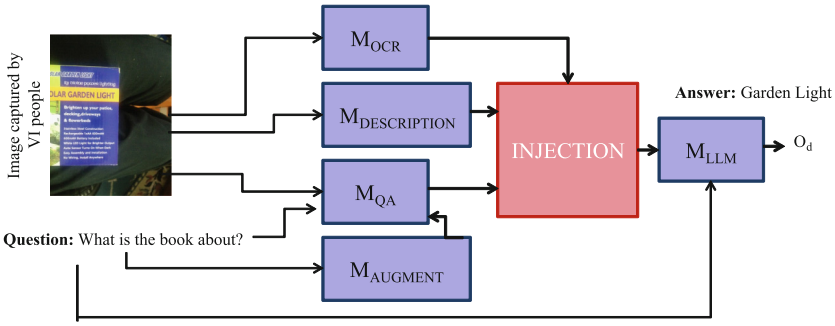


Fig. 8. Proposed customized and programmed pipeline for the VQA for VI people.

Furthermore, we emphasize the adoption of a "master model". We chose an LLM for a leader model who will take an "context" and question, and deliver a QA system. The proposed scientific methodology is visually depicted in Fig. 8. This design provides a competitive solution compared to the state-of-the-art.

```
def INJECTION(I,question):
    O_OCR=M_OCR(I)
    O_augment=M_Augment(question)
    O_QA=M_QA(I,O_augment)
    O_description=M_Description(I)
    context=O_description + ". " + O_OCR + ". " + O_QA
    O_LLM=LLM_QA_baseline(question,context)
    answer=O_LLM['answer']
  return answer
```

Algorithm 1. Algorithm of INJECTION adapter

We programmed the adapter in a straightforward template-based content generation mechanism. The contextual information (C) is formed by concatenating outputs from previously trained models: (C)= "The text presented in the scene is" + (O_{OCR}). + "The scene description is" + $O_{DESCRIPTION}$+ Q_i + O_{ANSWER_i}. Subsequently, the answer is derived by M_{LLM}, incorporating both C and Q. The function is demonstrated in Algorithm 1. The method serves as an effective tool for enhancing content generation in Visual Question Answering (VQA), offering a streamlined approach particularly beneficial for the VI use case.

4.1 Training, Results, and Discussion

Here, we discuss the datasets and training procedure, and the results of different configurations of the case study.

Datasets: Our experimentation encompasses three essential datasets for VI people. **VizWiz Dataset:** Serving as our primary dataset, VizWiz is a unique resource captured exclusively by VI people [24,25]. It comprises a diverse collection of images paired with associated questions, offering a genuine perspective on VI-centric VQA. Figure 9(a) visually illustrates the dataset's contents, highlighting specific hurdles encountered within. **Extended VizWiz Dataset:** To augment our evaluation further, we introduce the "Extended VizWiz" dataset, derived from VizWiz [24,25]. This extension focuses on the questions answerable by extracting text from the images. By curating this subset, we introduce new question-answer pairs, enhancing the dataset's diversity and enabling a more diverse assessment of our solution's capabilities. Figure 9(a) provides illustrative examples of the novel questions posed by this extension, showcasing the dataset's unique challenges. **P100:** As an in-house creation, the P100 dataset offers a novel perspective on VQA tasks within the context of VI people. Comprising short videos captured by VI individuals using smartphones involving handheld products. With approximately 490 videos and 100 distinct products, each video is associated with questions, facilitating a holistic evaluation of our solution's performance. Figure 9(b) offers a glimpse into the dataset's contents, highlighting its diversity and relevance to handheld product interactions.

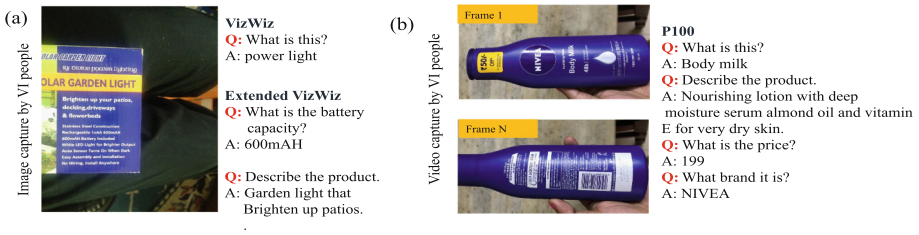


Fig. 9. (a) VizWiz image dataset for VQA. The dataset is extended by including new question-answers namely Ext-VizWiz. (b) P100 video dataset for VQA.

Together, these datasets provide a comprehensive and diverse evaluation framework, ensuring our proposed solution is rigorously assessed across a range of VI-centric VQA tasks and scenarios.

Zero-Shot-Zero-Training: As our method is capable of a zero-shot-zero-training setup, first we compare the proposed method shown in Fig. 8 with a pre-trained visual transformer, namely CLIP [43] using the setup shown in Fig. 7. We use 20% of the data (test samples) for evaluation. The results are demonstrated in Table 1. It is noted that the proposed method performs better compared to a single transformer like CLIP. These results also justify fusing different modalities of information for VQA.

Single-Model and Collaborative Training: According to Hypothesis 3, the model has the capability to train a single module (model) or can be trained jointly. In our The experimental results from re-training the pipeline as demonstrated in Fig. 10(a) and (b). The training dataset comprised 60% of the data, while 20% was used for validation and the remaining 20% for testing across all datasets. The outcomes reveal discernible improvements in final decisions when the re-training process is employed. It is noteworthy that the design of the solution involves a large permutations and combinations. The overarching objective is to strike a balance, maximizing accuracy while minimizing the requisite training data and energy. The experimental results are summarized in Table 2. It is observed that both of the retrain cases, significantly improve the accuracy of the

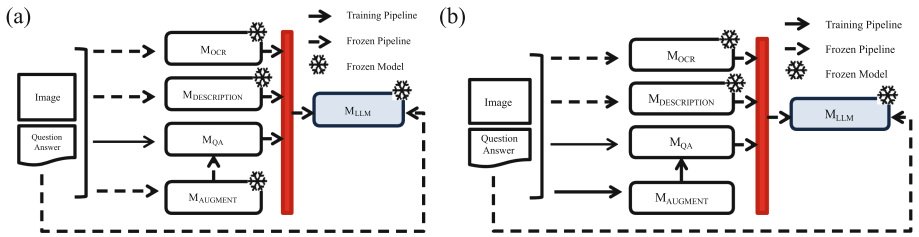


Fig. 10. (a) Single model (M_{QA}) is trained. All other methods are freezes. (b) Multiple models (M_{QA}, M_{Agumet}) are trained.

Table 1. Comparison of zero-shot-zero-training VQA for VI

Method	Dataset	Accuracy	BLEU (Avg.)
CLIP pre-trained [39]	VizWiz	0.20 ± 0.21	0.09
	Ex-VizW	0.06 ± 0.14	0.02
	P100	0.11 ± 0.24	0.02
VL-ADAPTER [44]	VizWiz	0.22 ± 0.19	0.11
	Ex-VizW	0.09 ± 0.21	0.03
	P100	0.11 ± 0.31	0.09
Proposed (zero-shot)	VizWiz	0.38 ± 0.16	0.16
	Ex-VizW	0.59 ± 0.21	0.42
	P100	0.14 ± 0.20	0.11

Table 2. Comparison of results of re-training different modules

Re-train	Dataset	Accuracy	BLEU (Avg.)
M_{QA} [39]	VizWiz	0.42 ± 0.19	0.22
	Ex-VizW	0.11 ± 0.16	0.07
	P100	0.21 ± 0.16	0.14
$M_{QA}M_{Augment}$	VizWiz	0.43 ± 0.17	0.21
	Ex-VizW	0.11 ± 0.11	0.07
	P100	0.22 ± 0.17	0.17
Proposed	VizWiz	0.51 ± 0.11	0.31
	Ex-VizW	0.61 ± 0.24	0.59
	P100	0.32 ± 0.14	0.26

method in the VizWiz dataset, as the method was fine-tuned using the same. Fine-tuning also improves the accuracy over the other two datasets i.e. Ex-VizWiz and P100.

Pre-trained Model Bias: We agree that the proposed method solely depends on the pre-trained model during zero-shot. The final pipeline has a similar bias to the dataset in which it is trained. The final pipeline's accuracy depends on the pre-trained model's individual performance. For example, we replace the OCR module (M_{OCR}) with a recent OCR development namely TrOCR [45], we observed a 1% improvement in zero-shot and 2% improvement in final prediction after re-train in P100. A smaller improvement of 0.5%,1.14% found in Ext-VizWiz and a 0.1%,0.32% improvement in VizWiz dataset.

4.2 Ablation Study

Here, we analyze the individual contributions of components, denoted as M_i, within the proposed system. We systematically removed each participant component (model) from the pipeline, and the results are presented in Table 4. The findings reveal that M_{OCR} exerts the most significant influence on the final output. This observation is consistent with the dataset's bias, where a predominant focus on text in images was maintained during collection. The question-answer module (M_{QA}) has a significant impact on overall accuracy. Whereas, the description generation module $(M_{Description})$ is a little. In contrast, the question augmentation module $(M_{Augment})$ demonstrates a relatively minor influence. These results suggest that introducing new suitable model participants could potentially enhance accuracy further without the need for additional training.

5 Results on Diverse Applications

Here, we discuss the results of the diverse applications over diverse datasets. The concept of programmable and customizable deep learning is used in all the cases. We have reported the results of 6 different scenarios for 4 cases reported earlier (Fig. 2 to Fig. 5). All the proposed methods can be implemented using our framework. Table 3 reports the

Table 3. Results of different applications used the concept of customization and programming new algorithms to solve new problems in deep learning

Method	Application	Dataset	Baseline	Improved	Contributions
Fig. 2(b)	Anomaly [20]	UCF-Crime	82.1% (AUC)	84.48%	Adapter
Fig. 2(c)	Segmentation [21]	Breast Cancer	93.0% (F1)	96.0%	Adapter
Fig. 3(b)	Segmentation [46]	Brain MRI	88.0% (F1)	90.0%	Adapter
Fig. 4(b)	Anomaly [47]	UCSD	90.0% (AUC)	91.1%	Adapters
Fig. 5(b)	FIR [36]	Ornament	14.80% (ACC)	26.74%	Adapters, Fusion
Fig. 5(c)	Classification [16]	Hindi-News	56.0% (F1)	62.0%	Adapters, Fusion

Table 4. Ablation study

Excluded	Dataset	Accuracy	BLEU (Avg.)
M_{OCR}	VizWiz	0.44 ± 0.17	0.22
	Ex-VizW	0.08 ± 0.24	0.06
	P100	0.19 ± 0.20	0.19
$M_{Description}$	VizWiz	0.47 ± 0.14	0.21
	Ex-VizW	0.59 ± 0.24	0.51
	P100	0.21 ± 0.22	0.19
$M_{QA}, M_{Augment}$	VizWiz	0.11 ± 0.24	0.08
	Ex-VizW	0.56 ± 0.12	0.51
	P100	0.17 ± 0.21	0.19
$M_{Augment}$	VizWiz	0.51 ± 0.21	0.29
	Ex-VizW	0.61 ± 0.22	0.59
	P100	0.28 ± 0.11	0.24
None	VizWiz	0.51 ± 0.11	0.31
	Ex-VizW	0.61 ± 0.24	0.59
	P100	0.32 ± 0.14	0.26

results. It is noted that, in all the cases, the fusion or addition of new adapters improved accuracy on specific data and it is evidenced in many datasets.

Q: *What is the use of the product?*
A: nourishing lotion
Context: a person holding a bottle. NIVEA. care. Serum. Almond Oil. Vitamin E, no, Dry skin, no, no,
Answer: a person holding a bottle. Vitamin

Q: *What is the product?* **A:** Orange slices
Context: Packet of berry. Farley's. NETWT 5LB(2.26kg). 655, train, yes, no, yes,
Answer: Packet of berry. Farley's

Q: *What is this?* **A:** green plant
Context: There is a tree, the tub is placed on the floor.
soil, nothing, no, no, train.
Answer: tub.

Fig. 11. Failure cases in P100, VizWiz, and Ex-VizWiz dataset.

5.1 Failure and Limitations

The main limitation of the proposed method is that the pre-trained model comes with its own bias and limitations, the proposed pipeline will also have a similar bias and limitation. For example, the proposed method encounters difficulties addressing scenarios when it is "easy to answer for humans" but difficult for an AI method. for example, an automated method cannot answer simple questions as illustrated in Fig. 11 across various datasets. This is observed among all across different methods and datasets. This limitation arises from the dependence on pre-trained models, highlighting the method's inability to adapt to new data. However, the problem can be solved by fine-tuning the baseline models and we have the option for this.

6 Conclusion

In this study, we have presented a novel approach to deep learning, focusing on the concept of customization and programmability. Our framework is designed to use the potential of pre-trained models in a versatile manner, allowing users to adapt them to various tasks and challenges. By combining different pre-trained models and employing programmable adapters, our framework offers a flexible solution to address a wide range of problems.

Through a detailed case study focusing on Visual Question Answering (VQA) for individuals with visual impairments, we have demonstrated the practical efficacy of our methodology. By leveraging diverse pre-trained models and adapters, we were able to develop a VQA system tailored specifically to the needs of visually impaired users. This system showcases the potential of our framework to promote customizability, programmability, and re-usability in AI technology.

Looking ahead, there are numerous opportunities for further exploration and refinement of our framework. Future research can investigate its application in different domains and tasks, exploring its adaptability and effectiveness across diverse scenarios. Additionally, efforts to enhance the interoperability and compatibility of pre-trained models and adapters can improve the framework's usability and scalability.

Acknowledgements. The authors would like to thank the Science and Engineering Research Board (SERB), Govt. of India for funding the research by the form of the GPU Server. Project Number SRG/2022/000122 executed in XIM University, Bhubaneswar by A. A.

References

1. Demir, F., Akbulut, Y., Taşcı, B., Demir, K.: Improving brain tumor classification performance with an effective approach based on new deep learning model named 3ACL from 3D MRI data. Biomed. Signal Process. Control **81**, 104424 (2023)
2. Tianyu, W., et al.: A brief overview of ChatGPT: the history, status quo and potential future development. IEEE/CAA J. Automatica Sinica **10**(5), 1122–1136 (2023)
3. Team, G., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
4. Marcus, G., Davis, E., Aaronson, S.: A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807 (2022)
5. Koonce, B., Koonce, B.: ResNet 50. Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, pp. 63–72 (2021)
6. Niu, Z., Zhong, G., Hui, Yu.: A review on the attention mechanism of deep learning. Neurocomputing **452**, 48–62 (2021)
7. Savci, P., Das, B.: Comparison of pre-trained language models in terms of carbon emissions, time and accuracy in multi-label text classification using AutoML. Heliyon **9**(5), e15670 (2023)
8. Deng, A., Li, X., Hu, D., Wang, T., Xiong, H., Xu, C.-Z.: Towards inadequately pre-trained models in transfer learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19397–19408 (2023)
9. Wang, H., Li, J., Wu, H., Hovy, E., Sun, Y.: Pre-trained language models and their applications. Engineering **25**, 51–65 (2022)

10. Boyko, J., et al.: An interdisciplinary outlook on large language models for scientific research. arXiv preprint arXiv:2311.04929 (2023)
11. Ooi, K.-B., et al.: The potential of generative artificial intelligence across disciplines: perspectives and future directions. J. Comput. Inf. Syst. 1–32 (2023)
12. Le, D., Keren, G., Chan, J., Mahadeokar, J., Fuegen, C., Seltzer, M.L.: Deep shallow fusion for RNN-T personalization. In: 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 251–257. IEEE (2021)
13. Velasco, L., et al.: End-to-end intent-based networking. IEEE Commun. Mag. **59**(10), 106–112 (2021)
14. Liu, X., Chen, Y., Li, H., Li, B., Zhao, D.: Cross-domain random pre-training with prototypes for reinforcement learning. arXiv preprint arXiv:2302.05614 (2023)
15. Basiri, M.E., Nemati, S., Abdar, M., Asadi, S., Acharrya, U.R.: A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets. Knowl. Based Syst. **228**, 107242 (2021)
16. Chakraborty, A., Joardar, S., Sekh, A.A.: Ensemble classifier for Hindi hostile content detection. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **23**(1), 1–17 (2024)
17. Gao, J., Li, P., Chen, Z., Zhang, J.: A survey on deep learning for multimodal data fusion. Neural Comput. **32**(5), 829–864 (2020)
18. Wang, R., et al.: K-adapter: Infusing knowledge into pre-trained models with adapters. arXiv preprint arXiv:2002.01808 (2020)
19. Pantazis, O., Brostow, G., Jones, K., Aodha, O.M.: SVL-adapter: Self-supervised adapter for vision-language pretrained models. arXiv preprint arXiv:2210.03794 (2022)
20. Thakare, K.V., Sharma, N., Dogra, D.P., Choi, H., Kim, I.-J.: A multi-stream deep neural network with late fuzzy fusion for real-world anomaly detection. Expert Syst. Appl. **201**, 117030 (2022)
21. Saha, M., Chakraborty, C.: Her2Net: a deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation. IEEE Trans. Image Process. **27**(5), 2189–2200 (2018)
22. Li, H., Huang, J., Jin, P., Song, G., Qi, W., Chen, J.: Weakly-supervised 3D spatial reasoning for text-based visual question answering. IEEE Trans. Image Process. **32**, 3367–3382 (2023)
23. Yang, Z., et al.: TAP: text-aware pre-training for text-VQA and text-caption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8751–8761 (2021)
24. Gurari, D., et al.: VizWiz grand challenge: answering visual questions from blind people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3608–3617 (2018)
25. Gurari, D., et al.: VizWiz-Priv: a dataset for recognizing the presence and purpose of private visual information in images taken by blind people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 939–948 (2019)
26. Akula, A., Changpinyo, S., Gong, B., Sharma, P., Zhu, S.-C., Soricut, R.: CrossVQA: scalably generating benchmarks for systematically testing VQA generalization. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 2148–2166 (2021)
27. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077–6086 (2018)
28. Antol, S., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
29. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-OKVQA: a benchmark for visual question answering using world knowledge. In: European Conference on Computer Vision, pp. 146–162. Springer (2022)

30. Siyu, L., Ding, Y., Liu, M., Yin, Z., Yin, L., Zheng, W.: Multiscale feature extraction and fusion of image and text in VQA. Int. J. Comput. Intell. Syst. **16**(1), 54 (2023)
31. Jung, B., Gu, L., Harada, T.: bumjun_jung at VQA-Med 2020: VQA model based on feature extraction and multi-modal feature fusion. In: CLEF (Working Notes) (2020)
32. Jiajia, W., et al.: A multimodal attention fusion network with a dynamic vocabulary for textVQA. Pattern Recogn. **122**, 108214 (2022)
33. Wang, A., et al.: A novel deep learning-based 3D cell segmentation framework for future image-based disease detection. Sci. Rep. **12**(1), 342 (2022)
34. Masoudi, S., et al.: Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. J. Med. Imaging **8**(1), 010901–010901 (2021)
35. Wenhao, Yu., Huang, Q.: A deep encoder-decoder network for anomaly detection in driving trajectory behavior under spatio-temporal context. Int. J. Appl. Earth Obs. Geoinf. **115**, 103115 (2022)
36. Islam, S.M., Joardar, S., Sekh, A.A.: DSSN: dual shallow Siamese network for fashion image retrieval. Multimedia Tools Appl. **82**(11), 16501–16517 (2023)
37. Zhang, Y., et al.: Knowledgeable preference alignment for LLMs in domain-specific question answering. arXiv preprint arXiv:2311.06503 (2023)
38. Du, Y., et al.: PP-OCR: A practical ultra lightweight OCR system. arXiv preprint arXiv:2009.09941 (2020)
39. Kim, W., Son, B., Kim, I.: ViLT: vision-and-language transformer without convolution or region supervision. In: International Conference on Machine Learning, pp. 5583–5594. PMLR (2021)
40. Mokady, R., Hertz, A., Bermano, A.H.: ClipCap: Clip prefix for image captioning. arXiv preprint arXiv:2111.09734 (2021)
41. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)
42. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
43. Song, H., Dong, L., Zhang, W.-N., Liu, T., Wei, F.: Clip models are few-shot learners: Empirical studies on VQA and visual entailment. arXiv preprint arXiv:2203.07190 (2022)
44. Sung, Y.-L., Cho, J., Bansal, M.: VL-adapter: parameter-efficient transfer learning for vision-and-language tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5227–5237 (2022)
45. Li, M., et al.: TrOCR: transformer-based optical character recognition with pre-trained models. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13094–13102 (2023)
46. Ullah, F., et al.: Brain MR image enhancement for tumor segmentation using 3D U-Net. Sensors **21**(22), 7528 (2021)
47. Nawaratne, R., Alahakoon, D., De Silva, D., Xinghuo, Yu.: Spatiotemporal anomaly detection using deep learning for real-time video surveillance. IEEE Trans. Industr. Inf. **16**(1), 393–402 (2019)

SegXAL: Explainable Active Learning for Semantic Segmentation in Driving Scene Scenarios

Sriram Mandalika and Athira Nambiar(✉)

Department of Computational Intelligence, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamil Nadu, India
{mc9991,athiram}@srmist.edu.in

Abstract. Most of the sophisticated AI models utilize huge amounts of annotated data and heavy training to achieve high-end performance. However, there are certain challenges that hinders the deployment of AI models "in-the-wild" scenarios i.e. inefficient use of unlabeled data, lack of incorporation of human expertise and lack of interpretation of the results. To mitigate these challenges, we propose a novel Explainable Active Learning (XAL) model *viz.* '**XAL-based semantic segmentation model "SegXAL"** , that can (i) effectively utilize the unlabeled data, (ii) facilitate the "Human-in-the-loop" paradigm and (iii) augment the model decisions in an interpretable way. In particular, we investigate the application of the SegXAL model for semantic segmentation in driving scene scenarios. The SegXAL model proposes the image regions that require labelling assistance from Oracle by dint of explainable AI (XAI) and uncertainty measures in a weakly-supervised manner. Specifically, we propose a novel Proximity-aware Explainable-AI (PAE) module and Entropy-based Uncertainty (EBU) module to get an Explainable Error Mask, which enables the machine teachers/human experts to provide intuitive reasoning behind the results and to solicit feedback to the AI system, via an active learning strategy. Such a mechanism bridges the semantic gap between man and machine through collaborative intelligence, where humans and AI actively enhance each other's complementary strengths. A novel high-confidence sample selection technique based on the DICE similarity coefficient is also presented within the SegXAL framework. Extensive quantitative and qualitative analyses are carried out in the benchmarking Cityscape dataset. Results show the outperformance of our proposed SegXAL against other state-of-the-art models.

Keywords: Active learning · Explainable AI · Semantic segmentation

1 Introduction

Over the past decade, the world has witnessed an unprecedented technological revolution with the help of Artificial Intelligence (AI) towards accelerating automation, improving decision-making processes, and extracting insights

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 117–134, 2025.
https://doi.org/10.1007/978-3-031-78107-0_8

from vast datasets. Despite these advancements, deep learning models commonly encounter substantial challenges while deploying in real-world or "in-the-wild" settings, such as limitation of well-annotated data, contextual & prior information and interpretability of the results [1].

Annotation of new data points is an expensive and laborious task, yet crucial for enriching training datasets with valuable information. In tasks like image semantic segmentation, manually labelling each pixel with its class label is arduous. Supervised algorithms provide efficient solutions for this task, whereas in unsupervised scenarios, automatic labelling poses a significant challenge for machines. Furthermore, *integrating prior and contextual information* can significantly enhance AI model performance, especially in high-risk scenarios e.g. medical and defence. Domain experts can contribute valuable knowledge to AI systems in such situations, enabling a "Human-in-the-loop" paradigm for more rational analysis and informative results. However, most existing AI systems lack mechanisms to incorporate *additional human-collected information or domain expertise*. In real-world scenarios, the inverse situation also exists, wherein the operators often have to rely on visual inspection to make decisions due to the *lack of explainability in machine decisions*. Despite the advancements in deep neural networks, the integration of AI tools in various fields is hindered by the opacity of these "black-box" models, which fail to provide explanations for their actions. All of these scenarios highlight the semantic gap between human and machine analysis, emphasizing the need for human involvement in decision-making as well as the development of Explainable AI tools towards better interpretability of the model.

To mitigate the aforementioned challenges, we propose a novel Explainable Active Learning (XAL) model that combines domain expert assistance and explainable AI (XAI) support within the active learning (AL) paradigm.

In particular, we propose a novel **XAL based semantic segmentation model "SegXAL"** for the driving scene scenarios. *Active learning* facilitates effective training set by iteratively curating the most informative unlabeled data for annotation with the help of human intervention (oracle) accentuating the "human-in-the-loop" paradigm [8,9]. This "domain expert teaching" emphasizes productivity and enhances trust in AI systems, especially in low-resource as well as high-risk scenarios. Similarly, the *explainability* aspect of the SegXAL model enables the "machine teachers" (human experts) to obtain intuitive reasoning behind the results and to give solicit feedback to the system [12]. This is inspired by the rationale that humans' cognizance leverages causal and interpretable information to make decisions [10,11]. Both of these AL and XAI notions within the SegXAL model bridge the semantic gap between man and machine through collaborative intelligence, wherein humans and AI actively enhance each other's complementary strengths.

The key component of the SegXAL framework is the Explainable Error Mask (EEM) module that provides intuitive reasoning as well as uncertainty measures for the sample selection. The EEM module internally contains two components viz. Entropy-based Uncertainty (EBU) module and Proximity-aware

Explainability (PAE) module. Following popular active learning approaches, the EBU module utilizes uncertainty or disagreement in the unlabeled data to identify the most uncertain and informative samples for annotation by the oracle [2,3,6]. Whereas, the PAE module acts as an interpretable proximity approximator that prioritizes the relevant nearby class information, leveraging depth estimation technique and explainable AI. In particular, two advanced AI models viz. MiDaS [34] and DINOv2 [5] are used as the instances of depth estimation. Refering to the XAI technique, we leverage Gradient-weighted Class Activation Mapping (GradCAM) [35], which interprets and visualizes the regions of an input image that are crucial for the network's prediction of a specific class.

Thus, the PAE module along with the EBU module provides the Explainable-Error Mask (EEM) with both informativeness and explainability, thereby facilitating meaningful annotation from the oracle. Two modes of oracle annotations are presented in this work: The first mode is via **Machine annotated pseudolabels**, wherein the machine itself does an automatic pixel annotation. The second mode is via **Manual annotation**, wherein the human annotator labels the region relevant to the object based on the candidate prompts. The major contributions of the paper are as follows:

- Proposal of a **'XAL based semantic segmentation model "SegXAL" for the driving scene scenarios'**, which is the first Explainable Active Learning (XAL) framework in semantic segmentation.
- Development of a novel **Explainable Error Mask (EEM)**, fusing proximity-aware explainability (PAE) and entropy-based uncertainty (EBU) measures, thereby enhancing the efficiency of oracle annotation.
- Proposal of two manual annotations schemes within the Active learning framework viz. **Manual-M and Manual-D**, leveraging MiDaS and DINOv2-based explainable error masks, respectively.
- Proposal of a novel **high-confidence sample selection technique based on DICE** similarity coefficient.
- Extensive experimental analysis, ablation studies and state-of-the-art comparative analysis in benchmarking Cityscapes dataset.

The rest of the paper is organized as follows: The related works are described in Sect. 2. The proposed SegXAL active learning framework is presented in Sect. 3. The experimental setup and the results are discussed in detail in Sect. 4 and Sect. 5 respectively. Finally, the summary of the paper and some future plans are enumerated in Sect. 6.

2 Related Works

Explainable AI: Explainable Artificial Intelligence (XAI) is an emerging area of research in machine learning [11]. XAI techniques make AI models more interpretable by humans by divulging the hidden "black-box" and providing insights into how the model arrives at a particular decision. Some of the recent research works have been investigating XAI in such cutting-edge areas, e.g. medical

domain to find out the feature importance [25] and to visualize the biologically relevant information [26]. Some XAI models were developed for remote sensing and satellite applications, [27] to analyze synthetic aperture sonar (SAS) data and for Explainable Machine Learning in Satellite Imagery, respectively. The application of XAI approaches in driving scene scenarios is also reported in the recent literature bestowing ideas towards comprehensible and trustworthy autonomous driving technologies [24].

Active Learning: Active Learning (AL) entails the training process of a learning algorithm through an iterative collaboration with a human oracle [17]. AL involves selecting the most relevant data samples from a pool of unlabeled data based on uncertainty, representativeness, or diversity scores computed directly with the model [2,3]. To this end, some popular approaches to obtain confidence, margin and uncertainty measures are via entropy [13], Softmax probabilities [14], Monte Carlo dropout [15] and Ensemble methods [16]. Such AL models have been widely applied in various vision applications, such as medical scenarios [18], satellite imagery analysis [19] etc. The necessity for AL frameworks for autonomous driving scenarios is reported in [20], mentioning that 'vehicles need 11 billion miles of driving (500 years of nonstop driving with a fleet of 100 cars) to perform just 20 per cent better than a human.' Motivated by this notion, some recent AL works on driving scenes were reported in the literature [22].

AL for Semantic Segmentation: There are AL methods specially designed for semantic segmentation that work at image, region or pixel levels [2,3]. The Variational Adversarial Active Learning (VAAL) approach employs adversarial learning to determine whether the latent space signifies labelled or unlabeled data [23]. The work Difficulty-awarE Active Learning (DEAL) [2] incorporates the semantic difficulty to measure the informativeness and select samples at the image level. Another work 'ViewAL' [3] leverages inconsistencies in model predictions across view-points to measure the uncertainty of super-pixels. Yet another work S4AL [21] utilizes pseudo labels generated with a teacher-student framework to identify image regions that help disambiguate confused classes.

Contrary to the aforementioned AL approaches that measure uncertainty/informativeness, our proposed SegXAL additionally augments the notion of explainability in the model. In particular, the PAE module in our proposed SegXAL model imparts contextual and proximity-aware explainability to the oracle to prioritize the annotation of nearby objects, which are pivotal in autonomous driving scenarios. This kind of explainable active learning (XAL) in semantic segmentation is proposed for the first time, to the best of our knowledge. Further, the significance of pixel-level and object-level annotation by the oracle (Machine annotator vs. Human annotator) is also investigated in our proposal.

3 Methodology: SegXAL - Explainable Active Learning for Semantic Segmentation

The Active Learning (AL) protocol ensures that by intelligently selecting instances for labelling, a learning algorithm can achieve good performance with significantly less training data. Formally, it can be expressed as follows: Let (x^l, y^l) be an annotated sample from the original labelled dataset D^L and x^u represent an unannotaed sample from a significantly larger pool of unlabeled data, D^U. The goal of AL is to iteratively query a subset D^S, that contains the most informative n samples $x_1^u, x_2^u, ..., x_n^u$ from D^U in an iterative manner, given n is the fixed labelling budget.

In this work, we present a novel Explainable Active Learning paradigm for semantic segmentation (SegXAL) in driving scene imagery. Refer to Fig. 1 for the overall architecture of the SegXAL framework. It contains training of the model, prediction of semantic maps, "Explainable Error Mask" (EEM) computation, annotation, selection mechanism and retraining steps. Each of these steps is explained in detail in the forthcoming subsections:

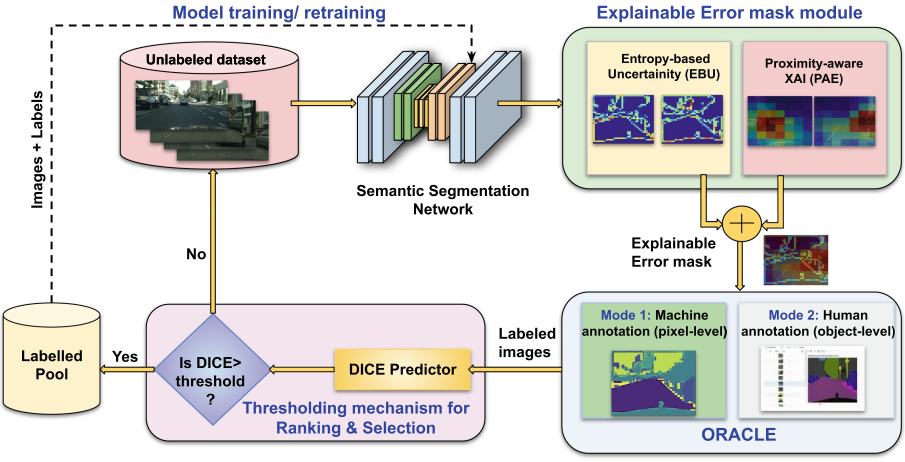


Fig. 1. Visual representation of Explainable Active Learning for semantic segmentation (**SegXAL**) framework. The framework starts with an initial segmentation of unlabeled data, leveraging pre-trained semantic segmentation deep neural network (e.g. U-net). Further, the Explainable Error Mask (EEM) module computes the uncertainty measure and proximity-aware XAI mask. Based on this EEM output, machine/human expert (oracle) makes intuitive labelling feedback to the system. Further, based on the Dice predictor-based query ranking mechanism, reannotated data are used for labeled pool update and model retraining.

3.1 Step 1: Semantic Segmentation Training and Prediction

We leverage U-Net [30] as the semantic segmentation network for the model training. Typically, any segmentation model such as FCN [29] or DeepLab [31], among others, could also be utilized. U-Net is employed in this pilot study, due to its ability for the precise localization of objects while maintaining a high level of contextual information as well as lower memory consumption. The U-Net model embodies an encoder-decoder framework. The encoder is responsible for the initial feature extraction and dimensionality reduction, by utilizing successive convolutional and pooling layers followed by nonlinear activation functions (ReLUs) and batch normalization. Whereas, the decoder works on reconstructing the feature map to the original image size for detailed segmentation using transposed convolutions (or deconvolutions). It also incorporates skip connections, that concatenate feature maps from the contracting path to preserve the high-resolution details that are crucial for accurate segmentation.

In this initial step, a small randomly selected subset of the labelled dataset D_L will be used to train a semantic segmentation network. Following the widely adopted protocol, we randomly sample 10% of the data as labelled data from the train set as our labelled data pool[1]. After training the network on D^L, the model performance is evaluated on unlabeled dataset D^U. AL approach strives to forecast which samples from this unlabeled segment of dataset, are most likely to provide the most informative insights, given the current state of the network. To this end, a novel **Explainable Error Mask (EEM)** module is proposed.

3.2 Step 2: Explainable Error Mask Module

The Explainable Error Mask (EEM) module is the key component of our SegXAL framework. In contrast to the vanilla Active learning models that provide uncertainty/ representativeness insights for the annotation, this novel EEM module presents an explainable error mask for the interactive annotation by the oracle. It consists of the following components: *i) Entropy-based Uncertainty (EBU)*, *ii) Proximity-aware XAI (PAE)* and *iii) fusion of PAE and EBU.*

i) Entropy-Based Uncertainty (EBU) Module:
One of the most important postulations in active learning strategy is to guide the user towards the most relevant areas to annotate, to fix errors. To this end, some standard uncertainty measuring techniques such as entropy [32], or ODIN [33] are exploited in the literature. Following many of the popular AL pipelines, our EBU module leverages entropy metric to measure the uncertainty/ disagreement for the unlabeled data, to obtain the most uncertain data which is informative and worthful ones to be annotated by the oracle.

Entropy is a measure of uncertainty or information content in a probability distribution [32]. In the context of image segmentation, it is commonly used to

[1] (Ablation studies are carried out by varying the splits of labelled data pool i.e. 10%, 15%, 20%, 25%, 30%, 35%, 40%).

quantify the uncertainty of pixel-wise predictions across different classes within a batch of segmented images. Let us denote a batch of segmented images as X with dimensions $[B, C, H, W]$, where B is the batch size, C is the number of classes, H is the height and W is the width of images. Each image in the batch consists of pixel-wise predictions across C classes. The entropy $H(x_{i,j})$ for each pixel $x_{i,j}$ can be calculated as:

$$H(x_{i,j}) = -\sum_{c=1}^{C} P(c|x_{i,j}) \log_2(P(c|x_{i,j})) \tag{1}$$

where $P(c|x_i)$ represents the probability that pixel $x_{i,j}$ belongs to class c. Higher entropy values indicate greater uncertainty or ambiguity in the predictions, implying lower confidence in the model's predictions. Conversely, lower entropy values signify higher confidence or clarity in the predictions.

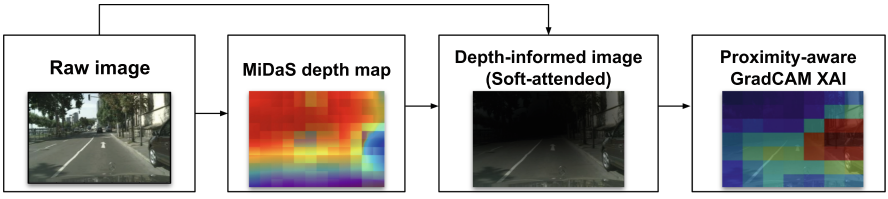


Fig. 2. Proximity-aware Explainable-AI (PAE) Module using MiDaS depth estimation technique. Analogous to MiDaS, DINOv2 depth map is also investigated in this paper.

ii) Proximity-Aware Explainable-AI (PAE) Module: The high entropy pixels generated by the Entropy-based Uncertainty (EBU) module can be spread across the entire image, making it challenging from an Oracle perspective to determine where to prioritize attention. Consequently, this may lead to missing out of some of the vital regions to be annotated first. For instance, in driving scene imagery with high entropy scores in the sky, vegetation, and vehicles, annotation priority should be given to nearby classes i.e. vehicles, considering safety concerns. We hypothesise that such a proximity awareness can improve the oracle annotation. In addition, uncertainty techniques often lack human interpretability, hindering an intuitive understanding of why certain regions are crucial for annotation.

Based on the aforesaid rationale, we propose a novel **Proximity-aware Explainable-AI (PAE)** module to mitigate the priority and interpretability concerns. The PAE module is capable of focusing on the key objects and regions of interest in the proximity regions with the help of an explainability heatmap. The working pipeline of our proposed PAE module is depicted in Fig. 2. Either MiDaS or DINOv2 model is leveraged to obtain the given image's relative depth map. MiDaS [34] is a robust monocular depth estimation technique that employs mixed-dataset training to create a robust and generalizable

depth estimation model. Whereas, DINOv2 [5] is a self-supervised vision transformer model that uses a teacher-student architecture to provide object-level feature extraction. Both of the models are capable of providing monocular depth map outputs. By integrating the MiDaS/DINO-v2 patchwise depth map with the raw image using a thresholding mechanism, the proximity coverage will be estimated. This results in a depth-informed or soft attention image as shown in Fig. 2. Note that the threshold for generating a depth-informed image varies with each image based on the proximity of the nearest objects. Upon this image, a Gradient-weighted Class Activation Mapping (GradCAM) [35] explainability map is applied to visualize the important objects and regions. GradCAM is a technique for visualizing CNN decisions, highlighting regions crucial for predictions. The mathematical equation for GradCAM activation at spatial position (i, j) for class c i.e. $Grad - CAM^c_{i,j}$ can be summarized as:

$$\text{GradCAM}^c_{i,j} = \text{ReLU}\left(\sum_k \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial f_k(i,j)} \cdot f_k(i,j)\right), \tag{2}$$

where, y^c is the output score for class c before softmax, $f_k(i, j)$ is the activation value of the k^{th} feature map at spatial position (i, j) and Z is the normalization constant, typically sum of positive gradients. By applying the GradCAM upon the depth-informed image, we obtain the proximity-aware GradCAM explainability map i.e. $ProxGradCAM^c_{i,j}$, which prioritizes the object class information which is relevant in the proximity region.

iii) Fusion of PAE and EBU Modules: The PAE heatmap $ProxGradCAM^c_{i,j}$ is further fused with EBU uncertainty heatmap $H(x_{i,j})$, to obtain the Explainable Error mask $EEM_{i,j}$. Formally,

$$EEM_{i,j} = \alpha \cdot ProxGradCAM^c_{i,j} + \beta \cdot H(x_{i,j}) \tag{3}$$

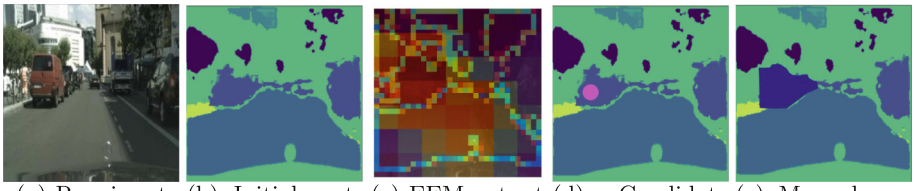
where α and β are the weights for the $ProxGrad - CAM^c_{i,j}$ and $H(x_{i,j})$, respectively. Albeit we used equal contribution for the weights in this work, it can be made learnable.

3.3 Step 3: Oracle for Annotation

Next, we acquire labels for the superpixels/Region of Interest (ROI) selected by EEM module, with the help of oracle. In particular, two modes of oracle annotations are envisaged in this work: machine and human oracle. In the former mode (Machine oracle), automatic pixel annotations are simulated by the machine itself. We term these annotations as '**pseudolabels**'. In the latter mode (Human oracle), the reannotations are carried out manually by a domain expert. By keeping the interpretable information of the potential error map obtained from EEM as a reference, the annotation process is carried out using tools like Label Studio[2]. Specifically, two manual annotation schemes are devised within the Active

[2] Label studio: https://labelstud.io/.

learning framework viz. **Manual-M** and **Manual-D**, leveraging MiDaS and DINOv2-based explainable error masks, respectively.



(a) Raw input (b) Initial output (c) EEM output (d) Candidate prompt (e) Manual annotation

Fig. 3. Oracle's Reannotation workflow. The magenta point shown in 3(d) is the EEM output prompt corresponding to the relevant object candidate to be annotated.

Figure 3 depicts a sample human oracle-based reannotation workflow. Based on the initial segmentation mask output from the raw image as shown in Fig. 3(b), EEM produces the output $EEM_{i,j}$ (Refer Fig. 3(c)). Further, based on the object candidate prompt as shown 3(d), the human annotator corrects the miss-segmented image regions by providing object-level annotation (Fig. 3(e)). These newly reannotated segmentation masks will be further fed into the sample selection module towards the next iteration of the AL loop.

3.4 Step 4: Thresholding Mechanism for Sample Selection

After the oracle, the labeled images are fed into the Ranking & Selection module. Analogous to the high-confidence sample selection techniques as in [14], we use a novel thresholding mechanism to select high-confidence samples to be incorporated into the labeled data pool. In particular, a standard evaluation metric i.e. 'DICE predictor' is utilized to compute the quantitative measure of performance of the segmented images. Mathematically, DICE computation can be written as:

$$\text{DICE} = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{4}$$

where A represents the segmented image and B denotes the reannotated pseudo labels/ human annotations, within each AL cycle.

This **DICE predictor-based sample selection strategy** is devised based on the assumption that *"in every AL cycle, the oracle contributes a significant amount of annotation to improve the quality of semantic segmentation"*. Based on this intuition, we postulate that whenever the similarity between the segmented image and the reannotated image becomes high, a convergence is achieved in the segmentation result. In other words, even after a significant amount of contribution from the oracle, the segmentation result does not improve further, which can be observed as an increase in the DICE similarity coefficient. To guarantee the reliability of high-confidence sample selection, at the end of each iteration,

this DICE value is compared against a predefined threshold θ. If the DICE score is above θ, select it and add to the labeled pool and clear it from the unlabeled set; otherwise, feed it back to the unlabeled dataset placed in the unlabelled pool for potential future iterations.

3.5 Step 5: Iterative Active Loop for Semantic Segmentation Improvement

After the Ranking & Selection module, high-confidence segmentation images are added to the labelled data pool D^L, as shown in Fig. 1. Based on this updated dataset, the semantic segmentation model retraining will be carried out. This concludes a complete active learning cycle. Further, a new AL cycle will start based on the updated model weights and the unlabeled dataset D^U. All the series of steps - *Semantic map prediction from unlabelled data, EEM computation, Annotation, Ranking & Selection and Retraining* - are repeated until the labelling budget is reached or all the data is labelled. This iterative AL cycle optimally selects the most informative samples via EEM information and Oracle annotation, enhancing model performance with minimal labelling costs.

4 Experimental Setup

Dataset: We evaluate our proposed SegXAL framework on the Cityscapes dataset for semantic segmentation [28]. Cityscape is a large-scale benchmark for urban street scene understanding, at 1024×2048 pixel resolution with 30 classes including road, car, pedestrian, bicycle, traffic sign, and more. The dataset is divided into three subsets: *train* (2975 images), *validation* (300 images), and *test* (500 images). We follow the widely adopted protocol for the dataset - we sample 40% of the data from the trainset as our labelled data pool D^L for initial training then iteratively query 5% new data from the remaining training set, which is used as the unlabeled data pool D^U. Considering samples in the street scenes have high similarities, we first randomly choose a subset D^S from the entire pool of D^U, then query n samples from the subset.

Evaluation Protocol: We evaluate our proposed SegXAL model using the standard segmentation evaluation metrics i.e. Intersection over Union (**IoU**) and DICE coefficient. To assess the accuracy of pixel-wise classification, the standard evaluation metric IoU (Intersection over Union) score is utilized. IoU is computed as the ratio of the intersection and union of the ground truth mask and the predicted mask for each class. Further, the DICE similarity coefficient is utilized for ranking & selection of samples, as described in Sect. 3.4. It provides a balanced measure of segmentation accuracy, especially in cases of class imbalance, and hence is used for our sample selection strategy.

Implementation Details: The images with a dimension of 256×512 are normalized using the RGB mean and standard deviation of ImageNet before passing to the network. Our baseline UNet model was evaluated using a stratified K-fold cross-validation approach to ensure robustness and generalizability. The network is trained using a Stochastic Gradient Descent (SGD) optimizer with the following hyper-parameters: $\beta 1 = 0.9$, $\beta 2 = 0.999$, batch size = 16, initial learning rate = 0.0001. The batch size used is 16 images. For all methods and the upper bound method with the full training data, we train 100 epochs with an unweighted cross-entropy loss function. The proposed method is implemented using the PyTorch framework. The implementation was done in a machine with NVIDIA DGX A100 GPU with 24GB RAM and takes around 8 h to train the model.

5 Experimental Results

5.1 Evaluation Results

To verify the effectiveness of our proposed SegXAL framework, various quantitative and qualitative analyses are carried out in the Cityscape dataset. The mean Intersection over Union (mIoU) at each AL stage i.e. 10%, 15%, 20%, 25%, 30%, 35%, 40% of the full training set are adopted as the evaluation metric. Every method is run 5 times and the average mIoUs are reported.

Refer to Table 1 for the per-class IoU and mIoU for each method at the fifth AL cycle, using 40% training data in the Cityscapes dataset. Compared to other popular approaches such as DEAL [2] and Core-set [36], SegXAL is found to be

Table 1. Class-wise IoU and mIoU on Cityscape dataset with 40% training data. For clarity, only the average of 5 AL runs are reported, and the best and the second best results are highlighted in **bold** and *italics*.

Method	Road	Sidewalk	Building	wall	Fence	Pole	Traffic Light	Traffic sign	Vegetation	Terrain	
Fully-supervised	**97.58**	**80.55**	**88.43**	**51.22**	**47.61**	35.19	**42.19**	**56.79**	**89.41**	**60.22**	
Random [21]	96.03	72.36	86.79	43.56	44.22	36.99	35.28	53.87	86.91	54.58	
Entropy [21]	96.28	73.31	87.13	43.82	43.87	*38.10*	37.74	55.39	87.52	53.68	
Core-Set[36]	96.12	72.76	87.03	44.86	*45.86*	35.84	34.81	53.07	87.18	53.49	
DEAL [2]	95.89	71.69	87.09	45.61	44.94	**38.29**	36.51	55.47	87.53	56.90	
Ours (Pseudolabels)	96.67	72.42	87.04	*46.91*	45.02	36.26	*37.83*	56.11	*87.93*	57.54	
Ours (Manual-M)	96.91	72.68	87.44	46.62	45.22	35.62	36.24	*55.78*	87.66	57.86	
Ours (Manual-D)	*96.98*	*73.43*	*88.34*	46.88	45.38	36.12	37.36	55.38	87.84	*59.87*	
Method	Sky	Pedestrian	Rider	Car	Truck	Bus	Train	Motor Cycle	Bicycle	mIoU	STD
Fully-supervised	*92.69*	**65.12**	37.32	**90.67**	**66.24**	**71.84**	**63.84**	**42.35**	**61.84**	**65.30**	19.48
Random [21]	91.47	62.74	37.51	88.05	56.64	61.00	43.69	30.58	55.67	59.00	20.61
Entropy [21]	92.05	63.96	34.44	88.38	59.38	64.64	*50.80*	36.13	*57.10*	61.46	20.14
Core-Set[36]	91.89	62.48	36.28	87.63	57.25	*67.02*	56.59	29.34	53.56	60.69	20.61
DEAL [2]	91.78	*64.25*	**39.77**	88.11	56.87	64.46	50.39	*38.92*	56.59	61.64	19.41
Ours (Pseudolabels)	92.18	62.53	38.82	*88.61*	59.07	65.72	47.12	35.41	55.83	63.56	20.12
Ours (Manual-M)	92.84	62.73	*39.34*	87.97	59.43	66.01	46.92	34.98	54.93	64.37	19.96
Ours (Manual-D)	**92.93**	62.56	*39.07*	88.11	*59.47*	65.70	46.88	35.53	54.71	*65.11*	20.15

outperforming in overall mIoU (Pseudolabels-63.56; Manual-M -64.37; Manual-D -65.11), as well as on various classes, such as road, building, wall, traffic light, traffic sign, vegetation, terrain, sky, rider, car and truck. Furthermore, between the two modes of oracle annotation i.e. Pseudolabel vs Manual, we observe that the manual mode outperforms with a 0.8% increase against the former, and has a significant boost in class-wise IoUs. We also provide a statistical measure of standard deviation (STD) to give an insight into the variability of the model performance. Further, Table 2 displays the incremental trend of mIoU values over multiple iterations. It is observed that at the end of 5 AL cycles itself, mIoU is improved from 20.71 to 63.56 using Pseudolabels, 23.62 to 64.37 using Manual-M and 24.24 to 65.11 using Manual-D.

Table 2. Comparison of mean IoU of SegXAL model over 5 active learning cycles, with 40% training data, using pseudo labels annotated by machine vs and human annotations using MiDaS (Manual-M) and DINOv2 (Manual-D) variants.

Mode	ALcycle1	ALcycle2	ALcycle3	ALcycle4	ALcycle5
Pseudolabel	20.71	27.11	39.23	50.47	63.56
Manual-M	23.62	28.02	39.11	51.33	64.37
Manual-D	24.24	30.02	39.96	52.31	65.11

5.2 Visualisation Results

To demonstrate the efficacy of our proposed EEM module, we visualize the qualitative results. Referring to Fig. 4, the visualization of 5 AL cycles of a sample raw image shown in Fig. 2 are depicted column-wise. The pixel entropy, Explainable Error Mask (EEM) output and the machine annotated pseudolabel-based segmentation results are shown along the first, second and third rows respectively.

Referring to Fig. 4 (a)-(e), high entropy areas represented in red or orange patches indicate a high degree of variability in pixel values. Conversely, low entropy regions, in blue, signify homogenous or less complex segments, where pixel intensities are similar and are in their class boundaries. This entropy map thus serves as a useful visualisation tool to analyse the complexity of the scenes over the loops. Further, Proximity-aware GradCAM-XAI is fused with this entropy mask to obtain an Explainable Error Mask as shown in Fig. 4 (f)-(j)(Refer Sec. 3.2). These EEM outputs clearly "*explain*" the oracle to focus and prioritise the annotation of the closer objects/regions with *high entropy*, which are quite critical in decision-making in the real-world scenario. The oracle-annotated results Fig. 4 (k)-(o)depicts the significant improvement in segmentation quality over 5 active learning cycles.

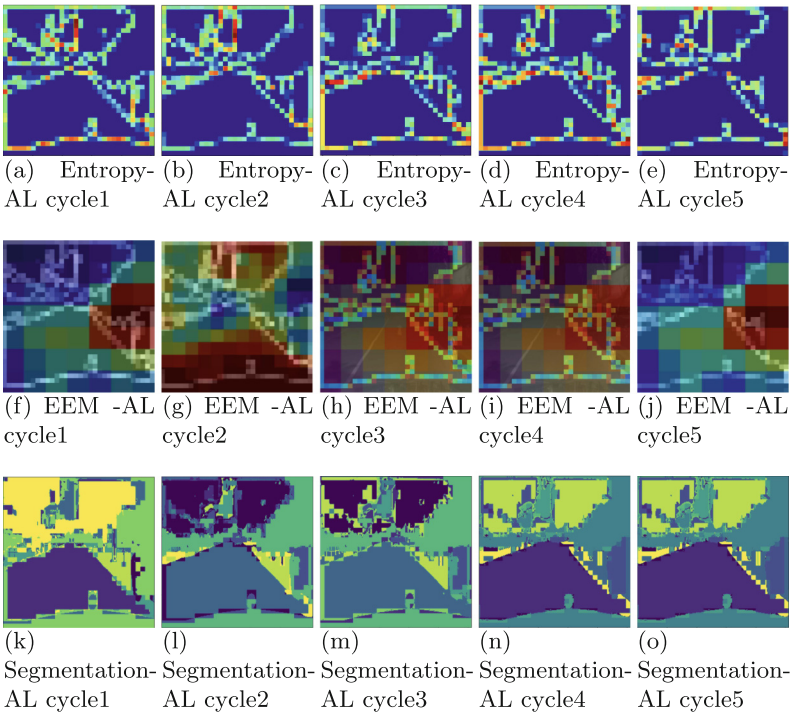
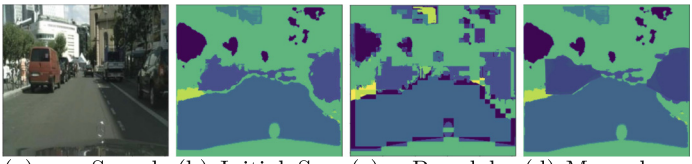


Fig. 4. Visualization of model performance over 5 Active Learning cycles.

5.3 Ablation Study

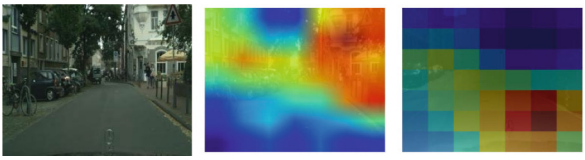
I) Impact of Machine Based Pseudo Label Annotation vs Manual Annotation/ Impact of Pixel level strategy and object level strategy: In this ablation study, we analyse the effect of Machine-based pseudo label reannotation and Manual reannotation. As mentioned earlier, the machine oracle mode leverages pixel-level pseudolabel values for annotation whereas the human oracle employs object-level annotation via Label Studio. We could observe from Table 1, Table 2 and Fig. 5 that both approaches provide superior performance in semantic segmentation. Specifically, the manual annotation outperforms the Pseudolabel annotation (Refer Table 1) and smooth segmentation masks (See Fig. 5). Nevertheless, Machine-based auto labelling is faster and bestows a promising automated AL solution from a practical perspective compared to manual annotation, wherein a human expert reviews every image and reannotates.

ii) Impact of Proximity-Aware XAI EBU and PAE Modules: To understand the impact of EBU and PAE modules, quantitative ablation studies are carried out. Referring to Table 3, it can be observed that the lack of EBU submodule within the EEM block results in a mIoU drop of 3.69, 3.94 and 4.02



(a) Sample Raw Image (b) Initial Segmentation (c) Pseudolabel@AL1 (d) Manual annotation@AL1

Fig. 5. Visualization of Machine-based Pseudolabel vs. Manual annotation outputs



(a) Image (b) GradCAM (c) PAE

Fig. 6. Visual representation of Proximity-aware XAI (PAE)

in Pseudolabel, Manual-M and Manual-D cases, respectively. Its counterpart results in the absence of PAE sub-modules are 3.47, 2.69 and 3.4 respectively. Additionally, a qualitative study is also conducted to comprehend the visual interpretation of Proximity-aware XAI, as depicted in Fig. 6. It is observed that PAE outperforms the Vanilla GradCAM [35], which provides insights of the scene by localizing on the key areas semantic classes via saliency heat maps (Refer Fig. 6(a, b)). Built on top of this Grad-CAM concept, our Proximity-aware XAI module refines the attention further onto the nearby objects in the proximity regions e.g. nearby vehicles and sidewalks, as shown in Fig. 6(c). This PAE enhancement notably fosters safety and transparency in autonomous driving scenarios.

Table 3. A quantitative study on impact of EEM module and their components.

Mode	PsuedoLabels	Manual-M	Manual-D
With EEM	63.56	64.37	65.11
Without EBU	59.87	60.43	61.09
Without PAE	60.09	61.68	61.87

iii) Impact of Change in % of Data Split In this ablation study, we investigate the effect of data split on the SegXAL performance. In particular, we perform various splits of 10%, 15%, 20%, 25%, 30%, 35%, and 40% of the dataset for the initial model training. Referring to Fig. 7 showing the fifth AL cycle mIoU

result, it can be observed that based on the increase of labelled data from 10% to 40%, there is a significant increase in mIoU for Pseudolabel 51.02 to 63.56, Manual-M 52.29 to 64.37 and Manual-D 52.83 to 65.11.

5.4 State-of-the-Art Comparison

We compare SegXAL with other Active Learning-based semantic segmentation approaches that are deployed on the Cityscapes dataset under similar conditions (with 40% training data over 5 AL cycles) i.e. DEAL [2], core-set approach [36], random, entropy [2,21] and QBC [2]. Although another recent study S4AL [21] achieves a competitive result of mIoU 64.80, it is not included in the comparison due to its different setting of 16% training data. Referring to the results as shown in Table 1 and Fig. 7, it can be observed that SegXAL outperforms the state-of-the-art approaches with a significant margin, achieving the best result of 65.11 mIoU with human annotations with DINOv2 depth map (blue-dotted line). It is also observed from Table 1 that, the segmentation performance on the nearby classes such as road (96.98), sidewalk (73.43), wall (73.48) and vehicles such as truck (59.47), rider (39.34) are better or on par with the previously proposed methods. This superior performance could be accredited to the Explainable Error Mask module that facilitates object-level proximity mechanism using XAI attention and Entropy metric, which prioritizes the highly informative nearby objects' annotations compared to far away objects such as train, vegetation etc.

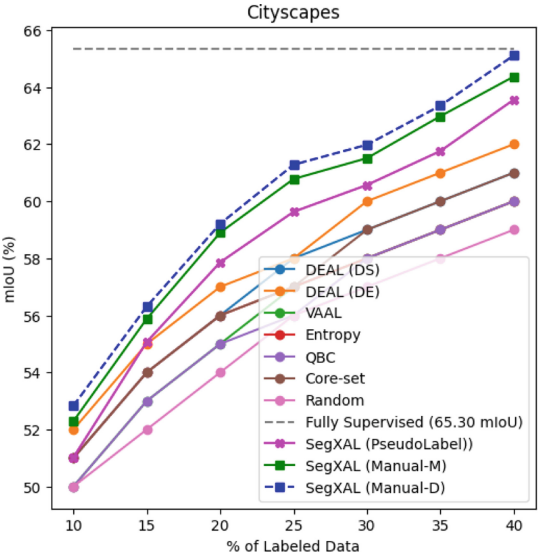


Fig. 7. SegXAL performance against state-of-the-art on the Cityscapes dataset with 40% training data. Every method is evaluated at the end of 5 AL cycles.

6 Conclusions and Future Work

In this work, we proposed a novel Explainable Active Learning framework viz. SegXAL for semantic segmentation. A pilot study on the application of the SegXAL model for driving scene semantic segmentation is presented in this paper. In contrast to most of the existing Active learning methods that annotate using uncertainty information, the proposed model additionally "explains" the proximity region of interests and key objects to be prioritized while annotating by the oracle, with the help of a newly proposed Explainable Error Mask (EEM) module. Such XAI heatmap explanations not only improve the segmentation accuracy but also bridge the semantic gap that exists between human and machine interpretation. Our SegXAL model outperforms state-of-the-art results. Future improvements can be made by introducing better attention mechanisms such as Vision transformers and extending the applications to other driving datasets and other domains.

References

1. Khoei, T., Tala, H.O.S., Kaabouch, N.: Deep learning: systematic review, models, challenges, and research directions. Neural Comput. Appl. **35**(31), 23103–23124 (2023)
2. Xie, S., Feng, Z., Chen, Y., Sun, S., Ma, C., Song, M.: DEAL: difficulty-aware active learning for semantic segmentation. In: Asian Conference on Computer Vision (2020)
3. Siddiqui, Y., Valentin, J., Nießner, M.: ViewAL: active learning with viewpoint entropy for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9433–9443 (2020)
4. Ranftl, R., et al.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 1623–1637 (2019)
5. Oquab, M., et al.: DINOv2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
6. Lenczner, G., Chan-Hon-Tong, A., Le Saux, B., Luminari, N., Le Besnerais, G.: DIAL: deep interactive and active learning for semantic segmentation in remote sensing. IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. **15**, 3376–3389 (2022)
7. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
8. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Machine Learning Proceedings 1994, Morgan Kaufmann, pp. 148–156 (1994)
9. Margatina, K., et al.: Active learning by acquiring contrastive examples. arXiv preprint arXiv:2109.03764 (2021)
10. Rottman, B.M., Hastie, R.: Reasoning about causal relationships: inferences on causal networks. Psychol. Bull. **140**(1), 109 (2014)
11. Yang, S.C.-H., Folke, N.E.T., Shafto, P.: A psychological theory of explainability. In: International Conference on Machine Learning, PMLR (2022)

12. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020)
13. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (2008)
14. Wang, K., et al.: Cost-effective active learning for deep image classification. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2591–2600 (2016)
15. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: International Conference on Machine Learning, PMLR (2017)
16. Beluch, W.H., et al.: The power of ensembles for active learning in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
17. Settles, B.: Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences (2009)
18. Liebgott, A., et al.: Active learning for magnetic resonance image quality assessment. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2016)
19. Goupilleau, A., Ceillier, T., Corbineau, M.-C.: Active learning for object detection in high-resolution satellite images. arXiv preprint arXiv:2101.02480 (2021)
20. Shapiro, D.: What Is Active Learning? NVIDIA blog (2020). https://blogs.nvidia.com/blog/what-is-active-learning/
21. Rangnekar, A., Kanan, C., Hoffman, M.: Semantic segmentation with active semi-supervised learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2023)
22. Schmidt, S., et al.: Advanced active learning strategies for object detection. In: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE (2020)
23. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
24. Atakishiyev, S., et al.: Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. arXiv preprint arXiv:2112.11561 (2021)
25. Zuallaert, J., et al.: SpliceRover: interpretable convolutional neural networks for improved splice site prediction. Bioinformatics **34**(24), 4180–4188 (2018)
26. Rajpurkar, P., et al.: AppendiXNet: deep learning for diagnosis of appendicitis from a small dataset of CT exams using video pretraining. Sci. Rep. **10**(1), 3958 (2020)
27. Stomberg, T., et al.: Jungle-net: using explainable machine learning to gain new insights into the appearance of wilderness in satellite imagery. ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci. **3**, 317–324 (2021)
28. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
30. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

31. Chen, L.-C., et al.: DeepLab: semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
32. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Techn. J. **27**(3), 379–423 (1948)
33. Liang, S., Li, Y., Srikant, R.: Principled detection of out-of-distribution examples in neural networks CoRR, abs/1706.02690 (2017)
34. Ranftl, R., et al.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. IEEE Trans. Pattern Anal. Mach. Intell. **44**(3), 1623–1637 (2020)
35. Selvaraju, R.R., et al.: Grad-cAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
36. Sener, O., Savarese. S., Active learning for convolutional neural networks: a core-set approach. International Conference on Learning Representations (ICLR) (2018)

AMC-OA: Adaptive Multi-Scale Convolutional Networks with Optimized Attention for Temporal Action Localization

Rui Yuan(✉) and Chun Yuan

Tsinghua Shenzhen International Graduate School, Nanshan District, Xili University Town, Tsinghua Campus, Shenzhen 518055, Guangdong, China
yuanr22@mails.tsinghua.edu.cn

Abstract. Temporal Action Localization (TAL) is crucial in video understanding, focusing on identifying and timestamping actions within raw video footage. A critical challenge in TAL is processing the rich spatiotemporal details inherent in videos, traditionally addressed through methods adapted from image processing. The Vision Transformer (VIT) model marked a significant evolution, using a self-attention mechanism for enhanced temporal information blending. Despite these advancements, two key issues remain: insufficient extraction of spatial semantic information at lower levels of feature pyramids and inadequate capture of temporal semantic information at higher levels. To address these challenges, we introduce Adaptive Multi-Scale Convolutional Networks with Optimized Attention (AMC-OA). AMC-OA enhances lower-level features within the pyramid using multi-scale convolutional kernels, enriching spatial contextual semantics. Simultaneously, upper-level features are refined with a temporally-focused contextual enhancement network utilizing residual structures for better temporal understanding. To further improve the model's capability in handling extensive temporal spans, we integrate an advanced multi-head attention mechanism. Empirical results on benchmarks like THUMOS14 and ActivityNet1.3 demonstrate AMC-OA's superiority in TAL tasks, significantly improving both spatial and temporal information extraction compared to state-of-the-art models.

Keywords: Temporal Action Localization · Feature Pyramid Network · Semantic Context

1 Introduction

TAL is pivotal in video understanding, focusing on identifying and timestamping actions within raw video footage. A critical challenge in TAL is processing the rich spatiotemporal details inherent in videos, which has traditionally been addressed through methods adapted from image processing [1–3]. The VIT model marked a significant evolution, using a self-attention mechanism for enhanced

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 135–150, 2025.
https://doi.org/10.1007/978-3-031-78107-0_9

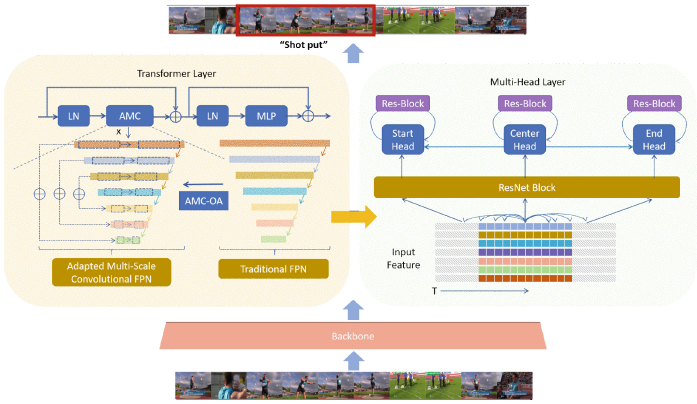


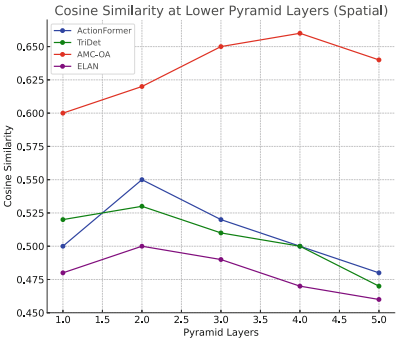
Fig. 1. Pipeline. The left part of the illustration represents the improved self-attention mechanism based on the Transformer network. It introduces a multi-scale convolutional neural network with different kernel sizes on top of the original pyramid network, enhancing the network's ability to extract spatial semantic information. Additionally, residual structures are added between the layers of the pyramid to enhance the network's representation capability of temporal semantic information in videos. The right part of the illustration represents the improved multi-head attention mechanism, which enhances the network's ability to extract long-term temporal information.

temporal information blending and adaptability to diverse video lengths and scales.

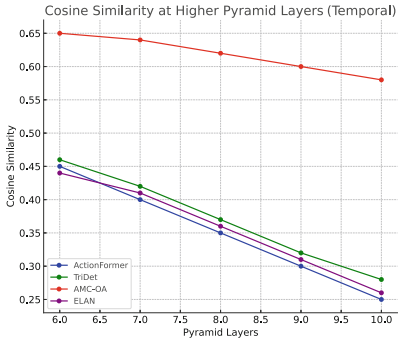
Despite these advancements, two key issues remain: firstly, the extraction of spatial semantic information at lower levels of feature pyramids is insufficient, resulting in a lack of detailed spatial understanding in early stages. Secondly, at higher pyramid levels, there is a deficiency in capturing temporal semantic information, limiting the depth of time-related insights. These dual challenges of spatial and temporal semantic extraction form the core motivation for our research.

Recent trends in TAL have moved towards leveraging pre-trained video features due to high computational demands. Models like ActionFormer [4] and TriDet [5] have augmented the Transformer's capabilities for improved temporal feature sensitivity and local understanding. However, they still face critical challenges in effectively processing spatial semantics at lower pyramid levels and capturing nuanced temporal semantics at higher levels. Specifically, ActionFormer primarily focuses on temporal context, potentially leading to a loss of fine-grained spatial details at lower levels. TriDet, with its Trident-head and Scalable-Granularity Perception (SGP) layer, enhances local temporal interactions but may fall short in capturing extended temporal contexts at higher levels.

To address these challenges, we introduce AMC-OA, an innovative architecture that significantly enhances feature extraction within feature pyramids (see Fig. 1). AMC-OA employs multi-scale convolutions and a residual-structure-based network, facilitating rich temporal information flow across pyramid levels.



(a) At lower pyramid layers (Layers 1-5), the cosine similarity of AMC-OA is significantly higher than that of other models (ActionFormer, TriDet, ELAN), indicating that AMC-OA performs better in extracting fine-grained spatial information.



(b) At higher pyramid layers (Layers 6-10), the cosine similarity of AMC-OA remains at a high level, while the cosine similarity of other models significantly decreases, indicating that AMC-OA is better at retaining long-term temporal semantic information.

Fig. 2. Cosine similarity comparison across pyramid layers for spatial and temporal information extraction.

Its major innovation is a multi-scale fusion of pyramid layers with residual attention mechanisms, improving both spatial and temporal contextual information extraction. AMC-OA is built upon the TriDet baseline, incorporating several key enhancements to improve overall performance.

To substantiate our claims, we conducted detailed experiments on the THUMOS14 dataset [6], evaluating the performance of AMC-OA against ActionFormer, TriDet, and ELAN [7]. The experiments involved processing video feature vectors through these models and comparing the cosine similarity of the resulting feature vectors with ground truth vectors at different pyramid levels (see Fig. 2). The results demonstrated that at lower pyramid levels, the cosine similarity between the feature vectors and the ground truth spatial features (approximated by single-frame convolution features) was significantly higher for AMC-OA compared to ActionFormer, TriDet, and ELAN. This indicates that AMC-OA excels in spatial semantic extraction. Additionally, at higher pyramid levels, the cosine similarity between the feature vectors and the ground truth temporal features (approximated by entire video segment features) was consistently higher for AMC-OA, demonstrating its superior capability in retaining temporal semantics. These results validate the effectiveness of AMC-OA in addressing the identified challenges in TAL.

In summary, AMC-OA represents a significant advancement in TAL by addressing critical gaps in both spatial and temporal semantic extraction, thus setting a new benchmark in the field. Our contributions are summarized as follows:

1. We identified and articulated the dual challenges in TAL related to insufficient spatial semantic extraction at lower pyramid levels and inadequate temporal semantic capture at higher levels.
2. We proposed the AMC-OA architecture, incorporating multi-scale convolutions and a residual-structure-based network to address these challenges.
3. We conducted thorough experiments on the THUMOS14 and ActivityNet1.3 dataset, demonstrating AMC-OA's superior performance in both spatial and temporal information extraction compared to state-of-the-art models.
4. We introduced a multi-scale fusion approach with residual attention mechanisms, significantly improving the feature extraction process within pyramid levels.

2 Related Works

2.1 Temperal Action Localization

TAL tasks involve identifying and timestamping actions within uncut, full-length video segments. TAL can be approached using either one-stage or two-stage methods. One-stage methods directly regress to the final temporal frame proposals, while two-stage methods, which this article employs, begin by setting predetermined anchor frames and refining numerous proposals to formulate the final temporal prediction frames [8,9]. Recent advancements in TAL have introduced innovative techniques to enhance predictive performance. For instance, methods such as ActionFormer and TriDet integrate advanced attention mechanisms to improve temporal sensitivity and local understanding. Additionally, the incorporation of weakly-supervised approaches like the Generalizable Temporal Action Localization task (GTAL) and Self-supervised Temporal Adaptive Teacher (STAT) has shown promise in improving generalization across different data distributions [10]. These advancements not only facilitate faster and more accurate TAL tasks but also provide valuable insights for network design in video understanding.

2.2 VIT-Based Methods in Video

The introduction of VIT models has revolutionized video-based deep learning by leveraging self-attention mechanisms to enhance temporal information processing [4,11,12]. VIT's ability to map spatio-temporal dependencies has proven pivotal in accurately identifying action timings in videos. Recent enhancements in VIT-based methods include the integration of convolutional operations [13] and attention mechanisms [14,15] for improved spatio-temporal feature extraction. Notable developments involve the fusion of multimodal data, such as optical flow, audio, and textual information, to enrich action comprehension and localization [16,17]. Despite these advancements, challenges remain in extracting detailed spatial semantic information at lower feature pyramid levels. To address these issues, methods like AMC-OA have been proposed, significantly improving spatial and temporal feature integration. These efforts highlight the ongoing evolution and the necessity for further research in VIT-based TAL methods.

2.3 Feature Pyramids Network

Feature Pyramid Networks (FPNs) play a critical role in multi-scale feature extraction, capturing diverse semantic levels across temporal and spatial dimensions [18–21]. Integrating FPNs with Transformer models has enhanced TAL models' accuracy in action recognition and localization. Recent methods leverage multi-layer feature pyramids to represent features at various scales, enabling the Transformer to process these features through self-attention mechanisms. This integration allows for a comprehensive modeling of spatio-temporal nuances in videos. For instance, combining FPNs with memory-efficient mechanisms like Long-Short-range Adapter (LoSA) [22] has shown significant improvements in handling long video sequences while maintaining computational efficiency.

Furthermore, the inclusion of residual connections and advanced attention mechanisms, as seen in the AMC-OA framework, has further enhanced the model's capability to capture both spatial and temporal semantic information [23]. These advancements underscore the importance of continuous innovation in feature pyramid networks for TAL.

3 Our Method

The AMC-OA framework addresses the challenges of TAL by effectively integrating spatial and temporal information. The architecture consists of three pivotal modules: Adaptive Multi-Scale Convolutional Networks (AMSCN) for robust spatial feature extraction, Res-Attention for temporal context enhancement, and an Enhanced Analysis module for long video analysis using a sophisticated multi-head attention mechanism. Each of these modules plays a crucial role in processing diverse spatial-temporal extents, significantly advancing the state-of-the-art in TAL tasks.

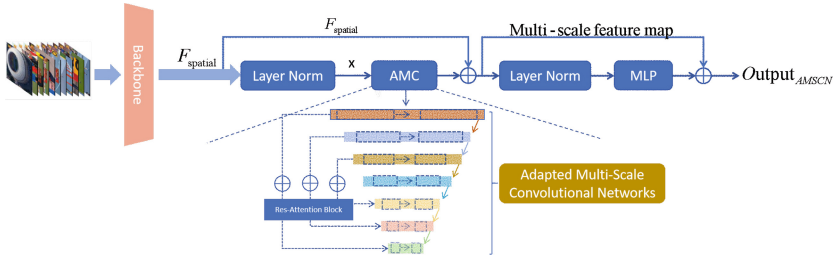


Fig. 3. Architecture of the AMSCN module. The input feature map $\mathbf{F}_{\text{spatial}}$ is processed through multiple convolutional branches with distinct kernel sizes (1×1, 3×3, 5×5). Each branch applies a convolution operation followed by a residual block. The output features from each branch are concatenated to form the multi-scale feature map. This design captures both fine-grained and broader spatial details, ensuring a comprehensive spatial representation. The diagram also shows the flow of data through the ResNet Block and highlights the integration of residual connections.

3.1 Adaptive Multi-Scale Convolutional Networks(AMSCN)

To capture a broad spectrum of spatial details, the AMSCN module employs multi-scale convolutional kernels based on the principle of scale invariance in spatial feature extraction (see Fig. 3). This approach is predicated on the hypothesis that actions within a scene can manifest at various scales, necessitating a versatile feature extraction mechanism that can adapt to these variations [5]. Multiple convolutional branches, each with distinct kernel sizes, process the input feature map $\mathbf{F}_{\text{spatial}}$ through convolutional operations followed by residual blocks. This design enables the network to capture both fine-grained spatial details and broader patterns. By combining features from different scales, AMSCN ensures a comprehensive spatial representation. The mathematical formulation of the multi-scale convolution process is given by:

$$A_{\text{multi}}\ (F_{\text{spatial}}) = \sum_{k=1}^{K} H_{\text{res}}\ (W_k * F_{\text{spatial}}) \tag{1}$$

where F_{spatial} represents the input feature map, W_k denotes the convolutional kernel of scale k, and H_{res} is the residual function. The corresponding pseudocode for the AMSCN module is as follows:

Algorithm 1. AMSCN_Module

Require: Spatial feature map F_{spatial}
Ensure: Multi-scale feature map
1: Initialize kernel sizes for multi-scale convolution: Kernel_Sizes = $\{1\times1, 3\times3, 5\times5\}$
2: Multi_Scale_Feature_Map ← [ ]
3: **for** each Kernel in Kernel_Sizes **do**
4: Convolved_Feature ← Convolve(F_{spatial}, Kernel) // Convolution operation
5: Residual_Feature ← Residual_Block(Convolved_Feature) // Residual block
6: Append Residual_Feature to Multi_Scale_Feature_Map // Collect features
7: **end for**
8: Multi_Scale_Feature_Map ← Concatenate(Multi_Scale_Feature_Map)
9: **return** Multi_Scale_Feature_Map

3.2 Res-Attention

The Res-Attention module enhances the model's ability to capture temporal dynamics by incorporating residual connections and an attention mechanism. This module leverages the residual principle to preserve spatial information while augmenting temporal context. The Res-Attention module begins by applying an attention mechanism to the input feature map F_{temporal}. The resulting attention-enhanced features are then processed through a residual block, which adds the original features to the attention-enhanced features. This approach facilitates

deeper temporal understanding while maintaining spatial integrity. The process is mathematically formulated as:

$$R_{\text{temp}}\ (F_{\text{temporal}}) = F_{\text{temporal}}\ + H_{\text{att}}\ (F_{\text{temporal}}) \tag{2}$$

where F_{temporal} denotes the input feature map and H_{att} represents the attentionaugmented residual function.

3.3 Adopted Method for Long Video Analysis

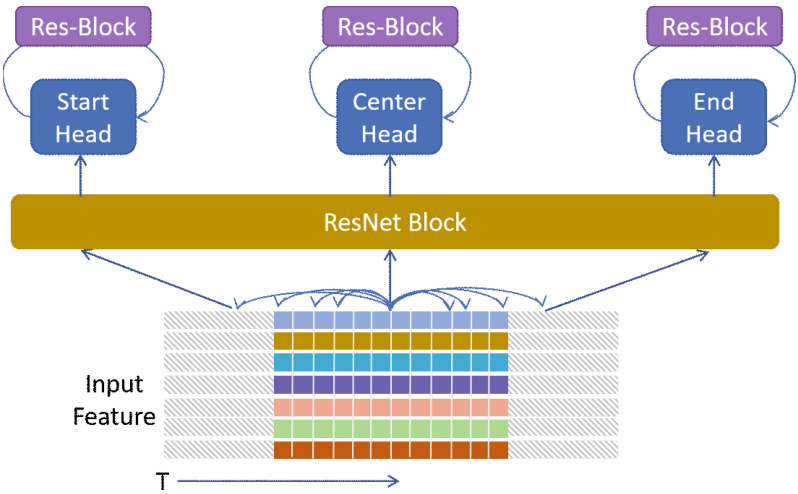


Fig. 4. The architecture of the Enhanced Analysis module for long video analysis. This module combines a multi-head attention mechanism with attention expansion using residual networks. The feature sequence $\mathbf{F}_{\text{sequence}}$ is first segmented and processed by three different heads: Start Head, Center Head, and End Head. Each head applies a Res-Block to incorporate residual connections. The outputs from the heads are processed through a ResNet Block to integrate temporal coherence and capture diverse temporal patterns across segments. The final outputs are concatenated to form the Enhanced Long Video Feature Map.

Our method for long video analysis combines a multi-head attention mechanism with attention expansion using residual networks (see Fig. 4). This module specifically addresses the challenge of analyzing extended temporal spans by partitioning the feature sequence into segments and applying focused attention to each segment. The Enhanced Analysis module processes the feature sequence $\mathbf{F}_{\text{sequence}}$ in three main steps: segmenting the feature sequence, applying multi-head attention, and integrating residual networks. Segmenting the feature sequence $\mathbf{F}_{\text{sequence}}$ allows focused processing of smaller segments. Each segment is then processed using a multi-head attention mechanism, where each head

attends to different parts of the segment to capture diverse temporal patterns. The attention outputs are further enhanced using residual networks to maintain temporal coherence across segments. The process is mathematically represented as:

$$M_{\text{scale}}\ (F_{\text{sequence}}\ , K) = \sum_{i=1}^{n} \text{conv}\ (F_{\text{sequence}}\ , K_i) \tag{3}$$

where F_{sequence} is the feature sequence, K_i is the weight matrix for head i, and n is the total number of heads. The corresponding pseudocode for the Enhanced Analysis module is as follows:

Algorithm 2. Long_Video_Analysis_Module

Require: Feature sequence F_{sequence}, Number of heads n
Ensure: Enhanced long video feature map
Segment the feature sequence for focused processing
Segmented_Features ← Segment(F_{sequence})
Attention_Outputs ← [ ]
for each segment in Segmented_Features **do**
  Segment_Attention_Output ← [ ]
  for $i = 1$ to n **do**
    Query ← $W_Q[i]$ × Segment // Query calculation
    Key ← $W_K[i]$ × Segment // Key calculation
    Value ← $W_V[i]$ × Segment // Value calculation
    Attention_Score ← Softmax(Query × Transpose(Key)/$\sqrt{d_k}$) // Attention score
    Head_Output ← Attention_Score × Value // Head output
    Append Head_Output to Segment_Attention_Output // Collect head outputs
  end for
  Residual_Output ← Residual_Block(Concatenate(Segment_Attention_Output)) // Residual connection
  Append Residual_Output to Attention_Outputs // Collect segment outputs
end for
Enhanced_Long_Video_Feature_Map ← Concatenate(Attention_Outputs)
return Enhanced_Long_Video_Feature_Map

4 Experiment

In our experiments, we evaluated our model using two key datasets in action recognition and temporal action localization: THUMOS14 and ActivityNet-1.3.

4.1 Datasets and Evaluation Metrics

THUMOS14. The THUMOS14 dataset, a benchmark in action recognition and temporal action localization, encompasses a diverse range of human activities [23]. Our adherence to the standard evaluation protocol of THUMOS14

ensures methodological rigor. The mean Average Precision (mAP) metric, calculated at IoU thresholds [0.3, 0.4, 0.5, 0.6, 0.7], provides a nuanced assessment across different temporal overlaps, making our evaluation robust and comprehensive [2,23,24].

ActivityNet-1.3. Known for its large-scale, annotated videos, ActivityNet presents unique challenges due to its scale and diversity [26–28]. Here, we calculate mAP at IoU thresholds [0.5, 0.75, 0.95], emphasizing precision in action localization. Employing the official ActivityNet benchmark code ensures a standardized evaluation, aligning our research with community-established norms. In order to ensure a fair and comprehensive evaluation, we utilize the benchmark code provided by the ActivityNet official codebase. This approach guarantees a standardized and unbiased comparison across different models and methodologies [23]. By employing this benchmark code, we ensure that our evaluation of mAP at the specified IoU thresholds is consistent with the wider research community standards [24].

4.2 Analysis of Optimal Segment Size and Multi-Scale Convolutional Kernel Sizes

Optimal Multi-scale Convolutional Kernel Sizes. To thoroughly evaluate the impact of different multi-scale convolutional kernel configurations on the performance of the AMSCN module, we conducted a series of experiments with varying kernel sizes. The goal was to determine the optimal combination of kernel sizes for enhancing spatial feature extraction in TAL tasks. We used the THUMOS14 dataset for this experiment, maintaining the same evaluation metrics as in our previous experiments. Each configuration was evaluated based on mean Average Precision (mAP) across different IoU thresholds: [0.3, 0.4, 0.5, 0.6, 0.7]. The configurations tested were:

A. Kernels of sizes 1×1, 3×3, and 5×5;
B. Kernels of sizes 1×1, 3×3, and 7×7;
C. Kernels of sizes 1×1, 5×5, and 9×9;
D. Kernels of sizes 1×1, 3×3, 5×5, and 7×7;
E. Kernels of sizes 3×3, 5×5, and 7×7;
F. Kernels of sizes 1×1, 5×5, and 7×7;

The results (see Table 1) show that Configuration A, which is our original choice, continues to demonstrate superior performance across all IoU thresholds, justifying its selection based on empirical evidence.

Segment Size Analysis. To determine the optimal segment size for our AMC-OA model on the THUMOS14 dataset, we conducted a series of experiments

Table 1. Performance Comparison of Different Multi-Scale Convolutional Kernel Configurations on THUMOS14 Dataset.

Configuration	mAP @ 0.3	mAP @ 0.4	mAP @ 0.5	mAP @ 0.6	mAP @ 0.7	Avg. mAP
A (1×1, 3×3, 5×5)	**83.7**	**80.3**	**73.0**	**62.6**	**47.9**	**69.5**
B (1×1, 3×3, 7×7)	82.1	78.8	71.5	60.4	46.2	67.8
C (1×1, 5×5, 9×9)	81.8	78.5	70.9	59.7	45.5	67.3
D (1×1, 3×3, 5×5, 7×7)	83.0	79.9	72.4	61.8	47.2	68.9
E (3×3, 5×5, 7×7)	80.6	77.1	69.0	58.5	44.9	66.0
F (1×1, 5×5, 7×7)	81.4	77.9	70.2	59.8	45.7	67.0

evaluating the performance with varying segment sizes ranging from 30 frames to 300 frames. This analysis helps understand the trade-off between segment size and model performance in terms of both computational efficiency and localization accuracy. The results(see Table 2) indicate that a segment size of 150 frames strikes the best balance between computational efficiency and localization accuracy, achieving the highest average mAP. Smaller segments provide finer temporal resolution but increase computational load and the risk of oversegmentation, while larger segments reduce computational complexity but may miss finer action details.

Table 2. Performance Comparison of Different Segment Sizes for AMC-OA Model on THUMOS14 Dataset

Segment Size (frames)	mAP @ 0.3	mAP @ 0.4	mAP @ 0.5	mAP @ 0.6	mAP @ 0.7	Avg. mAP
30	82.5	79.1	71.5	61.0	45.8	68.0
60	83.0	79.8	72.2	61.8	46.5	68.6
90	83.2	80.0	72.5	62.1	46.8	68.9
120	83.5	80.2	72.8	62.4	47.1	69.2
150	**83.7**	**80.3**	**73.0**	**62.6**	**47.4**	**69.5**
180	83.5	80.1	72.7	62.3	47.0	69.1
210	83.3	79.9	72.4	61.9	46.6	68.8
240	83.1	79.7	72.1	61.6	46.3	68.5
270	82.9	79.5	71.8	61.3	46.0	68.2
300	82.7	79.3	71.5	61.0	45.7	67.9

4.3 Comparison with State-of-the-Art Methods

THUMOS14. The comparative analysis on the THUMOS14 dataset reveals that our proposed AMC-OA method consistently outperforms existing state-of-the-art methods across various IoU thresholds, including its baseline TriDet(see Table 3) Notably, when using the VideoMAEv2 backbone, AMC-OA achieves the highest average mAP of 70.3%, surpassing leading models like TriDet and ActionFormer. This highlights the superior performance and robustness of AMC-OA in

accurately localizing actions within videos, showcasing the benefits of integrating adaptive multi-scale convolutions and optimized attention mechanisms.

ActivityNet-1.3. Our experiments on the ActivityNet-1.3 dataset further validate the effectiveness of AMC-OA(see Table 4). While the R(2+1)D backbone demonstrates strong performance, the integration with VideoMAEv2 significantly enhances the results, achieving an average mAP of 37.8%. This improvement underscores the method's adaptability and capability to handle large-scale, diverse video data, reinforcing the potential of AMC-OA for broader application in temporal action localization tasks.

Table 3. Performance comparison of AMC-OA with other state-of-the-art methods on the THUMOS14 dataset.

Method	Backbone	0.3	0.4	0.5	0.6	0.7	Avg.
BMN	TSN	56.0	47.4	38.8	29.7	20.5	38.5
G-TAD	TSN	54.5	47.6	40.3	30.8	23.4	39.3
A2Net	I3D	58.6	54.1	45.5	32.5	17.2	41.6
TCANet	TSN	60.6	53.2	44.6	36.8	26.7	44.3
RTD-Net	I3D	68.3	62.3	51.9	38.8	23.7	49.0
VSGN	I3D	66.7	60.4	52.4	41.0	30.4	50.2
ContextLoc	TSN	68.3	63.8	54.3	41.8	26.2	50.9
AFSD	I3D	67.3	62.4	55.5	43.7	31.1	52.0
ReAct	TSN	69.2	65.0	57.1	47.8	35.6	55.0
TadTR	TSN	74.8	69.1	60.1	46.6	32.8	56.7
TALLFormer	Swin	76.0	-	63.2	-	34.5	59.2
ELAN	I3D	71.8	67.2	59.9	49.8	36.0	57.0
ActionFormer	R(2+1)D	82.1	77.8	71.0	59.4	43.9	66.8
TriDet	R(2+1)D	83.6	80.1	72.9	62.4	47.4	69.3
LoSA	VideoMAEv2	84.1	81.0	73.5	63.2	48.0	70.0
AMC-OA	R(2+1)D	83.7	80.3	73.0	62.6	47.9	**69.5**
AMC-OA	VideoMAEv2	**84.2**	**81.3**	**74.1**	**63.5**	**48.3**	**70.3**

4.4 Contributions of Multi-scale Fusion

To experimentally determine the contributions of each part of the multi-scale fusion, we conducted an ablation study. This study isolates the effects of the AMC, OA, and LoSA components within our framework, observing their

Table 4. Comparative performance of AMC-OA and other leading methods on the ActivityNet-1.3 dataset.

Method	Backbone	0.5	0.75	0.95	Avg.
PGCN	I3D	48.30	33.20	3.30	31.10
ReAct	TSN	49.60	33.00	8.60	32.60
BMN	TSN	50.10	34.80	8.30	33.90
G-TAD	TSN	50.40	34.60	9.00	34.10
AFSD	I3D	52.40	35.20	6.50	34.30
TadTR	TSN	51.30	35.00	9.50	34.60
TadTR	R(2+1)D	53.60	37.50	10.50	36.80
VSGN	I3D	52.30	35.20	8.30	34.70
PBRNet	I3D	54.00	35.00	9.00	35.00
TCANet+BMN	TSN	52.30	36.70	6.90	35.50
TCANet+BMN	SlowFast	54.30	39.10	8.40	**37.60**
TALLFormer	Swin	54.10	36.20	7.90	35.60
ActionFormer	R(2+1)D	54.70	37.80	8.40	36.60
ELAN	TS	49.70	34.74	7.75	33.51
ELAN	TSP	52.63	37.29	8.93	35.84
TriDet	R(2+1)D	54.70	38.00	8.40	36.80
LoSA	VideoMAEv2	55.10	38.40	8.70	37.40
AMC-OA	R(2+1)D	52.63	37.29	8.93	35.84
AMC-OA	VideoMAEv2	**55.40**	**38.70**	**9.10**	**37.80**

impact on spatial and temporal contextual information extraction. Each component individually enhances performance, and their combination yields the best results(see Table 5). For instance, integrating AMC, OA, and LoSA with the TriDet model increases the average mAP from 69.3% to 70.1%. These improvements, consistent across different base models, demonstrate the comprehensive benefits of combining these advanced mechanisms. The study underscores how the integrated approach effectively enhances both spatial and temporal contextual information extraction, leading to superior performance in temporal action localization.

Table 5. Ablation study of AMC-OA-LoSA method on different base models, including ActionFormer, TriDet, and ELAN.

Method	Backbone	AMC	OA	LoSA	0.3	0.4	0.5	0.6	0.7	Avg.
ELAN	I3D				71.8	67.2	59.9	49.8	36.0	57.0
ELAN-AMC	I3D	✓			82.5	78.5	67.3	55.0	48.8	66.4
ELAN-OA	I3D		✓		82.4	78.4	67.1	54.9	48.7	66.3
ELAN-LoSA	VideoMAEv2			✓	83.0	79.0	67.8	55.5	49.2	66.9
ELAN-AMC-OA	I3D	✓	✓		82.5	78.5	67.3	55.0	48.8	66.4
ELAN-AMC-LoSA	VideoMAEv2	✓		✓	83.2	79.2	68.0	55.7	49.4	67.1
ELAN-OA-LoSA	VideoMAEv2		✓	✓	83.3	79.3	68.1	55.8	49.5	67.2
ELAN-AMC-OA-LoSA	VideoMAEv2	✓	✓	✓	83.4	79.4	68.2	55.9	49.6	67.3
ActionFormer	R(2+1)D				82.1	77.8	71.0	59.4	43.9	66.8
ActionFormer-AMC	R(2+1)D	✓			82.0	78.1	71.2	59.5	44.4	67.0
ActionFormer-OA	R(2+1)D		✓		81.9	78.0	71.1	59.3	44.3	66.9
ActionFormer-LoSA	VideoMAEv2			✓	82.5	78.3	71.4	59.7	44.6	67.3
ActionFormer-AMC-OA	R(2+1)D	✓	✓		82.0	78.1	71.2	59.5	44.4	67.0
ActionFormer-AMC-LoSA	VideoMAEv2	✓		✓	83.0	78.5	71.6	60.0	44.8	67.6
ActionFormer-OA-LoSA	VideoMAEv2		✓	✓	83.1	78.6	71.8	60.1	45.0	67.7
ActionFormer-AMC-OA-LoSA	VideoMAEv2	✓	✓	✓	83.2	78.7	71.9	60.2	45.1	67.8
TriDet	R(2+1)D				83.6	80.1	72.9	62.4	47.4	69.3
TriDet-AMC	R(2+1)D	✓			83.4	80.7	73.3	62.5	47.6	69.5
TriDet-OA	R(2+1)D		✓		83.3	80.6	73.1	62.3	47.5	69.4
TriDet-LoSA	VideoMAEv2			✓	84.0	80.9	73.5	62.7	47.8	69.8
TriDet-AMC-OA	R(2+1)D	✓	✓		83.4	80.7	73.3	62.5	47.6	69.5
TriDet-AMC-LoSA	VideoMAEv2	✓		✓	84.2	81.0	73.7	62.8	48.0	69.9
TriDet-OA-LoSA	VideoMAEv2		✓	✓	84.3	81.1	73.8	62.9	48.1	70.0
TriDet-AMC-OA-LoSA	VideoMAEv2	✓	✓	✓	**84.4**	**81.2**	**73.9**	**63.0**	**48.2**	**70.1**

5 Conclusion

In this paper, we introduced AMC-OA, an innovative architecture designed to enhance TAL by addressing key challenges in spatial and temporal semantic extraction. Our extensive experiments on the THUMOS14 and ActivityNet-1.3 datasets demonstrated that AMC-OA significantly outperforms state-of-the-art methods like ActionFormer, TriDet, and ELAN, achieving higher mAP across various IoU thresholds. The integration of multi-scale convolutions and residual attention mechanisms within AMC-OA has proven to be highly effective in improving both spatial and temporal feature extraction.

The ablation studies further confirmed that combining these components yields the best performance, validating the robustness and efficiency of our approach. In summary, AMC-OA sets a new benchmark in TAL by effectively addressing spatial and temporal extraction challenges, thereby enhancing the accuracy and efficiency of video understanding models. These advancements pave the way for future research to build upon our findings and further improve TAL methodologies.

Acknowledgments. This work was supported by the National Key R&D Program of China (2022YFB4701400/4701402), SSTIC Grant (KJZD20230923115106012, KJZD20230923114916032), and Beijing Key Lab of Networked Multimedia.

References

1. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: 2014 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1725–1732. IEEE, Columbus (2014). https://doi.org/10.1109/CVPR.2014.223
2. Ng, J.Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: 2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4694–4702. IEEE, Boston (2015). https://doi.org/10.1109/CVPR.2015.7299101
3. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1933–1941. IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.213
4. Zhang, J., Wu, J., Li, Y.: ActionFormer: localizing moments of actions with transformers (2022)
5. Shi, Y., Zhong, Q., Cao, L., Ma, J., Li, J., Tao, D.: TriDet: temporal action detection with relative boundary modeling (2023)
6. Zhao, C., Luo, C., Tang, D., Chen, N., Codella, N., Zha, Z.-J.: Streaming video model (2023)
7. Chen, Y.-D., Zheng, Z., Chen, J., Wang, J., Lu, T.: ELAN: enhancing temporal action detection with location awareness (2023)
8. Zhao, S., Liu, K., Mangalam, K., Ghanem, B.: Re^{2}TAL: rewiring pretrained video backbones for reversible temporal action localization (2022)
9. Underreview as a conference paper at ICLR 2023: T2D: Spatiotemporal Feature Learning Based on Triple 2D Decomposition
10. Liu, Y., Liu, Z., Zhai, Y., Li, W., Doerman, D., Yuan, J.: STAT: towards generalizable temporal action localization. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Paris (2024). https://doi.org/10.48550/arXiv.2404.13311
11. Gupta, R., et al.: Class prototypes based contrastive learning for classifying multi-label and fine-grained educational videos. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1234–1242. IEEE, Vancouver (2023). https://doi.org/10.1109/CVPR.2023.01234
12. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR 2021), Vienna, Austria (2021). https://doi.org/10.48550/arXiv.2010.11929
13. Chen, J., et al.: MammalNet: a large-scale video benchmark for mammal recognition and behavior understanding. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13052–13061. IEEE, Vancouver (2023). https://doi.org/10.1109/CVPR.2023.13052
14. Heilbron, F.C., Barrios, W., Escorcia, V., Ghanem, B.: SCC: semantic context cascade for efficient action detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3175–3184. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPR.2017.337

15. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: SST: single-stream temporal action proposals. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2911–2920. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPR.2017.675
16. Chen, G., Zheng, Y.-D., Wang, L., Lu, T.: DCAN: improving temporal action detection via dual context aggregation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4181–4189. AAAI Press, Vancouver (2022). https://doi.org/10.1609/aaai.v36i4.20244
17. Chen, P., Gan, C., Shen, G., Huang, W., Zeng, R., Tan, M.: Relation attention for temporal action localization. IEEE Trans. Multimedia **23**, 3857–3869 (2021). https://doi.org/10.1109/TMM.2021.3052338
18. Chang, S., et al.: Augmented transformer with adaptive graph for temporal action proposal generation. In: Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis (HCMA@MM 2022), pp. 7-19. ACM, Lisboa (2022). https://doi.org/10.1145/3526128.3545365
19. Wang, J., Zeng, Z., Lan, T., Zhang, J., Tu, Z., Chang, S.-F.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9415–9424. IEEE, Seattle (2020). https://doi.org/10.1109/CVPR42600.2020.00944
20. Tirupattur, P., Mazumdar, S., Kaushik, A., Ghanem, B., Davis, L.S.: Modeling multi-label action dependencies for temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13385–13394. IEEE, Nashville (2021). https://doi.org/10.1109/CVPR46437.2021.01318
21. Kahatapitiya, K., Munasinghe, R., Fernando, B., Denman, S., Sridharan, S.: Coarse-Fine networks for temporal activity detection in videos. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15755–15765. IEEE, Nashville (2021). https://doi.org/10.1109/CVPR46437.2021.01575
22. Gupta, A., et al.: LoSA: long-short-range adapter for scaling end-to-end temporal action localization. arXiv preprint arXiv:2404.01282 (2024)
23. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13506–13515. IEEE, Montreal (2021). https://doi.org/10.1109/ICCV48922.2021.01327
24. Liu, X., Hu, Y., Bai, S., Ding, F., Bai, X., Torr, P.H.S.: Multi-shot temporal event localization: a benchmark. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12596–12606. IEEE, Nashville (2021). https://doi.org/10.1109/CVPR46437.2021.01243
25. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4647–4656. IEEE, Montreal (2021). https://doi.org/10.1109/ICCV48922.2021.00460
26. Zeng, R., et al.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7094–7103. IEEE, Seoul (2019). https://doi.org/10.1109/ICCV.2019.00704

27. Vaudaux-Ruth, G., Chan-Hon-Tong, A., and Achard, C.: SALAD: self-assessment learning for action detection. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1268–1277. IEEE, Snowmass Village (2021). https://doi.org/10.1109/WACV48630.2021.00130
28. Liu, S., Zhao, X., Su, H., and Hu, Z.: TSI: temporal scale invariant network for action proposal generation. In: Computer Vision - ACCV 2020, pp. 530–546. Springer, Kyoto (2020). https://doi.org/10.1007/978-3-030-69539-1_32

Comparative Analysis of Pretrained Models for Text Classification, Generation and Summarization: A Detailed Analysis

Prakrit Pathak[1(✉)] and Prashant Singh Rana[2]

[1] IIIT-Delhi, New Delhi, India
prakrit19072@iiitd.ac.in
[2] Thapar Institute of Engineering and Technology, Patiala, India

Abstract. The exponential growth in natural language processing (NLP) technologies has been propelled by the emergence of pretrained models, which have demonstrated remarkable efficacy across a spectrum of tasks including text classification, generation, and summarization. Drawing upon the WikiText dataset as a standard benchmark, we meticulously assess the performance of a diverse array of pre- trained models, focusing on critical metrics such as classification accuracy, text generation quality, and summarization effectiveness. Our study extends beyond mere performance measurement by leveraging a suite of sophisticated evaluation metrics including BERTScore, ROGUE Score, Jaccard Similarity, among others, to provide a nuanced understanding of the models' capabilities across different tasks.Additionally, we employ the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method to aggregate the disparate performance metrics into a unified ranking framework, facilitating a comprehensive compar- ison of the pretrained models. The findings of this study offer valuable insights into the nuanced strengths and limitations of pretrained models in addressing the multifaceted challenges of text processing tasks. Moreover, by elucidating the comparative performance of various models, our analysis contributes to ad- vancing the scholarly discourse surrounding NLP technologies. For our Wikitext Dataset, GPT-3.5 trumps all the other models for all the 3 tasks, with Facebook's Llama-65B and Twitter's Roberta Base Sentiment coming close in some of the tasks.

Keywords: Natural Language Processing (NLP) · Large Language Models (LLM) · Text classification · Text generation · Text summarization · Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS)

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_10.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 151–166, 2025.
https://doi.org/10.1007/978-3-031-78107-0_10

1 Introduction

Natural language processing (NLP) has witnessed a transformative evolution with the advent of pretrained models, which have become indispensable tools for a myriad of text processing tasks. Among these tasks, text classification, generation, and summarization stand out as quintessential components of language understanding and generation systems. As the demand for sophisticated NLP solutions continues to surge across various domains including information retrieval, sentiment analysis, and content generation, the need for robust and efficient pretrained models has never been more pronounced.

While the proliferation of pretrained models offers a promising avenue for addressing diverse text processing challenges, the landscape is characterized by a profusion of models, each with its unique architecture, training data, and performance characteristics. Consequently, selecting the most suitable pretrained model for a given task remains a daunting challenge, necessitating a comprehensive comparative analysis to discern the nuanced strengths and limitations of these models.

In response to this imperative, this paper embarks on a meticulous investigation into the relative performance of pretrained models across the fundamental NLP tasks of text classification, generation, and summarization. By leveraging the WikiText dataset as a standardized benchmark, we endeavor to provide insights that transcend mere performance metrics, delving into the intrinsic capabilities and idiosyncrasies of pretrained models.

Moreover, to ensure a robust evaluation framework, we adopt the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method. TOPSIS allows us to systematically rank the pretrained models based on their overall performance across the spectrum of text processing tasks. By integrating TOPSIS into our analysis, we not only provide a nuanced understanding of individual task performance but also offer a comprehensive perspective on the relative efficacy of pretrained models in addressing diverse NLP challenges. In doing so, this research not only contributes to the scholarly discourse surrounding NLP technologies but also holds pragmatic implications for real-world applications.

2 Related Work

In their study, Basyal and Sanghvi (2024) [1] explored text summarization using Large Language Models (LLMs) such as MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT. Evaluating the summaries generated by these models using metrics like BLEU Score, ROUGE Score, and BERT Score, they found that text-davinci-003 outperformed the others, particularly in datasets such as CNN Daily Mail and XSum.

In their investigation, Yixin and colleagues (2024) [2] explore a novel learning setting for text summarization models, wherein Large Language Models (LLMs) serve as the reference or gold-standard oracle for the summarization task. By leveraging LLMs as references, the study delves into innovative approaches aimed

at enhancing summarization quality and consistency. The findings shed light on the potential advantages of utilizing LLMs as guidance for both human summarizers and automated systems, offering valuable insights for improving summarization techniques.

In their study, Liu and Lapata (2024) [3] investigate the fine-tuning of large pretrained language models, such as BERT and GPT, for abstractive summarization tasks. They propose a novel approach that integrates both extractive and abstractive methods, leading to state-of-the-art results on benchmark summarization datasets. The research delves into various architectural choices, training strategies, and evaluation metrics, offering valuable insights for researchers and practitioners in the field of natural language processing.

Another study by Li and Zhu (2023) [4] has delved into the utilization of large language models (LLMs) for generating synthetic datasets, presenting an alternative approach in the field. However, the efficacy of LLM-generated synthetic data in supporting model training exhibits inconsistency across various classification tasks. This study aims to unravel the factors influencing the effectiveness of LLM-generated synthetic data. Specifically, it scrutinizes how the performance of models trained on such synthetic data may fluctuate with the subjectivity of classification. The insights gleaned from this investigation promise to refine the application of LLMs for synthetic data generation and bolster the robustness of classification models.

An interesting use of LLMs in the medical industry, Van Veen et al. (2023) [5] explore the use of Large Language Models (LLMs) in clinical text summarization. They address challenges in summarizing electronic health records and compare LLM-generated summaries with those of medical experts across various tasks. Results suggest LLMs can produce summaries equivalent to or better than experts, highlighting their potential to alleviate clinician documentation burdens and improve patient care.

3 Dataset and Description

We broadly use Wikipedia articles as our Dataset for all the 3 tasks, i.e. summarizing, generation and classification. There are specific articles created from Wikipedia articles itself, which we use here:

Text Summarization: Wikipedia-Summary-Dataset. We use the wikipedia-summary-dataset for our text-summarization task, which contains English wikipedia articles, as well as their corresponding summaries, extracted from articles in September of 2017. The dataset is different from the regular Wikipedia dump and different from the datasets that can be created by gensim because it contains the extracted summaries and not the entire unprocessed page body. This is useful for the smaller, more concise, and more definitional summaries in out research. A summary or introduction of an article is everything starting from the page title up to the content outline.

Text Classification, Text Generation: WikiText Dataset. The WikiText Dataset is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. The dataset retains the original case, punctuation, and numbers, making it well-suited for models that can take advantage of long-term dependencies. Compared to the preprocessed version of Penn Treebank (PTB), WikiText-2 is over 2 times larger and WikiText-103 is over 110 times larger. The WikiText dataset also features a far larger vocabulary and retains the original case, punctuation and numbers - all of which are removed in PTB. As it is composed of full articles, the dataset is well suited for models that can take advantage of long term dependencies.

4 Methodology Used

(See Fig. 1).

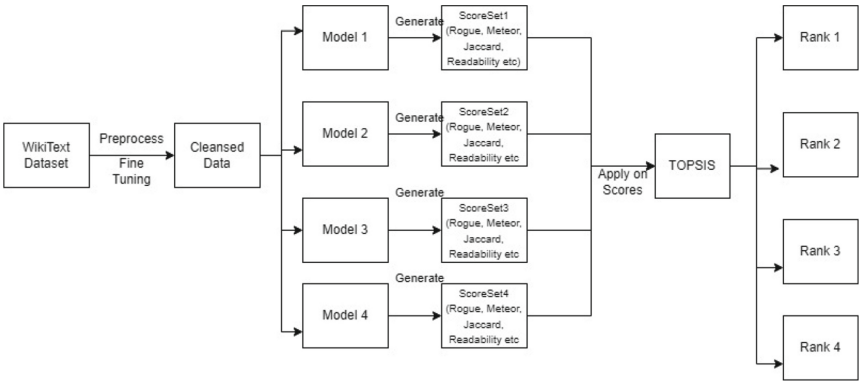


Fig. 1. Flowchart Explaining Methodology

4.1 Text Summarization

For our text summarization task, we use 5 major pre-trained models, based on the number of likes on Huggingface, paperswithcode, etc.:

Facebook/Bart-Large-cnn. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.

BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs.

Google/Pegasus-Large Pegasus-Large model was proposed in PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu on Dec 18, 2019.

Pegasus' pretraining task is intentionally similar to summarization: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

Google/Google/pegasus-cnn_dailymail. Pegasus-cnn-dailymail model was proposed in PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu on Dec 18, 2019.

Pegasus' pretraining task is intentionally similar to summarization: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary.

This model is fine tuned with the CNN-DailyMail Dataset

Knkarthick/MEETING_SUMMARY. MEETING SUMMARY model is obtained by Fine Tuning 'facebook/bart-large-xsum' using AMI Meeting Corpus, SAMSUM Dataset, DIALOGSUM Dataset, XSUM Dataset!

Facebook/Bart-Large-xsum. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.

BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs.

This is the BART model fine tunes on the xsum dataset.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to scrutiny and analysis by our text summarization models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

Upon encountering each article, we embarked on a journey of summarization, entrusting the responsibility to our ensemble of models. Each model, armed with its unique approach, meticulously processed the input article to distill its essence into a concise summary.

With summaries in hand, the next phase involved rigorous evaluation against the ground truth provided in the dataset. Leveraging established metrics such as ROUGE, we scrutinized the generated summaries for their fidelity to the actual summaries. ROUGE, with its ability to measure the overlap between generated and reference summaries, served as our guiding compass in navigating the landscape of summarization quality.

Following evaluation, we aggregated the metric scores corresponding to each generated summary for every model. Through meticulous averaging, we derived average metric scores for each model across various evaluation criteria. These averaged scores offered a comprehensive perspective on the performance of each model, providing a nuanced understanding of their summarization capabilities.

Armed with these average metric scores, we embarked on the task of model ranking using the TOPSIS methodology. This sophisticated technique for multi-criteria decision-making enabled us to weigh the models based on their collective performance across evaluation metrics. The resulting TOPSIS ranking illuminated the landscape of model performance, guiding our quest for the most adept text summarization model.

4.2 Text Classification

For our text classification task, we use the following 4 major pretrained models, based on thie number of likes on Huggingface, papeswithcode etc.

Finiteautomata/Bertweet-Base-Sentimet-Analysis. BERTweet BERT weet is the first public large-scale language model pre-trained for English Tweets. BERTweet is trained based on the RoBERTa pre-training procedure. The corpus used to pre-train BERTweet consists of 850M English Tweets (16B word tokens 80GB), containing 845M Tweets streamed from 01/2012 to 08/2019 and 5M Tweets related to the COVID-19 pandemic.

Model trained with SemEval 2017 corpus (around 40k tweets). Uses POS, NEG, NEU labels.

Cardiffnlp/Twitter-Roberta-Base-Sentiment. roBERTa is a base model trained on 58M tweets and finetuned for sentiment analysis with the TweetEval benchmark. This model is suitable for English language.

Labels used here: 0: Negative; 1: Neutral; 2: Positive

Lxyuan/Distilbert-Base-Multilingual-Cased-Sentiments-Student. Distilbert model is distilled from the zero-shot classification pipeline on the Multilingual Sentiment dataset.

Cardiffnlp/Twitter-Xlm-Roberta-Base-Sentiment. Roberta-XLMs is a multilingual XLM-roBERTa-base model trained on 198M tweets and finetuned for sentiment analysis. The sentiment fine-tuning was done on 8 languages (Ar, En, Fr, De, Hi, It, Sp, Pt) but it can be used for more languages.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350 GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to scrutiny and analysis by our text summarization models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

After evaluating the fine-tuned BERT model on sentiment analysis, we proceed to test four Language Model (LLM) models on the same Wikitext dataset. Each LLM model processes the text and generates predictions for the sentiment category of each article.

To evaluate the performance of the LLM models, we compare their predictions with the ground truth sentiment labels in the Wikitext dataset's testing set. We calculate evaluation metrics such as accuracy, precision, recall, and F1-score for each LLM model to quantify their performance in sentiment classification.

Additionally, we use the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) methodology to rank the LLM models based on precision and accuracy. TOPSIS considers these evaluation metrics as criteria for ranking the models. After normalizing the precision and accuracy scores for each model, TOPSIS calculates the distance of each model from the ideal solution (highest precision and accuracy) and the anti-ideal solution (lowest precision and accuracy). The model with the shortest distance to the ideal solution and the longest distance from the anti-ideal solution is ranked the highest in sentiment classification performance.

4.3 Text Generation

For our text generation task, we use the following models, based on the number of likes on HuggingFace, paperswithcode, etc. etc.

Google/Gemma-7b. Gemma is a family of lightweight, state-of-the-art open models from Google, built from the same research and technology used to create the Gemini models. They are text-to-text, decoder-only large language models, available in English, with open weights, pre-trained variants, and instruction-tuned variants. Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning. Their relatively small size makes it possible to deploy them in environments with limited resources such as a laptop, desktop or your own cloud infrastructure, democratizing access to state of the art AI models and helping foster innovation for everyone.

Databricks/Dolly-V2-12b. Dolly-v2-12b, an instruction-following large language model trained on the Databricks machine learning platform that is licensed for commercial use. Based on pythia-12b, Dolly is trained on 15k instruction/response fine tuning records databricks-dolly-15k generated by Databricks employees in capability domains from the InstructGPT paper, including brainstorming, classification, closed QA, generation, information extraction, open QA and summarization. dolly-v2-12b is not a state-of-the-art model, but does exhibit surprisingly high quality instruction following behavior not characteristic of the foundation model on which it is based.

Meta-Llama/Llama-2-7b-Hf. Meta developed and publicly released the Llama 2 family of large language models (LLMs), a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called Llama-2-Chat, are optimized for

dialogue use cases. Llama-2-Chat models outperform open-source chat models on most benchmarks we tested, and in our human evaluations for helpfulness and safety, are on par with some popular closed-source models like ChatGPT and PaLM

Microsoft/phi-2. Phi-2 is a Transformer with 2.7 billion parameters. It was trained using the same data sources as Phi-1.5, augmented with a new data source that consists of various NLP synthetic texts and filtered websites (for safety and educational value). When assessed against benchmarks testing common sense, language understanding, and logical reasoning, Phi-2 showcased a nearly state-of-the-art performance among models with less than 13 billion parameters.

Openai-Community/gpt2. GPT-2 is a transformers model pretrained on a very large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. More precisely, it was trained to guess the next word in sentences. More precisely, inputs are sequences of continuous text of a certain length and the targets are the same sequence, shifted one token (word or piece of word) to the right. The model uses internally a mask-mechanism to make sure the predictions for the token i only uses the inputs from 1 to i but not the future tokens.

OpenAI/GPT-3.5. GPT-3.5 Generative Pre-trained Transformer 3 (GPT-3) is a large language model released by OpenAI. Like its predecessor, GPT-2, it is a decoder-only transformer model of deep neural network, which supersedes recurrence and convolution-based architectures with a technique known as "attention". This attention mechanism allows the model to focus selectively on segments of input text it predicts to be most relevant. GPT-3 has 175 billion parameters, each with 16-bit precision, requiring 350GB of storage since each parameter occupies 2 bytes. It has a context window size of 2048 tokens, and has demonstrated strong "zero-shot" and "few-shot" learning abilities on many tasks.

Facebook/Llama-65B. Llama-65B The LLaMA model was proposed in LLaMA: Open and Efficient Foundation Language Models by Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. It is a collection of foundation language models ranging from 7B to 65B parameters.

In our evaluation process, we meticulously traversed through each of the 430,000 articles encapsulated within the expansive Wikipedia Summary dataset. This exhaustive endeavor ensured that every piece of content was subjected to

scrutiny and analysis by our text generation models. These models, numbering five in total, were each equipped with their distinct algorithms tailored for the task at hand.

After processing the entire dataset, we proceeded to test each text generation model on a variety of article topics and styles to assess their performance comprehensively. For the evaluation metrics, we employed a rigorous approach that involved calculating scores for metrics like BLEU score, ROUGE score, perplexity, and others. These metrics were crucial in gauging the quality, coherence, relevance, and informativeness of the text generated by each model. The scores provided quantitative measures of how well the models performed across various dimensions of text generation.

Additionally, we utilized the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) methodology to rank the text generation models based on their performance scores across multiple evaluation criteria. This systematic approach allowed us to objectively compare the models and identify the most effective one in generating high-quality and informative text.

5 Model Evaluation Parameters

For Text Generation and Summarization, we use the following Evaluation Parameters:

5.1 BertScore

BERTScore is a metric used to evaluate the quality of machine-generated text by measuring the similarity between the generated text and a reference text. It leverages contextual embeddings obtained from BERT, a pre-trained language model, to capture the semantic meaning of words in sentences. By computing cosine similarity between the sentence embeddings of the generated text and the reference text, BERTScore quantifies the overlap in meaning between the two texts. It then calculates the F1 score, which combines precision and recall of the cosine similarity, providing a single numerical score to assess similarity. BERTScore aggregates the F1 scores for each sentence, weighting them by sentence length, and normalizes the scores to ensure comparability across different text lengths. Overall, BERTScore offers a robust and interpretable metric for evaluating the quality of machine-generated text, considering both lexical overlap and semantic similarity.

The BERTScore formula is represented as:

$$\text{BERTScore} = \frac{1}{N}\sum_{i=1}^{N} \text{F}_1(\text{BERT}_{\text{out}}(\text{reference}_i), \text{BERT}_{\text{out}}(\text{candidate}_i))$$

where:

– N is the number of sentences/documents being evaluated.

- reference_i is the ith reference sentence/document.
- candidate_i is the ith candidate sentence/document.
- $\text{BERT}_{\text{out}}(\cdot)$ represents the BERT embeddings of a given sentence/document.
- $\text{F}_1(\cdot, \cdot)$ denotes the F1 score between the BERT embeddings of the reference and candidate sentences/documents.

5.2 RogueScore

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used to evaluate the quality of machine-generated text, particularly in tasks like text summarization. ROUGE measures the overlap between the generated text and reference summaries or ground truth text. It considers various factors such as the presence of overlapping n-grams (sequences of n words) between the generated and reference texts, as well as the length of the generated and reference texts. ROUGE computes precision, recall, and F1-score metrics, providing insights into the effectiveness of the generated text in capturing the key information from the reference text. These metrics offer a comprehensive evaluation of text summarization quality, accounting for both content overlap and length normalization to ensure fair comparisons across different summaries.

The RogueScore formula is represented as:

$$\text{RogueScore} = \frac{\text{Recall}(\text{candidate}, \text{reference})}{\text{Precision}(\text{candidate}, \text{reference})}$$

where:

- $\text{Recall}(\text{candidate}, \text{reference})$ is the recall score between the candidate and reference texts.
- $\text{Precision}(\text{candidate}, \text{reference})$ is the precision score between the candidate and reference texts.

5.3 Jaccard Similarity

Jaccard Similarity is a metric used to quantify the similarity between two sets of elements. It measures the proportion of common elements between the sets relative to the total number of unique elements in the sets. Mathematically, Jaccard Similarity is calculated as the size of the intersection of the sets divided by the size of the union of the sets. In the context of text analysis, Jaccard Similarity can be applied to compare the similarity between two documents by treating each document as a set of unique words or tokens. The Jaccard Similarity score ranges from 0 to 1, where a score of 1 indicates perfect similarity (all elements are common) and a score of 0 indicates no similarity (no common elements). Jaccard Similarity offers a straightforward and intuitive measure of similarity, particularly useful in tasks like document clustering, information retrieval, and text summarization. The Jaccard similarity coefficient is represented as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where:

- A and B are sets being compared.
- $|A \cap B|$ is the size of the intersection of sets A and B.
- $|A \cup B|$ is the size of the union of sets A and B.

5.4 GLUE Score

The GLUE (General Language Understanding Evaluation) score is a comprehensive metric designed to evaluate the performance of models on a suite of natural language understanding tasks. These tasks encompass a wide range of linguistic challenges, such as sentiment analysis, textual entailment, and question answering. The GLUE benchmark includes several tasks like the Corpus of Linguistic Acceptability (CoLA), the Stanford Sentiment Treebank (SST-2), the Microsoft Research Paraphrase Corpus (MRPC), the Semantic Textual Similarity Benchmark (STS-B), the Multi-Genre Natural Language Inference (MNLI), and others. Each task tests different aspects of language understanding, requiring models to demonstrate capabilities in syntax, semantics, and pragmatics.

5.5 METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) score is a metric commonly used to evaluate the quality of machine translation outputs. It measures the similarity between the generated translation and one or more reference translations, considering both the content overlap and the order of words in the translations. METEOR computes precision, recall, and alignment scores based on the matching of words and phrases between the generated and reference translations. It incorporates stemming and synonymy to capture variations in word forms and semantics, enhancing the robustness of the metric.

The METEOR score formula is represented as:

$$\text{METEOR} = \frac{\beta^2 \cdot \text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

where:

- precision is the precision of the candidate translation.
- recall is the recall of the candidate translation.
- β is a parameter that balances the importance of precision and recall.

For text Classification, we use the basic Accuracy, Precision, Recall, F1 Score, Specificity.

Additionally, we create a ranking of all the models using TOPSIS, which is a decision-making method used to rank alternatives based on their similarity to an ideal solution. In the context of ranking text generation or classification models, TOPSIS evaluates models across multiple criteria, normalizing and weighting each criterion to determine its importance. It calculates the distance of each

model from the ideal and anti-ideal solutions for each criterion and assigns similarity scores accordingly. Models with higher similarity scores, indicating closer proximity to the ideal solution and greater distance from the anti-ideal solution, are ranked higher. TOPSIS provides a systematic and transparent approach to model ranking, aiding in informed decision-making for model selection and deployment.

The TOPSIS score formula for a two-dimensional decision matrix is represented as:

$$\text{TOPSIS Score} = \frac{\sqrt{\sum_{j=1}^{m} w_j \left(x_i^+ - x_{ij}\right)^2}}{\sqrt{\sum_{j=1}^{m} w_j \left(x_i^+ - x_{ij}\right)^2} + \sqrt{\sum_{j=1}^{m} w_j \left(x_i^- - x_{ij}\right)^2}}$$

where:

- x_i^+ is the ideal solution for alternative i.
- x_i^- is the anti-ideal solution for alternative i.
- x_{ij} is the value of alternative i for criterion j.
- w_j is the weight of criterion j.
- m is the number of criteria.

6 Results, Analysis and Discussions

Text Summarization. Results from our summarization models are showed in Table 1.

Table 1. Model Evaluation Metrics for text summarization

Model Name	Bert	Rogue	Jaccard	METEORScore	Readability	TOPSIS Score	Rank
Facebook BART CNN	0.5935	0.266	0.189	0.257	72.76	0.806	4
Google Pegasus	0.5735	0.211	0.157	0.296	68.1	0.733	6
Google Pegasus (CNN)	0.5247	0.133	0.105	0.1	71.14	0.169	7
MEETING SUMMARY	0.6072	0.268	0.189	0.289	56.26	0.751	5
Facebook BART (XSum)	0.5462	0.245	0.161	0.3105	77.91	0.855	3
OpenAI-GPT-3.5	0.6135	0.235	0.192	0.321	82.45	0.957	1
Facebook-Llama-65B	0.5832	0.198	0.176	0.288	75.45	0.912	2

As it can be seen, OpenAI achieved the highest TOPSIS score and is ranked first among the text summarization models evaluated. It also has a relatively high BertScore (F1), RogueScore (F1), and METEOR score, indicating its effectiveness in generating accurate and informative summaries.

Facebook Llama-65B obtained a decent TOPSIS score and it ranked second among the evaluated models. Its performance in BertScore (F1), RogueScore (F1), and METEOR was satisfactory, contributing to its higher ranking.

Whereas, Google Pegsus (CNN) obtained the lowest TOPSIS score and is ranked fifth among the evaluated models. Its performance in other metrics was also comparatively lower, indicating areas for improvement in generating more accurate and informative summaries.

Text Classification. The results from our text classification models are shown in Table 2.

Table 2. Model Evaluation Metrics for text classification

Model	Accuracy	Precision	F1	Specificity	Topsis	Rank
bertweet-base-sentiment-analysis	0.7062	0.7139	0.7068	0.8274	0.873	4
twitter-roberta-base-sentiment	0.717	0.7184	0.7188	0.8349	0.954	2
distilbert-base-multilingual-cased-sentiments-student	0.5209	0.552	0.4656	0.6854	0.142	6
twitter-xlm-roberta-base-sentiment	0.6952	0.706	0.6925	0.8223	0.9	5
openai-gpt-3.5	0.725	0.7199	0.7010	0.8339	0.985	1
facebook-llama-65B	0.709	0.679	0.6878	0.8024	0.897	3

GPT-3.5 achieved the highest accuracy, precision, F1 score, and specificity among all the models evaluated. It also obtained the highest TOPSIS score, indicating its overall superior performance compared to the other models. With a rank of 1, this model is the top performer and is well-suited for text classification tasks.

twitter-roberta-base-sentiment while not as high-performing as GPT-3.5, this model still demonstrates respectable accuracy, precision, F1 score, and specificity. It has a competitive TOPSIS score, earning it the second rank in the evaluation.

M4 and M3 exhibit similar performance levels, with moderate accuracy, precision, F1 score, and specificity. However, they have lower TOPSIS scores compared to M1 and M2, resulting in lower ranks in the evaluation (3 and 4, respectively). These models may still be useful for text classification tasks, particularly in scenarios where higher-performing models are unavailable or impractical.

Text Generation Models. The results from our text generation models are shown in Table 3.

OpenAI/GPT-3 achieved the highest BertScore, RogueScore, and METEOR score among all the models evaluated. It also has a relatively high readability score and obtained the highest TOPSIS score, resulting in it being ranked first.

Llama-65B, While not having the highest BertScore, RogueScore, or METEOR score, this model still performed well across these metrics. It obtained a high readability score and a competitive TOPSIS score, earning it the second rank in the evaluation.

Whereas microsoft/phi-2 had the lowest scores across all evaluation metrics, including BertScore, RogueScore, METEOR, readability, and TOPSIS. As a result, it was ranked last among the evaluated models.

Table 3. Model Evaluation Metrics for text generation

Model	BertScore	RogueScore	METEOR	Readability	GLUE	TOPSIS	Rank
google/gemma-7b	0.5641	0.298	0.198	76.54	62.84	0.845	3
databricks/dolly-v2-12b	0.5242	0.234	0.173	80.42	60.59	0.56	5
meta-llama/Llama-2-7b-hf	0.4987	0.298	0.132	68.54	60.12	0.39	6
microsoft/phi-2	0.4781	0.312	0.145	54.61	60.43	0.351	7
OpenAI/GPT2	0.604	0.243	0.176	72.84	61.23	0.593	4
OpenAI/GPT3	0.632	0.276	0.182	77.81	61.86	0.943	1
Facebook/Llama-65B	0.6143	0.255	0.204	75.69	63.12	0.892	2

7 Conclusion and Future Work

The main goal of this study was to compare pretrained models for text classification, summarization and generation, on the wikitext dataset. We were able to rank our pretrained models using TOPSIS based on all the 3 tasks. GPT-3.5 performed much better than all the models it was compared to for all the 3 tasks, with Facebook's Llama-65B coming close in some of the tasks.

While our study provides valuable insights into the performance of various pretrained models for text classification, generation, and summarization, there are several avenues for further investigation and improvement

Future research can try transfer learning approaches using pretrained models from related domains or languages can also improve model adaptability, particularly when fine-tuned on smaller datasets with domain-specific annotations. Also multimodal pretraining extends analysis to include models combining text with other modalities like images, videos, or audio, showing promise in applications such as image captioning and video summarization can be used to enhance the results. Evaluating multimodal pretrained models on joint tasks like text-image alignment or cross-modal retrieval further expands their utility can be done as well. Ethical considerations delve into biases in pretrained models, necessitating mitigation strategies during fine-tuning or post-processing to ensure fairness and equity. Task-specific architectures and domain adaptation techniques can further enhance model performance by leveraging pretrained representations and adapting quickly to new tasks with minimal labeled data.

References

1. Basyal, L., Sanghvi, M.: Text summarization using large language models: a comparative study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT models. In: Proceedings of the 2024 International Conference on Natural Language Processing, pp. 123–136. IEEE (2024). https://doi.org/10.48550/arXiv.2310.10449
2. Liu, Y., Shi, K.: On learning to summarize with large language models as references. IEEE Trans. Natural Lang. Process. **42**(3), 123–136 (2024). https://doi.org/10.48550/arXiv.2305.14239

3. Liu, Y., Lapata, M.: Fine-tuning large pretrained language models for abstractive summarization. In: Proceedings of the 2024 IEEE International Conference on Natural Language Processing, pp. 237–250 (2024). https://ijisae.org/index.php/IJISAE/article/view/4500
4. Li, Z., Zhu, H.: Synthetic data generation with large language models for text classification. IEEE Trans. Natural Lang. Process. **42**(3), 123–136 (2023). https://doi.org/10.18653/v1/2023.emnlp-main.647
5. Van Veen, D., et al.: Adapted large language models can outperform medical experts in clinical text summarization. IEEE Trans. Natural Lang. Process. **42**(3), 237–250 (2024). https://doi.org/10.1038/s41591-024-02855-5
6. Text Classification via Large Language Models. https://doi.org/10.48550/arXiv.2305.08377
7. Arslan, Y., et al.: A comparison of pre-trained language models for multi-class text classification in the financial domain. In: Companion Proceedings of the Web Conference 2021, WWW 2021, pp. 260–268. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3442442.3451375
8. Avrahami, O., et al.: SpaText: spatio-textual representation for controllable image generation. CoRR arxiv: 2211.14305 (2022). https://doi.org/10.1109/CVPR52729.2023.01762
9. Cheng, J., Liang, X., Shi, X., He, T., Xiao, T., Li, M.: LayoutDiffuse: adapting foundational diffusion models for layout-to-image generation. CoRR arxiv:2302.08908 (2023). https://doi.org/10.48550/arXiv.2302.08908
10. Beltagy, I., Lo, K., Cohan, A.: SciBERT: a pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3606–3036 (2019). https://doi.org/10.18653/v1/D19-1371
11. Ye, J., et al.: A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. arXiv preprint arXiv:2303.10420/arXiv.2303.10420 (2023)
12. Touvron, H., et al.: LLaMA: open and efficient foundation language model (2023). https://doi.org/10.48550/arXiv.2302.13971

Predicting Judgement Outcomes from Legal Case File Summaries with Explainable Approach

Hima Varshini Surisetty, Sarayu Varma Gottimukkala, Priyanka Prabhakar, and Peeta Basa Pati(✉)

Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India
bp_peeta@blr.amrita.edu

Abstract. Legal Outcome Prediction (LOP) is the process of predicting the possible outcome of a court case based on the case file. The aim of this work is to assess the performance of four deep learning models namely BERT, LegalBERT, DistilBERT and RoBERTa. These models were trained and tested with complete case files and their summaries. The models were then evaluated using metrics like train-test accuracy, loss plots and F-score. Based on these evaluation metrics, RoBERTa was chosen as the model with the best performance. In order to understand the predictions made by the deep learning models, Local Interpretable Model-Agnostic Explanations (LIME) was used. LIME is an eXplainable AI (XAI) tool that highlights the words in a case file which were used in order to predict the outcome of that case.

Keywords: Legal Outcome Prediction · Case File · Case Summary · XAI · LIME

1 Introduction

People dealing with legal problems are generally under a lot of stress due to the lack of knowledge in the legal domain. Attorneys are legal practitioners trained to handle various cases and provide assistance to their clients; their advice may not be the most efficient due to their personal biases. The general public needs a way to understand their case and possible outcome without having to worry about any biases from their lawyers.

Earlier reported works in the area of outcome prediction mainly consist of basic machine learning and deep learning models like Support Vector Machine (SVM), Convolution Neural Networks (CNN) etc. The basic models are not trained to handle complex and lengthy documents. Hence, Large Language Models have come into the picture. LLM's are trained to handle large amounts of textual data in a quick and efficient manner. A few examples of LLM's include BERT, GPT etc. Specifically in legal domain, these models have been used for outcome prediction by providing the case files as input to the models. But in real time scenarios, the complete case file is not always provided to the

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 167–183, 2025.
https://doi.org/10.1007/978-3-031-78107-0_11

client. Instead, a summarized version of the case file is provided. For this reason, the models should be able to handle complete as well as summarized case files.

There are different types of summarizations based on the use case. For example, topic-based and indicator-based, single document and multiple documents, abstractive and extractive etc. The most popular type of summarization that is being used in recent works in this area is the abstractive-extractive summarization. The difference between these two types of summarizations is the way the sentences from the complete text are placed in the summary. In abstractive summarization, the sentences are paraphrased to generate a more meaningful summary while in extractive summary, the sentences are picked from complete text and placed directly into the summary.

Earlier works have shown that several summarizers have been developed in the legal domain. For example, Case Summarizer is a custom summary generation technique that was developed to handle legal case files. Automatic summarization and outcome prediction are growing areas of research in the legal domain. However, the outcomes of such research have started getting integrated into different platforms which are used for legal services. Yet there is a general hesitancy shown by legal practitioners across the world toward using such a technique. The reason for such behavior is that the legal practitioners consider the technologies as a black box wherein the output is generated but the reasoning for that outcome is unknown. With the help of explainable AI, the inner workings of a model and the output generated by it can be understood. This provides a way for legal practitioners and the public to understand and trust the models used.

In the proposed work, a custom dataset was constructed by collecting judgments from the Indian Kanoon website, a comprehensive legal database for Indian court cases. Each collected judgment was manually labeled based on its outcome. This approach, while necessitated by the unique nature of the Indian legal system and the specificity of the research goals, also presents an opportunity to contribute a novel dataset to the field of legal analytics in the Indian context. A cluster analysis is done on the dataset in order to understand the patterns in the data. The clusters generated are evaluated using Silhoutte score, Calinski Harabasz score and Davies Bouldin score. After cluster analysis, four deep learning models namely BERT, LegalBERT, DistilBERT and RoBERTa are trained with complete data, abstractive summarized data and extractive summarized data which are generated using spacy module and Lex Rank algorithm respectively. The models are evaluated using train accuracy, test accuracy, loss plot and F-score. Based on the evaluation metrics, the best model is chosen. In order to understand why a model predicted a certain outcome, explainable AI was used. LIME is an XAI tool which highlights the words or phrases in the text that was used by a model to generate a certain outcome. The dataset that has been used in this work has been created specifically to solve this problem and contributes to the novelty of this work. The research questions addressed in this work are:-

RQ1: Are summarized case files as efficient in predicting outcomes as detailed case files?
RQ2: Does integrating LIME explanations into the model's predictions assist in assessing the legal significance of terms utilized in the model's decision-making?

Legal outcome prediction aligns with the objectives of United Nation's SDG 10 (reduce inequality) & 16 (particularly in promoting the rule of law). The key contributions of this work are listed below.

- A parallel corpus of case files from Indian courts and extractive as well as abstractive summaries for each case file.
- Identification of a large language model that performs LOP task with equal effectiveness for complete text case files as well as summary text.

2 Literature Review

This section represents the information obtained through a detailed survey of various papers in the field of legal outcome prediction and text summarization. Text Summarization is the process of shortening a lengthy text document while retaining the most important information. This has become increasingly more demanding and necessary in modern times due to the abundance of information available online. Manually analyzing all the information available is a costly and intensive process. In order to make it easier for a person to find relevant information quickly, text summarization was introduced. There are different ways of performing text summarization based on the type of information that is being dealt with [1, 2]. Few of the most popular ways of performing text summarization include topic-based and indicator-based summarization. In topic-based text summarization, initially the main topics or ideas of the text are captured. Based on these topics, the sentences in the text are ranked based on their importance. Whereas, in indicator-based text summarization, rather than focusing on topicality, the text is defined as a set of possible indicators which are then used to rank the importance of sentences in the text. In both these techniques, once the sentences are ranked based on their importance in the text, either a greedy approach or a global optimization approach is used to create the summary [3].

Text summarization is one of the most important fields in the legal domain. Legal practitioners manually summarize a large amount of case files in order to understand their current case. It is a time taking process that can be made easy and quick with recent technological advancements. There have been many summarization tools that were developed in order to help with this problem. Though, very few legal practitioners have started using these tools [4]. There are two main types of summaries that can be generated namely extractive and abstractive summary. The main difference between both the types of summaries is the way that the text in the summary is generated. In extractive summary, sentences are picked directly from the main text and placed in the summary. Whereas, in abstractive summary, the sentences that are picked from the main text are paraphrased before being placed in the summary. Both these summaries have their own advantages in different tasks [5].

Nowadays, transformers are also being used widely for text summarization. For example, T5 transformer was trained and tested with data taken from Supreme Court of India [6]. Recent studies show that basic machine learning models cannot handle text summarization as well as deep learning models due to the large amounts of data that needs to be processed. Therefore, there are many deep learning models that have been developed specifically for text summarization. While there are some models that

have been fine tuned for classification and text summarization [7]. Once the summaries of text have been obtained, the next main area of research in legal domain is outcome prediction from the case files and their summaries. Some studies have tested machine learning models like Support Vector Machine (SVM) for this task. The results obtained showed that the model gave a 98% F1 score while predicting a case ruling. It was also noted that this model can perform tasks other than outcome prediction like predicting the law area or the date of ruling for a case [8]. Along with machine learning models, deep learning models like Convolution Neural Networks (CNN's) have also been used for outcome prediction otherwise known as verdict prediction [9].

Along with using models for text classification, summarization etc., they need to be evaluated. There are many metrics to evaluate the models but some of the metrics that are used popularly to evaluate text summarization include ROGUE score, BLEU score etc. These metrics belong to the Natural Language Processing (NLP) field. They measure the semantics and sentence formations in the summary rather than how accurate they are. This helps the developers understand how effectively the models are able to form sentences in the summary based on the text [10]. BERT is a Large Language Model (LLM) that is widely used in most NLP tasks since it is pretrained on a large amount of text data. Not only can BERT be used directly to perform classification, prediction etc., but it can also be finetuned for custom problems. Due to this feature, many developers have created several versions or types of BERT by pretraining it with different types of data. Some examples are LegalBERT, which is trained on legal case files, BioBERT, which is trained on biological data, AlphaBERT, which is trained on hospital information etc. These types of BERT are publicly available for anyone to use easily [11]. One such example of BERT is RoBERTa. The main difference between BERT and RoBERTa is the size of the training data. RoBERTa was trained on a larger data as compared to BERT base model [12]. Understanding the reason for a certain outcome is as important as predicting that outcome.

In order to achieve this, explainable AI is used. XAI tools highlight the features in the input that were used by the model to generate the output. In case of legal outcome prediction, XAI tools can be used to highlight the words in the case file that were considered by the model to predict the outcome of that case. This helps us understand whether the model is able to analyze the right words, or it is randomly predicting the outcome. It can be helpful in cases where the accuracy of the model is high but when the test cases are used, the model is unable to predict the correct outcome in most cases [13, 14]. Some popular XAI tools include LIME and ELI5. Both these tools have similar functionality on a local level wherein they highlight the text that led to certain predicted outcome. But they differ on a global level wherein ELI5 can also highlight the parameters of a model that were used to reach a certain outcome. This feature is not available in LIME [15].

Most legal practitioners have not started making use of deep learning or machine learning models for outcome prediction because they doubt the ability of a model to predict an outcome of a case since, they cannot understand why the model has predicted a certain outcome. In cases like this, XAI is helpful is providing an insight into the words or phrases that contributed to getting an outcome. This helps legal practitioners to verify the output manually if necessary [16].

3 Data Description

This section talks about the dataset used for the proposed work and how it was created. The data used in this work was procured from Indian Kanoon website which is a repository of judgements from across various courts in India. The case files were randomly picked from a span of seven decades between 1950–2023. A total of 600 records were collected and stored. These cases were divided into four main classes namely appeal allowed, appeal dismissed, petition dismissed, and petition allowed. The classes were encoded into values of 0, 1, 2 and 3 that represent each class respectively. The class distribution in the dataset is represented in Fig. 1.

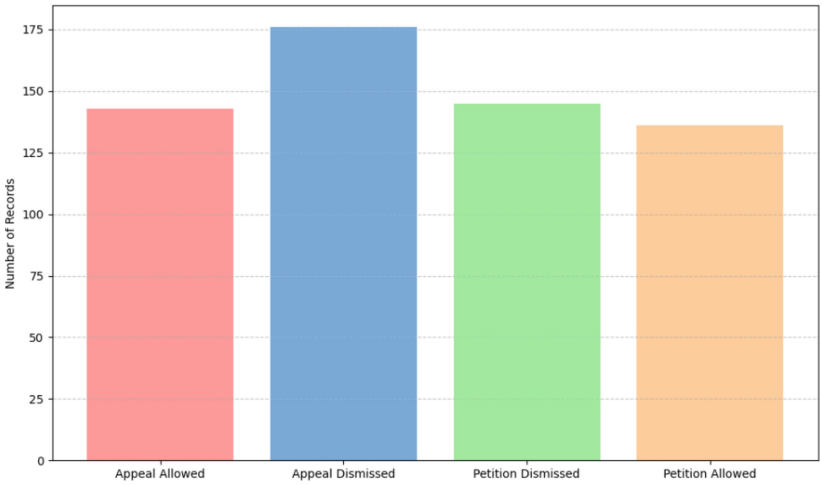


Fig. 1. Bar Chart representing class distribution.

The summaries of these case files are not available in the Indian Kanoon website. The abstractive summaries are generated using Spacy module while the extractive summaries are generated using LexRank algorithm. The spacy module and LexRank algorithm were chosen based on the studies in [17, 18]. The summaries were later analyzed by one of the authors who is a legal professional. The dataset was divided into 75:25 ratio for training and testing. These training and testing sets were used for pre-training as well as fine tuning the four deep learning models [19].

4 Proposed Methodology

This section depicts the process workflow of the proposed work. The workflow of the proposed system is represented in Fig. 2. The workflow is divided into three sections namely data pre-processing, classification models and explainable AI.

The proposed workflow begins with the text preprocessing block. Initially, the data is used to generate clusters which are analyzed and evaluated using Silhoutte score, Calinski Harabasz score and Davies Bouldin score. This is done in order to understand

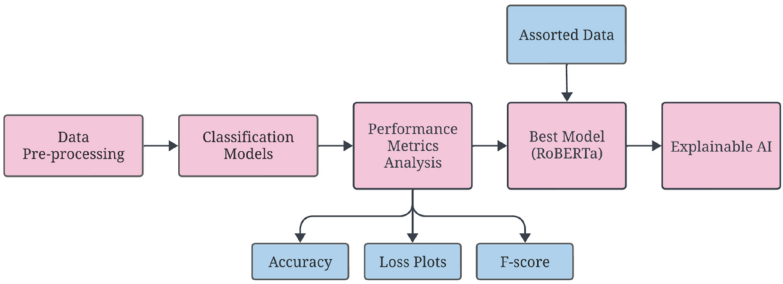


Fig. 2. Proposed Workflow

the patterns in the data and to make sense of the complex dataset. Then, the complete case files and their summaries are given to the data pre-processing block to generate the word embedding representations. These word embedding representations are then given to four classification models namely BERT, LegalBERT, DistilBERT, RoBERTa. The models are evaluated using train accuracy, test accuracy, loss plots and F-score. Based on the results of these evaluation metrics, the best model (RoBERTa) is selected. The best model is trained and tested with assorted data which is a combination of complete and summarized data. The model is then given to LIME which is an explainable AI tool. This tool highlights the words or phrases in the input text which contributed to the outcome predicted by the model.

4.1 Text Pre-processing

The text pre-processing block takes the complete case files and their summaries as input. The complete case files are initially tokenized then the stop words are removed. Then the punctuation marks are removed, and the output is given to the stemming block. Finally, the word embedding representations are generated as output. These word embeddings are given to the classification blocks as input. The summarized case files do not require the stop word removal, punctuation removal and stemming since they are a shorter version of the case files. So, the summaries are tokenized and their word embedding representations are generated while skipping the intermediate steps (Fig. 3).

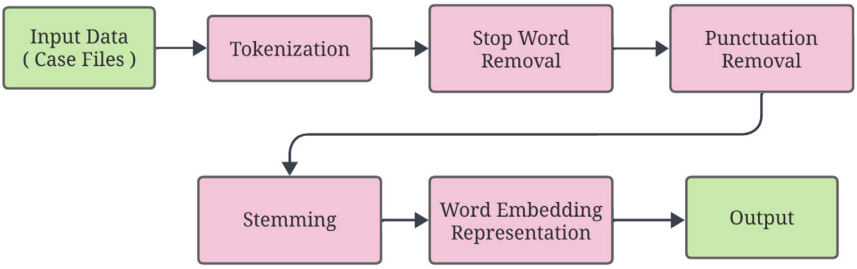


Fig. 3. Text Pre-processing block

4.2 Classification Models

The architecture of the four deep learning models is represented in Table 1. The model architecture consists of different layers such as max pooling, dense, dropout and embedding layer. The output shape of each layer is mentioned. The output shape of the final dense layer is the same which is (None, 4). This represents the number of classes in the dataset.

Table 1. Model Architecture.

Model	BERT	LegalBERT	DistilBERT	LegalBERT
Embedding (Output Shape)	TFBertModel	TFSequenceClassifier	TFDistilBertModel	TFSequenceClassifier
Max Pooling1D (Output Shape)	(None, 768)	–	–	–
Dense (Output Shape)	(None, 128)	(None, 32)	(None, 512)	(None, 128)
Dropout (Output Shape)	(None, 128)	(None, 128)	(None, 512)	(None, 128)
Dense (Output Shape)	(None, 4)	(None, 4)	(None, 4)	(None, 4)

4.3 Explainable AI

The deep learning model with the best performance which was observed to be the RoBERTa model was given to LIME which is an explainable AI tool that highlights the words in the text that contributed to outcome predicted by the model. LIME stands for Local Interpretable Model-agnostic Explanations. This tool provides an explanation for a single instance, so it is suitable for local interpretation.

5 Results

This section reports the results of training and testing the four different deep learning models on complete, summarized, and assorted data. The results section is divided into three sub sections namely cluster analysis, LOP with complete and summarized data and explainable AI.

5.1 Cluster Analysis

Before training the deep learning models, a cluster analysis was done on the dataset in order to understand the data in detail. The clusters were created using Kmeans clustering and they were evaluated using metrics like Silhoutte score, Calinski Harabasz score and Davies Bouldin score. The number of clusters in the data is not known beforehand so the elbow curve method is used to determine the optimal number of clusters. The graph depicting the elbow curve for the dataset used in this work is shown in Fig. 4.

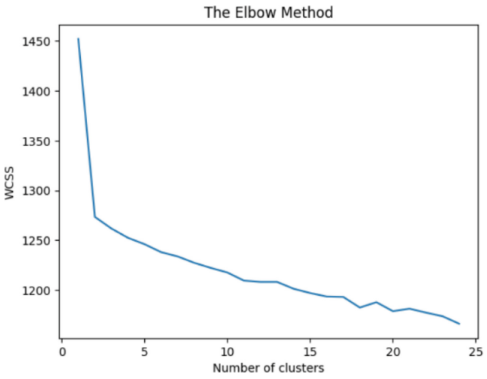


Fig. 4. Graph depicting elbow curve

From Fig. 4, it can be inferred that the curve is plotted by having Within Cluster Sum of Square (WCSS) on the y-axis and the number of clusters in the x-axis. WCSS is calculated by measuring the distance between each data point and the centroid and summing their squares. The optimal number of clusters is found by identifying the point in the curve which forms an elbow shape. In the proposed work, the optimal number of clusters obtained from the elbow curve is 2. Once the optimal number of clusters is known, Kmeans clustering is used to create the clusters. Principal Component Analysis (PCA) is used for dimensionality reduction. The number of dimensions is set to two because it is easier to visualize the clusters in 2-D space. The clusters and their centroids are shown in Fig. 5. The dimensionality of the data points is also reduced using PCA and the corresponding PCA elements are generated. PCA transforms the original features into a new set of linearly uncorrelated variables called principal components. These principal components are linear combinations of the original features.

In Fig. 5, the plot contains principal component 1 in the x-axis and principal component 2 in the y-axis. The principal components 1 and 2 represent the maximum variance and second maximum variance in the data. These clusters are evaluated using Silhoutte score, Calinski Harabasz score and Davies Bouldin score. The results of the cluster analysis are shown in Table 2.

The Silhoutte score determines how well suited a data point is in its own cluster as opposed to its neighboring clusters. The scores can range from –1 to 1. A higher Silhoutte score means that the data point is well matched in its own cluster and poorly matched in its neighboring cluster. From Table 2, it can be observed that the Silhoutte score is 0.15

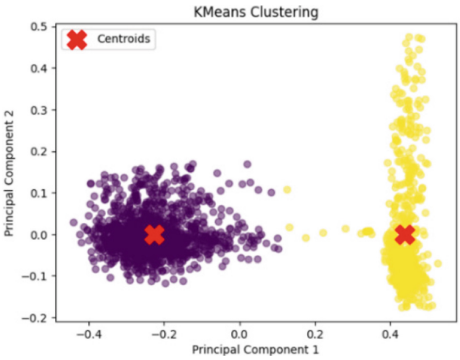


Fig. 5. Kmeans clusters and their centroids

Table 2. Cluster Evaluation Metrics.

Cluster Evaluation Measures	Scores
Silhouette Score	0.15
Calinski Harabasz Score	252.26
Davies Bouldin Score	2.59

which is an average score. It can be inferred that the data points are fairly well matched in their own clusters. The Calinski score is the ratio of between cluster and within cluster dispersion. There is no range of values for the Calinski score but a higher value indicates more well-defined clusters. From Table 2, it can be observed that the Calinski score is 252.26 which indicates that the clusters are well separated, and this can also be observed in Fig. 5. The Davies Bouldin score measures the similarity between each cluster and its most similar cluster. A lower Davies Bouldin score indicates good separation between the clusters. From Table 2, it can be observed that the Davies Bouldin score is 2.59 which is a good score so it can be inferred that the clusters are well separated. Based on the results of all the cluster evaluation measures, it can be concluded that the clusters are well defined and separated and the data points in the clusters are well suited to their own clusters. The cluster analysis helps understand the underlying structure of the dataset used.

5.2 LOP with Complete and Summarized Data

The four deep learning models namely BERT, LegalBERT, DistilBERT and RoBERTa were initially trained and tested with complete and summarized data. The best model was selected based on the evaluation metrics like accuracy, loss plots and F-score. The results obtained from the models when the complete data, abstractive summarized data and extractive summarized data are given as input are shown in Fig. 6.

Architecture	Complete Data			Abstractive Summarized Data			Extractive Summarized Data		
	Accuracy (%)		F1 Score	Accuracy (%)		F1 Score	Accuracy (%)		F1 Score
	Train	Test		Train	Test		Train	Test	
BERT	97.77	45.64	0.44	99.77	36.91	0.35	99.55	40.94	0.41
LegalBERT	55.18	40.27	0.26	56.98	40.27	0.32	52.93	41.61	0.27
DistilBERT	81.53	40.94	0.39	96.85	36.24	0.36	97.97	40.94	0.41
RoBERTa	**62.16**	**48.99**	**0.47**	**95.27**	**36.91**	**0.35**	**97.07**	**44.31**	**0.43**

Fig. 6. Results of four deep learning models with complete and summarized data

In Fig. 6, the accuracy and F1-score of the four deep learning models are shown when the different types of data are given as input. Based on these results, the RoBERTa model was chosen as the best model. This can be observed from the highlighted values in Fig. 6. The test accuracy and F-score of RoBERTa were better than the other models for complete and summarized data. It can also be observed from the green and blue highlighted values that the extractive summarized data was as efficient of an input to the RoBERTa model as complete data. But the abstractive summarized data was not on par with the complete data which can be observed from the orange highlighted values. Based on these values, RoBERTa was chosen as the best model.

5.3 LOP with Assorted Data

Since RoBERTa was chosen as the best model, it was trained with assorted data. The model was then evaluated using the same metrics as the other types of data. The results are shown in Table 3.

Table 3. Results of RoBERTa model with assorted data.

	Assorted Data		
RoBERTa	Accuracy (%)		F1 Score
	84.12	49.78	0.47

It can be observed from Fig. 6 and Table 3 that the assorted data performs slightly better than the complete data. This was the expected behavior as the assorted data consists of the complete and summarized data together. The highest F-score was 0.47 which belonged to the RoBERTa model when the complete and assorted data were given as inputs. Since assorted data performed the best, the accuracy plot loss plot of the RoBERTa model with assorted data is shown in Fig. 7(a) and (b).

From Fig. 7 (b), it can be observed that the gap between the training and validation loss is initially less and later the training loss decreases while the validation loss increases steadily. This is the expected behavior which indicated that the model is able to learn well from the seen data and the since the loss is varying, the model is able to learn new information from the unseen data as well. After the models are tested and evaluated, they are given to LIME to get the explanations for the predictions.

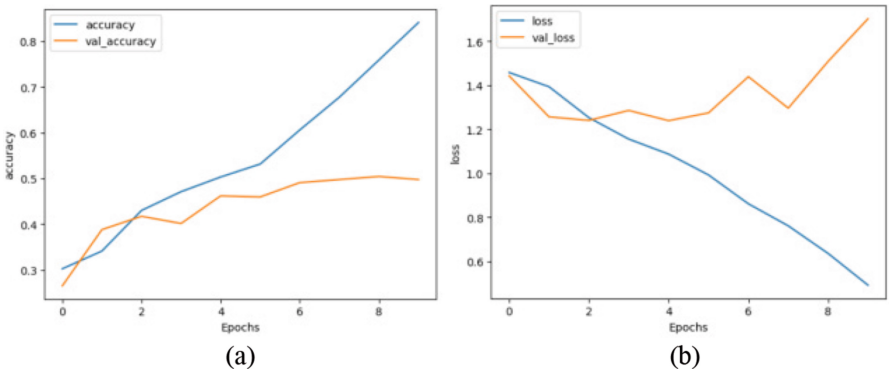


Fig. 7. Accuracy plot (a) and Loss plot (b) of RoBERTa model with assorted data as input

5.4 Explainable AI

The LIME model takes the deep learning model and one instance as an input. It provides the outcome prediction from the model and the explanations for that prediction as the output. Since RoBERTa was chosen as the best model, LIME was implemented with RoBERTa model when the complete data, summarized data and assorted data were given as inputs. LIME shows the probability for the prediction of each class and the top 10 words that contributed to the prediction.

The LIME explanations when different types of data were given as input to the RoBERTa model are shown in Table 4, 5, 6 and 7. The column titled "words against prediction of class" refers to the words among the top 10 words that contributed to the respective class not being predicted. Whereas the column titled "words for prediction of class" refers to the words among the top 10 words that contributed to the prediction of the respective class. The LIME explanations for the RoBERTa model when complete data is given as input is shown in Table 4.

From Table 4, the prediction of the outcome is highlighted in the table which is class '1' in this case. The probability column shows the percentage of the data having an outcome of 0, 1, 2 or 3. The highest probability is 69% for class '1' which is the reason the final prediction was class '1'. The words that contributed to the prediction of class '1' were 'incometax', 'appeal', 'refer' and 'date'. The text in the case file is shown in Fig. 8. The words in the text that contributed to the model predicting the outcome as '1' are also highlighted in Fig. 8.

Class 1 corresponds to 'Appeal Dismissed' and it can be observed from Fig. 8 that the highlighted words in the text include the word 'appeal'. This indicates that the model is focusing on relevant words while making a prediction.

The LIME explanations for the RoBERTa model when abstractive summarized data is given as input is shown in Table 5.

From Table 5, the prediction of the outcome is highlighted in the table which is class '1' in this case. The probability column shows the percentage of the data having an outcome of 0, 1, 2 or 3. The highest probability is 99% for class '1' which is the reason the final prediction was class '1'. The words that contributed to the prediction of class '1' were 'held', 'void, 'It, 'I, etc. The text in the case file is shown in Fig. 9. The words in

Table 4. LIME explanations of RoBERTa model with complete data (weights of words added in brackets).

	Words against prediction of class	Words for prediction of class	Probability (%)
CLASS - 0	incometax (0.11), refer (0.01)	calcutta (0.06), court (0.05), high (0.05), judgement (0.04), appel (0.04), chakravartti (0.03), gupta (0.03)	31
CLASS - 1	**judgement (0.04), high (0.03), court (0.03), calcutta (0.03), court (0.03), chakravartti (0.03)**	**incometax (0.08), appeal (0.05), refer (0.01), date (0.01)**	**68**
CLASS - 2	appeal (0.02), calcutta (0.01), court (0.01), appel (0.01), high (0.01), gupta (0.001), judgement (0.001), chakravartti (0.001)	incometax (0.01)	0
CLASS - 3	appeal (0.03), calcutta (0.02), court (0.02), appel (0.01), high (0.01), gupta (0.01), judgement (0.01), chakravartti (0.01)	incometax (0.02)	1

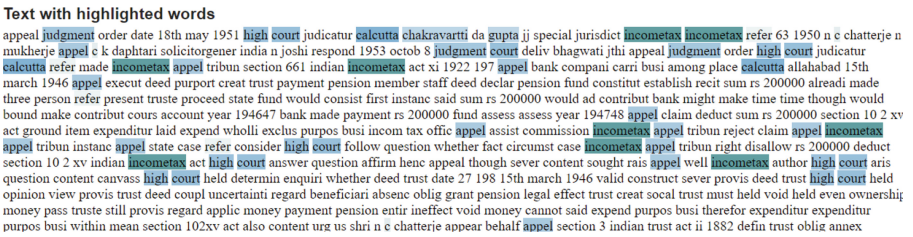


Fig. 8. Top 10 words from LIME highlighted in the text from complete case file

the text that contributed to the model predicting the outcome as '1' are also highlighted in Fig. 9.

From Fig. 9, it can be observed that the model is mainly focusing on stop words like 'of', 'that' etc. This is reflected in Fig. 6 since the abstractive summarized data has the least accuracy and F-score compared to all other types of data for all the models. This behaviour of the model can be explained using the LIME explanations.

The LIME explanations for the RoBERTa model when extractive summarized data is given as input is shown in Table 6.

From Table 6, the prediction of the outcome is highlighted in the table which is class '1' in this case. The probability column shows the percentage of the data having an outcome of 0, 1, 2 or 3. The highest probability is 98% for class '1' which is the reason

Table 5. LIME explanations of RoBERTa model with abstractive summarized data (weights of words added in brackets).

	Words against prediction of class	Words for prediction of class	Probability (%)
CLASS - 0	held (0.19), void (0.14), It (0.14), of (0.12), I (0.12), deed (0.12), that (0.10), court (0.08), provisions (0.07), further (0.05)	–	1
CLASS - 1	**–**	**held (0.26), void (0.17), It (0.17), I (0.14), deed (0.11), that (0.16), court (0.10), provisions (0.09), the (0.18)**	**99**
CLASS - 2	the (0.05), held (0.03), trust (0.03), that (0.02), had (0.02), On (0.01), passed (0.01)	deed (0.01), of (0.001)	0
CLASS - 3	the (0.07), trust (0.04), held (0.03), that (0.03), had (0.02), It (0.02), passed (0.02), money (0.01), ownership (0.01)	deed (0.01)	0

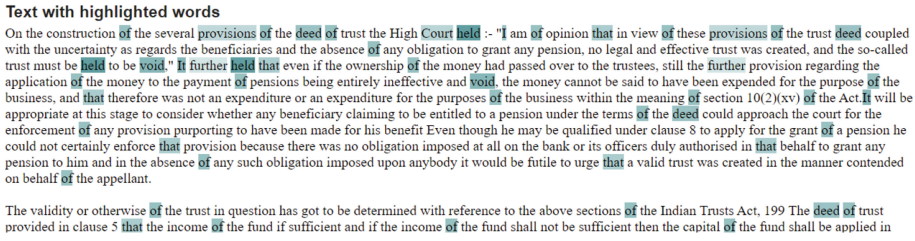


Fig. 9. Top 10 words from LIME highlighted in the text from abstractive summarized case file

the final prediction was class '1'. The words that contributed to the prediction of class '1' were 'of', 'court', 'calcutta', 'the', etc. The text in the case file is shown in Fig. 10. The words in the text that contributed to the model predicting the outcome as '1' are also highlighted in Fig. 10.

From Fig. 10, it can be observed that for an outcome of 'Appeal Dismissed', the model is able to focus on relevant words like 'appeal', but it is also focusing on some stop words like 'of','the' etc. So, the model is able to perform better with extractive summarized data as compared to abstractive summarized data.

The LIME explanations for the RoBERTa model when assorted data is given as input is shown in Table 7.

From Table 7, the prediction of the outcome is highlighted in the table which is class '2' in this case. The probability column shows the percentage of the data having an outcome of 0, 1, 2 or 3. The highest probability is 57% for class '2' which is the reason

Table 6. LIME explanations of RoBERTa model with extractive summarized data (weights of words added in brackets).

	Words against prediction of class	Words for prediction of class	Probability (%)
CLASS - 0	of (0.15), 1922 (0.06), provisions (0.05), held (0.04), that (0.04), the (0.04), court (0.04), I (0.04)	appeal (0.06)	1
CLASS - 1	**–**	**of (0.21), court (0.14), calcutta (0.14), 1922 (0.11), held (0.10), the (0.09), that (0.08), bhagwati (0.07), judgement (0.07), the (0.07)**	**98**
CLASS - 2	Calcutta (0.03), court (0.03), appeal (0.03), Bhagwati (0.02), 1922 (0.02), of (0.02), deed (0.02), judgement (0.02), held (0.02), high (0.01)	–	0
CLASS - 3	Calcutta (0.08), Bhagwati (0.07), court (0.07), appeal (0.07), high (0.05), held (0.04), judgement (0.04), the (0.04)	Judicature (0.03), Sect. (0.01)	1

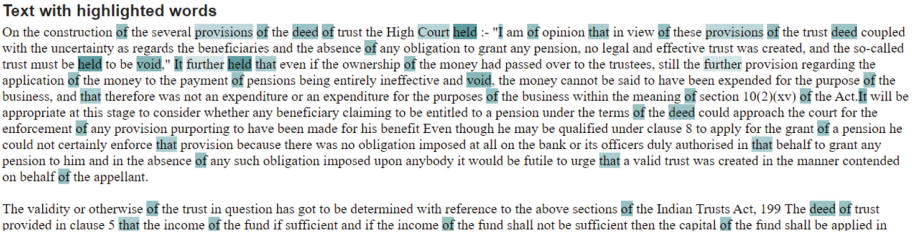


Fig. 10. Top 10 words from LIME highlighted in the text from extractive summarized case file

the final prediction was class '2'. The words that contributed to the prediction of class '2' were 'petitioners', 'changed, 'property, 'code', etc. The text in the case file is shown in Fig. 11. The words in the text that contributed to the model predicting the outcome as '2' are also highlighted in Fig. 11.

From Fig. 11, it can be observed that for an outcome of 'Petition Dismissed', the model is able to focus on relevant words like 'petitioners' and the model ignores most of the stop words. This indicates that the RoBERTa model performs better with assorted data as compared to summarized and complete data. This is reflected in the results obtained in Fig. 6. From all the LIME explanations, it can be inferred that the RoBERTa model performs well with complete and assorted data as compared to summarized data. Among

Table 7. LIME explanations of RoBERTa model with assorted data (weights of words added in brackets).

	Words against prediction of class	Words for prediction of class	Probability (%)
CLASS - 0	petitioners (0.11), application (0.09), petitioner (0.03)	property (0.04), changed (0.02), 1994 (0.02), the (0.01), called (0.01), injunction (0.01), court (0.01)	1
CLASS - 1	petitioners (0.25), application (0.09), petitioner (0.07), property (0.02), changed (0.02)	the (0.03), appellate (0.03), 39 (0.03), 1994 (0.03), civil (0.03)	1
CLASS - 2	**suit (0.04), appellate (0.04), respect (0.02)**	**petitioners (0.30), changed (0.09), property (0.08), code (0.04), passed (0.03), procedure (0.03), application (0.03)**	**57**
CLASS - 3	property (0.11), changed (0.10), how (0.05), court (0.05), the (0.04), evidence (0.04)	application (0.14), petitioner (0.08), petitioners (0.06), suit (0.02)	42

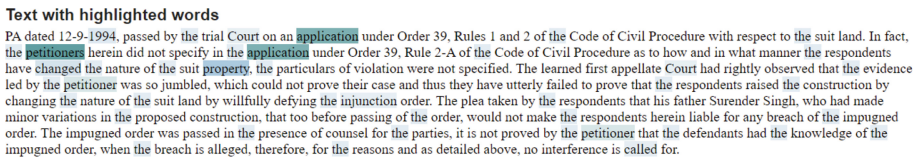


Fig. 11. Top 10 words from LIME highlighted in the text from assorted case file

abstractive and extractive summarized data, the model performs better with extractive summarized data.

6 Conclusion and Discussion

In conclusion, this work proposed a system that can predict the outcome of a case from both complete and summarized data. This system can be utilized in order to get a second opinion about the possible outcome of a legal case. In the proposed work, Kmeans clustering was used to generate the clusters from the dataset. These clusters were evaluated using Silhoutte score, Calinski Harabasz score and Davies Bouldin score. Four deep learning models were built using complete and summarized data. Based on the performance metrics of these models, the model with the best performance was chosen which was the RoBERTa model. This model was then built using assorted data. The performance metrics of the RoBERTa model with different types of data as inputs was analyzed. The research questions addressed in this work were answered based on

the analysis of the performance metrics of the best model on different types of data. It was observed that the extractive summarized data works as efficiently as the complete data since the RoBERTa model gave almost similar accuracies and F1 score when the extractive summarized data and complete data were given as inputs. It was also observed that LIME was able to identify the words used in the model's decision-making process. The relevance and meaningfulness of these words were determined based on the outcome predictions that the model made. It was observed that the model was able to predict more accurate outcomes when the words used by the model contained legal terms rather than stop words, conjunctions etc.

The proposed Indian legal outcome prediction system faces significant challenges. First, the limited size of the dataset constrains the model's ability to capture the full spectrum of legal scenarios and outcomes. This restriction may reduce accuracy and generalizability, particularly for less common cases or unique legal situations. Second, the inherently complex and subjective nature of legal decision-making poses a formidable obstacle to purely data-driven predictions. Dynamic factors such as evolving societal norms, individual judicial interpretations, and the nuanced application of precedents often influence legal outcomes. These elements are difficult to quantify making it challenging for the system to consistently produce accurate predictions across changing legal landscapes.

References

1. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**, 1–66 (2017)
2. Ramachandran, R., Jayachandran, S., Das, V.: A novel method for text summarization and clustering of documents. In: IEEE 3rd Global Conference for Advancement in Technology (GCAT) (2022)
3. Nenkova, A., McKeown, K.: A survey of text summarization techniques. Mining Text Data (2012)
4. Kanapala, A., Pal, S., Pamula, R.: Text summarization from legal documents: a survey. Artif. Intell. Rev. **51**, 371–402 (2019)
5. Jain, D., Borah, M.D., Biswas, A.: Summarization of legal documents: where are we now and the way forward. Comput. Sci. Rev. **40**, 100388 (2021). https://doi.org/10.1016/j.cosrev.2021.100388
6. Prabhakar, P., Gupta, D., Pati, P.B.: Abstractive summarization of Indian legal judgments. In: OITS International Conference on Information Technology (OCIT) (2022)
7. Sheik, R., Nirmala, S.J.: Deep learning techniques for legal text summarization. In: 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (2021)
8. Sulea, O.M., Zampieri, M., Malmasi, S.: Exploring the use of text classification in the legal domain. In: Computation and Language (2017)
9. Pillai, V.G., Chandran, L.R.: Verdict prediction for Indian courts using bag of words and convolutional neural network. In: Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (2020)
10. Yousefi-Azar, M., Hamey, L.: Text summarization using unsupervised deep learning. Expert Syst. Appl. **68**, 93–105 (2017). https://doi.org/10.1016/j.eswa.2016.10.017
11. Mohammed, A.H., Ali, A.H.: Survey of BERT (bidirectional encoder representation transformer) types. In: Journal of Physics: Conference Series (2021)

12. Liu, Y., Ott, M., Goyal, N., et al.: RoBERTa: a robustly optimized BERT pretraining approach. In: Computer Science, Computational Language (2019)
13. Norkute, M., Herger, N., Michalak, L.: Towards explainable AI: assessing the usefulness and impact of added explainability features in legal document summarization. In: Conference on Human Factors in Computing Systems (2021)
14. Ganeshkumar, M., Vinayakumar, R., Sowmya, V., Gopalakrishnan, E.A., Soman, K.P.: Explainable deep learning-based approach for multilabel classification of electrocardiogram. IEEE Trans. Eng. Manag. **70**, 2787–2799 (2023)
15. Vij, A., Nanjunadan, P.: Comparing strategies for post-hoc explanations in machine learning models. In: Mobile Computing and Sustainable Informatics (2021)
16. Mahoney, C.J., Zhang, J., Huber-Fliflet, N., Gronvall, P., Zhao, H.: A framework for explainable text classification in legal document review. In: IEEE International Conference on Big Data (Big Data) (2019)
17. Sharma, A., Aggarwal, R., Alawadhi, R.: A comparative study of text summarization using gensim, NLTK, spacy, and sumy libraries. J. Xi'an Shiyou Univ. Nat. Sci. Ed. **19** (2023)
18. Deo, S., Banik, D.: Text summarization using textrank and lexrank through latent semantic analysis. In: OITS International Conference on Information Technology (OCIT) (2022)
19. Surisetty, H.V., Gottimukkala, S.V., Priyanka, P., Peeta, B.P.: Comparative analysis of legal outcome prediction with detailed and summarized text. In: International Conference for Convergence in Technology (I2CT) (2024)

Multi-view Ensemble Clustering-Based Podcast Recommendation in Indian Regional Setting

Yashwant Pravinrao Bangde[1] and Naveen Saini[2(✉)]

[1] Indian Institute of Information Technology Lucknow, Lucknow, India
[2] Indian Institute of Information Technology Allahabad, Prayagraj, India
nsaini1988@gmail.com

Abstract. Podcasts are now a favored means of consuming audio content across various languages, yet the sheer volume of available podcasts poses a challenge for users in discovering content that matches their preferences. In this study, we propose unsupervised techniques for podcast recommendation in a multi-lingual context. Our novel approach integrates multi-view representations, encompassing both sentence-level and keyword-level perspectives, to capture the diverse facets of podcast content. Utilizing autoencoders, we derive meaningful latent representations for both sentence and keyword views from the podcast dataset. These representations encapsulate semantic relationships crucial for subsequent clustering analysis. Then we employ unsupervised learning algorithms on these learned representations to cluster similar podcasts together followed by ensemble learning. Our experimental evaluations on a varied multi-lingual podcast dataset (Hindi and English) showcase the promising performance of our approach in terms of podcast recommendation accuracy and user satisfaction. By leveraging multiple views and unsupervised learning techniques, we effectively address the challenges posed by language diversity and content heterogeneity in podcast recommendation systems. The results reveals that the ensemble algorithm emerges as a standout performer, achieving a silhouette score, diversity and coverage of (a) 0.6640, 0.6150, and 0.7720 for the English Dataset, and (b) 0.4860, 0.4620, and 0.8380 for the Hindi Dataset, outperforming other methods. These results underscore the efficacy of our multi-view ensemble clustering approach in tackling language diversity and content heterogeneity, thus advancing personalized podcast recommendation in the multi-lingual domain.

Keywords: Recommendation Systems · Podcasts · Unsupervised Learning · Clustering · Autoencoders

1 Introduction

Nowadays, podcasts have gained immense popularity as a medium for consuming audio content across various languages and genres. With the ever-increasing number of podcasts available, there is a pressing need for effective recommendation systems that can assist users in discovering relevant and personalized content

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 184–200, 2025.
https://doi.org/10.1007/978-3-031-78107-0_12

[3,27]. Recommendation systems can be broadly categorized into content-based filtering, collaborative filtering and hybrid. The first one suggests items similar to those a user has liked based on item features and user profiles. The second one encompassing user-based and item-based approaches that recommend items based on user interactions and similarities among users or items. The hybrid one combine the first two techniques for more accurate recommendations. Each type offers unique advantages and is chosen based on factors like available data and the complexity of the recommendation task.

The ease of accessing podcasts on portable devices such as smartphones, tablets, and laptops has had a significant impact on the popularity of podcasts, especially in countries like India. However, the multi-lingual nature of the podcast domain presents unique challenges in understanding and recommending content to users across different languages [19]. The emerging podcast industry in India, as highlighted in "PwC Global Entertainment & Media Outlook 2020–2024" c by PricewaterhouseCoopers (PwC) International Limited, underscores a prime opportunity for the development of podcast recommendation and summarization tools tailored to the country's linguistic diversity. With an estimated 40–50 million monthly listeners, the rapid growth of the podcast medium in India is evident. This growth is expected to continue, fueled by increasing internet penetration and smartphone usage across the nation. The linguistic landscape, boasting almost 22 official languages in India, necessitates the evolution of podcast summarization tools to cater to regional languages.

In this paper, we present an investigation into unsupervised techniques for personalized content-based podcast recommendation in a multi-lingual domain, targeting Hindi and English language. Hindi is chosen because 53.6% of the Indian population declared that they speak Hindi as either their first or second language and English due to its global acceptance. Our approach incorporates multi-view representations, specifically utilizing both sentence-level and keyword-level views, to capture the diverse aspects of podcast content. By leveraging these multi-view representations, we aim to overcome the limitations of traditional single-view approaches, which may fail to capture the nuances of multi-lingual content [12]. To extract meaningful representations from the podcast data, we employ auto-encoders [6], a type of neural network architecture renowned for their ability to learn compressed representations of input data. By training auto-encoders on the podcast dataset, we obtain latent representations that capture the semantic relationships between sentences and keywords within the podcast content. After getting the latent representation corresponding to each view, we explore the application of traditional and graph-based clustering algorithms to group similar podcasts together. Thereafter, to obtain the consensus partitioning satisfying both the views, we investigate the use of ensemble learning. As per our knowledge, this is a first of its kind work for personalized recommendation system which is utilizing the efficacy of auto-encoders, clustering algorithms, multi-linguality and ensemble learning in an integrated framework.

Thus, the objective of our study is to develop a robust and effective podcast recommendation system that caters to the diverse language preferences of users.

By incorporating unsupervised techniques and multi-view representations, we aim to address the challenges posed by language diversity and content heterogeneity in podcast recommendation.

For experimental purposes, a dataset comprising textual descriptions of 10,000 podcasts in Hindi and English was considered. As the proposed algorithm is based on multi-view ensemble clustering, therefore, firstly, clustering performance is validated using the silhouette score, and then the top-N recommendations are evaluated using the well-known diversity and coverage score in the field of recommendation. The results are compared with the different variants of the proposed algorithm in terms of the clustering algorithm. To check the efficacy of the proposed multi-view framework, an auto-encoder-based single-view ensemble clustering framework is also developed.

Key-Contributions: The key-contributions of the paper are listed below: (a) We propose a novel approach that utilizes multi-view representations of podcast content, considering both sentence-level and keyword-level views. By incorporating multiple views, we aim to capture the diverse aspects of podcast content and overcome the limitations of single-view approaches; (b) We propose a novel approach that utilizes multi-view representations of podcast content, considering both sentence-level and keyword-level views. By incorporating multiple views, we aim to capture the diverse aspects of podcast content and overcome the limitations of single-view approaches; (c) Our proposed methodology has been successfully applied to a Hindi dataset, demonstrating its adaptability and effectiveness in handling linguistic intricacies and cultural nuances specific to the Hindi language. This extension of our approach to a non-English dataset not only underscores its versatility but also highlights its potential for cross-lingual applications, thereby contributing significantly to the advancement of natural language processing (NLP) techniques in multicultural contexts. Along with, the results also reported on English language due to its global acceptance.

The outline of this paper is structured as follows: Sect. 2 provides an overview of related work in podcast recommendation and multi-view clustering. Section 3 presents the proposed methodology. Section 4 discusses the experimental setup, including dataset description, comparative methods, and evaluation metrics. Section 5 presents and analyzes the experimental results. Finally, Sect. 6 concludes the paper.

2 Related Work

Various recommendation system techniques have been proposed across different domains in the literature. Sharma et al. [17] suggested utilizing the Amazon Apparel dataset by employing a combination of Convolutional Neural Networks (CNN) and Natural Language Processing (NLP) to scrutinize product titles and images, aiming for predictive applications. Bellini et al. [2] introduced a fashion retail recommendation system addressing cold start and seasonal dynamics through multi-clustering. For podcast recommendations, pioneers like [20], used music as a resource, while Benton et al. [4] considered user interaction order and

semantic relationships from podcast knowledge graphs for precise recommendations. In [21], an online clustering method using auto-encoders managed the continuous flow of articles. The authors of Walek et al. [26] presented the Eshop recommender, integrating collaborative filtering and content-based subsystems with a fuzzy expert system. In [29], authors introduced AGRE, leveraging knowledge graphs to capture user preferences. Introducing a novel approach, our research focuses on multi-view ensemble clustering [10], combining outputs from distinct base clustering algorithms for enhanced performance and capturing complex data structures effectively. In contrast to deep multi-view clustering algorithms [10] employing multiple autoencoders, our work emphasizes efficiency and simplicity, utilizing a single autoencoder to integrate different views (sentence and keyword-based) and an ensemble clustering algorithm for robust cluster formation, thereby offering a more streamlined and effective clustering framework. In contrast to existing multi-view clustering approaches, our research introduces multi-view ensemble clustering, a novel approach that combines outputs from two distinct base clustering algorithms to form robust clusters. By leveraging the strengths of both algorithms, our method aims to enhance clustering performance, capture complex data structures effectively, and provide a comprehensive solution to multi-view clustering challenges.

Our work is motivated by several research gaps in the literature. Firstly, there's a lack of attention to podcast recommendation systems, especially concerning multi-view representations for improved accuracy and user satisfaction. Additionally, existing research often overlooks Indian regional languages, which offer unique challenges and opportunities in recommendation systems. While various techniques exist, like CNN and NLP for product recommendation, our approach fills this void by focusing on podcast recommendations with multi-view ensemble clustering. Unlike traditional methods, we integrate two distinct clustering algorithms to create more robust and diverse clusters. This novel approach aims to address the challenges of multi-view clustering, particularly in the context of Indian regional language podcasts.

3 Methodology

This section delineates a comprehensive methodology tailored for podcast recommendation within the nuanced landscape of multi-lingual domains, harnessing the potential of unsupervised techniques like multi-view analysis with autoencoders alongside a spectrum of traditional clustering algorithms. Our methodology unfolds in three primary dimensions: Sentence View, Keyword View and Concatenated Views, each engineered to cater to the diverse needs and preferences of users across linguistic boundaries. The flow chart of the proposed methodology is shown in Fig. 1.

3.1 Input Podcast Representation

In crafting a robust representation framework for podcasts, we adopt a dual-view paradigm comprising the sentence view and the keyword view. The sentence

view encapsulates the rich textual narratives woven within podcast descriptions, while the keyword view distills pivotal thematic elements through sophisticated keyword extraction techniques. This multi-faceted representation strategy allows for a holistic understanding of podcasts, capturing both their semantic essence and thematic underpinnings.

Sentence View: In the Sentence View, Robustly Optimized BERT Approach (RoBERTa) [18] for English dataset and Cross-Lingual Language Model Robustly Optimized BERT Approach (XLM-RoBERTa) [7] for Hindi dataset, are utilized to create Sentence embeddings that capture the semantic essence of podcast descriptions. Thereafter, an auto-encoder is employed to distill meaningful representations from the sentence view, leveraging the innate capacity of neural networks to discern intricate patterns within textual data.

Keyword View: In the Keyword View, keywords extracted using Rapid Automatic Keyword Extraction (RAKE) [22] are transformed into Keyword embeddings using RoBERTa and XLM-RoBERTa, for English and Hindi dataset, respectively, capturing their semantic significance within the podcast landscape.

3.2 Latent Vectors Corresponding to the Input Podcast Representation

Corresponding to the input vector representation for each view, an autoencoder is employed to distill meaningful representations from the sentence view, leveraging the innate capacity of neural networks to discern intricate patterns within textual data. The details about the auroencoder is provided below.

Autoencoder Framework. The autoencoder framework is a powerful unsupervised learning technique used for dimensionality reduction, feature learning, and data denoising. It consists of an encoder, which compresses the input data into a lower-dimensional representation, and a decoder, which reconstructs the original input data from the compressed representation. By minimizing the reconstruction error between the input and the output, the autoencoder learns to capture meaningful features of the input data in an efficient and compact representation.

Encoder: The encoder compresses the input data into a lower-dimensional representation. Let X denote the input data, h_i represent the hidden representation at layer i, W_i be the weight matrix for layer i, b_i be the bias vector for layer i, and Activation be the activation function used in each layer. The forward pass of the encoder can be represented as:

$$\begin{aligned} h_1 &= \text{Activation}(W_1 \cdot X + b_1) \\ h_2 &= \text{Activation}(W_2 \cdot h_1 + b_2) \\ &\cdots \\ h_n &= \text{Activation}(W_n \cdot h_{n-1} + b_n) \end{aligned}$$

Decoder: The decoder reconstructs the original input data from the compressed representation obtained from the encoder. Let Z represent the compressed representation obtained from the encoder, and h_i represent the hidden representation at layer i in the decoder. The forward pass of the decoder can be represented as:

$$\begin{aligned} h_{n+1} &= \text{Activation}(W_{n+1} \cdot Z + b_{n+1}) \\ h_{n+2} &= \text{Activation}(W_{n+2} \cdot h_{n+1} + b_{n+2}) \\ &\cdots \\ h_{2n} &= \text{Activation}(W_{2n} \cdot h_{2n-1} + b_{2n}) \end{aligned}$$

Training Objective (Loss): The autoencoder is trained on the individual view representation. The training objective is to minimize the reconstruction error between the input and the output. Let X be the input data and $\hat{X}$ be the output (reconstruction) of the autoencoder for the input X. The loss function measures the difference between X and $\hat{X}$, commonly chosen as mean squared error (MSE).

$$\text{Loss} = \text{MSE}(X, \hat{X}) \quad \text{and} \quad \text{MSE}(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{1}$$

Here, n represents the number of elements (dimension) in the input vectors, x_i is the i-th element of the actual input vector, and $\hat{x}_i$ is the i-th element of the predicted (reconstructed) vector.

Training Procedure: The autoencoder is trained by minimizing the loss function using an optimization algorithm The parameters weights and biases of the autoencoder are updated iteratively using the gradients of the loss function with respect to the parameters:

$$(W, b) \leftarrow (W, b) - \eta \frac{\partial(\text{Loss})}{\partial(W, b)}$$

In this equation, (W, b) represents the parameters weights and biases of the autoencoder, η is the learning rate, which controls the step size of the parameter updates and $\frac{\partial \text{Loss}}{\partial \theta}$ represents the gradient of the loss function with respect to the parameters.

3.3 Partitioning Similar Podcasts Using Concatenated Views

To form the grouping/partitioning/clusters of similar podcasts based on their intrinsic similarities, several traditional clustering algorithms along with graph-based algorithms (Affinity Propagation [13], KMeans [1], KMedoid [16], Hierarchical Clustering [15], Gaussian Mixture Model (GMM) [8], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [11], Spectral Clustering [24], and Balanced Iterative Reducing and Clustering using Hierarchies

(BIRCH) [28]) are explored to navigate the multi-view feature space. This clustering process forms the bedrock of our recommendation framework, facilitating the extraction of top-tier podcasts aligned with user preferences. To validate the cluster quality, an internal cluster validity index is used which measure compactness and separation among clusters. It should be note that for most of the clustering algorithms, the number of clusters should be known a-priori. Therefore, we have the considered the number of clusters estimated by affinity propagation algorithm.

3.4 Perform Ensemble Clustering

It aims at enhancing podcast recommendation system through the integration of multi-view ensemble clustering techniques. Our objective is to offer personalized recommendations for the top-10 podcasts, leveraging the amalgamation of diverse views and ensemble clustering methodologies. In our architecture (based on the experimental analysis), we leverage two robust base clustering algorithms (a) BIRCH and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [5] for English Dataset; (b) BIRCH and Spectral for Hindi Dataset, for handling extensive datasets and capturing diverse structures inherent in podcast data. These algorithms excel in identifying clusters of varying shapes and densities, as well as modeling complex data structures and detecting clusters of varying densities within concatenated views. By capitalizing on their strengths, we aim to devise a comprehensive clustering solution for effectively capturing diverse patterns in podcast data. This architecture represents a significant advancement over traditional recommendation systems by harnessing multiple perspectives of podcast data and enhancing the reliability of clustering results through ensemble techniques. This architecture represents a significant advancement over traditional recommendation systems by harnessing multiple perspectives of podcast data and enhancing the reliability of clustering results through ensemble techniques.

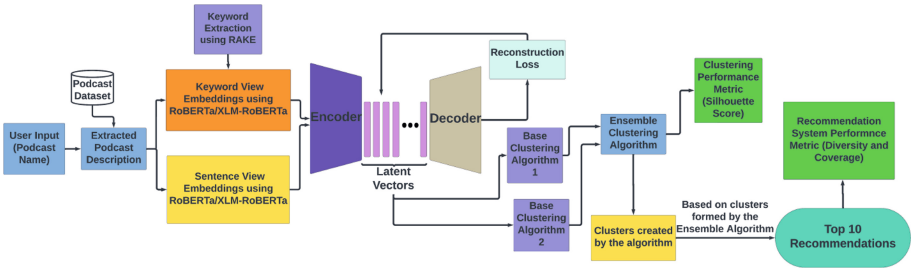


Fig. 1. Flowchart and Architecture of the Multi-view Ensemble Clustering-based Podcast Recommendation

The overall mathematical representation of the clustering process using Ensemble Clustering can be expressed as a combination of similarity computation, aggregation, and optional transformations:

$$C = \frac{1}{n} \sum_{i=1}^{n} S_i \tag{2}$$

where, C is the cluster matrix, n is the number of base learners, S_i is the similarity matrix computed by the ith base learner.

Then, $C = \log(|C_{ij}|+1e^{-8})$, where the logarithmic transformation $\log(|C_{ij}|+1e^{-8})$ is applied to each element C_{ij} of the matrix with a small additive term to prevent numerical instability. The purpose of this transformation is to stabilize the calculation and prevent numerical issues, such as division by zero or taking the logarithm of zero, which can lead to numerical instability or errors in computations. Finally, if model weights are provided, the cluster matrix is element-wise multiplied by these weights as $C = C \times W$, where, W is the matrix of model weights. In ensemble clustering, model weights assign varying significance to each base learner's outcome, either specified manually or learned during the process. When weights aren't explicitly provided, each learner contributes equally, resulting in a democratic fusion of clustering perspectives. This approach ensures all learners are equally valued, potentially resulting in a well-balanced ensemble outcome. These tasks culminate in fitting the ensemble model and furnishing aggregated cluster labels, thereby empowering the clustering process to embrace diverse data perspectives and potentially enhance performance through comprehensive insights extraction.

3.5 Suggesting Recommendations

The recommendation procedure begins by encoding the test data point, which represents the user selected podcast, using the encoder model to obtain its latent vector representation. Next, the cosine distances between the test latent vector and the cluster centers formed by the ensemble clustering algorithm are computed to determine the clusters, aiming to identify the cluster centers with the minimum cosine distances from the test latent vector. Next, podcasts within the same cluster as the test data point are identified based on their cluster labels and added to a set. Within this cluster, pairwise distances between the test data point and each podcast are computed, and the podcasts are sorted based on these distances. Finally, the top 10 podcasts with the closest latent vectors to the test data point are selected as recommendations. This approach leverages clustering to group similar podcasts and focuses on recommending podcasts within the same cluster, ensuring that the recommendations are relevant to the test data point's content and preferences.

4 Experimental Setup

4.1 Dataset

The dataset utilized in this research comprises around 10,000 podcasts obtained from the iTunes platform, now referred to as Apple Podcasts. Each podcast entry is characterized by attributes such as title, brief description, web URL, genre identification (later converted to genre names), episode count, and duration of each episode. Before conducting experimental analyses, rigorous preprocessing procedures were applied to uphold data integrity. These steps involved removing stop words, special characters, and URLs. This meticulous preprocessing ensured that the dataset was cleansed of irrelevant information and prepared for subsequent analyses, enhancing the accuracy and reliability of the research findings. After preprocessing, the dataset was split into training (80%) and testing (20%) sets, comprising 8,124 and 2,031 samples, respectively. The training set facilitated model development, while the testing set served for independent validation, ensuring model performance and generalization capabilities.

4.2 Comparative Methods Used

Our research rigorously scrutinized a range of clustering algorithms within an innovative multiview framework. We meticulously evaluated the effectiveness of various techniques, including Affinity Propagation [13], K-Means [1], K-Medoid [16], Hierarchical [15], Gaussian Mixture Model [8], DBSCAN [11], Spectral Clustering [24], BIRCH [28], and HDBSCAN [5]. Additionally, we pioneer a novel approach by crafting a single-view-based ensemble clustering strategy. These methodologies are applied to two distinct perspectives: the sentence view and the keyword view, allowing for comprehensive analysis and comparison.

4.3 Parameter Settings

The parameter settings for our experiments involve utilizing RoBERTa and XLM-RoBERTa, variants of Bidirectional Encoder Representations from Transformers (BERT) [9] known for its efficacy in natural language processing tasks, for tokenization and word embedding generation for English and Hindi language, respectively. To get the latent representation, standard Autoencoder is used coupled with Mean Squared Error (MSE) as the loss function for training. RMSProp is an optimizer that helps improve the training process of autoencoders by adjusting the learning rates dynamically based on the history of gradients, ultimately leading to more stable and efficient model training. For our purpose, the Root Mean Square Propagation (RMSProp) [25] optimizer is used which handles varying learning rates effectively, ensuring model stability. The learning rate value utilized in the study was the default setting for RMSProp, which is 0.001. Training spans 100 epochs with a batch size of 32 for computational efficiency. These parameters are selected after performing the sensitivity analysis test on different parameters. We employ the cosine distance metric to measure similarity, crucial

for accurate clustering and representation learning. Our experiments were conducted on Google Colaboratory, providing substantial resources, including 12.7 GB RAM, 16 GB GPU memory, and a 78.2 GB total disk size, leveraging the Python 3 Google Compute Engine backend (Nvidia Tesla T4 GPU) for efficient processing.

4.4 Evaluation Measure

In evaluating the performance of our recommendation system, we prioritize the silhouette score [23] as a fundamental metric for assessing clustering quality. This metric offers valuable insights into the effectiveness of clustering algorithms utilized to group similar items within our recommendation system. Additionally, our evaluation framework incorporates a suite of metrics including diversity and coverage, all aimed at providing a comprehensive assessment of recommendation system performance. Diversity [14] measurement evaluates the variety among recommended items, catering to a wide array of user preferences and ensuring a diverse selection of content. Furthermore, our coverage [14] assessment examines the system's capability to recommend items across the entire catalog, guaranteeing comprehensive coverage of available content and enhancing the overall user experience.

Table 1. Comparison of clustering algorithms and proposed ensemble clustering algorithm on Sentence View of English and Hindi Datasets

Algorithm	English Dataset			Hindi Dataset		
	Sil.	Div.	Cov.	Sil.	Div.	Cov.
Affinity Propagation	0.0770	0.0280	0.8120	0.2340	0.8220	0.0010
Kmeans	0.1160	0.0290	0.6910	0.2650	0.0010	0.4340
Kmedoids	0.0660	0.0290	0.6790	0.2170	0.0010	0.4320
DBSCAN	0.0080	0.6270	0.7920	–0.0620	0.5850	0.8060
Gaussian Mixture Model	0.1150	0.0290	0.6860	0.2470	0.0010	0.4410
Spectral	0.5160	0.6270	0.7930	0.4190	0.4620	0.8310
Birch	0.5160	0.6270	0.7930	0.4610	0.4620	0.8320
Hierarchical	0.0480	0.6270	0.8080	0.2150	0.4620	0.8280
HDBSCAN	0.4230	0.6123	0.7607	0.3578	0.4679	0.8354
Ensemble Clustering	**0.5234**	**0.6087**	**0.7600**	**0.3507**	**0.4729**	**0.8335**

Abbreviation: Sil. - Silhouette Score, Div. - Diversity, Cov. - Coverage

5 Discussion of Results

5.1 For English Dataset

The *English* dataset, the results from individual views (sentence view and keyword view), whether examining sentence or keyword perspectives, exhibit varied

performance among traditional clustering algorithms as shown in Tables 1 and 2. While some algorithms show promising results in certain metrics, such as K-means in the keyword view achieving a relatively high silhouette score of 0.1240, none consistently outshine the others across all evaluation criteria. However, across both views, the ensemble clustering algorithm consistently outperforms individual methods, showcasing its robustness and versatility. This comprehensive understanding from the result Tables 1 and 2, reflected in the silhouette score of 0.5234 and 0.4814, diversity of 0.6087 and 0.6042, and coverage of 0.7600 and 0.7809 for sentence and keyword views, respectively, highlights the effectiveness of Ensemble Clustering in leveraging diverse insights for enhanced clustering quality. DBSCAN exhibited the worst performance in the English dataset, particularly in the Concatenated View, with negative silhouette scores and lower diversity and coverage metrics. This suggests challenges in forming well-separated clusters, due to difficulties in handling varying densities and sensitivity to parameter tuning. Furthermore, by combining information from multiple perspectives, the again ensemble clustering approach is better equipped to capture the nuanced structure of the English dataset, resulting in more cohesive and meaningful clusters as can be seen from Table 3. The concatenated view exhibits superior clustering performance in the English dataset compared to both the keyword and sentence views. It achieves higher silhouette scores, indicating better-defined clusters, along with increased diversity and coverage metrics. This suggests that combining information from both views enhances the clustering process, resulting in more comprehensive and effective groupings of data items. The ensemble clustering method's ability to harness complementary information from various views fosters a more holistic representation of the dataset, enabling it to uncover underlying patterns that may remain elusive to individual algorithms alone.

Table 2. Comparison of clustering algorithms and proposed ensemble clustering algorithm on Keyword View of English and Hindi Datasets

Algorithm	English Dataset			Hindi Dataset		
	Sil.	Div.	Cov.	Sil.	Div.	Cov.
Affinity Propagation	0.0800	0.0220	0.8090	0.2510	0.0008	0.8260
Kmeans	0.1240	0.0240	0.6640	0.3190	0.0010	0.4090
Kmedoids	0.0760	0.0240	0.6650	0.2740	0.0010	0.4070
DBSCAN	–0.0660	0.5930	0.8080	–0.0370	0.5910	0.8010
Gaussian Mixture Model	0.1240	0.0250	0.6640	0.2850	0.0010	0.4170
Spectral	0.4670	0.6250	0.8030	0.4110	0.4680	0.8380
Birch	0.1080	0.5930	0.8090	0.3480	0.4670	0.8390
Hierarchical	0.0520	0.5930	0.8110	0.2740	0.4660	0.8360
HDBSCAN	0.1240	0.5802	0.8270	0.3672	0.4734	0.8360
Ensemble Clustering	**0.4814**	**0.6042**	**0.7809**	**0.4021**	**0.4683**	**0.8401**

Abbreviation: Sil. - Silhouette Score, Div. - Diversity, Cov. - Coverage

5.2 For Hindi Dataset

Turning to the Hindi dataset, a similar pattern emerges where traditional clustering algorithms exhibit varied performance across different views. While certain algorithms may excel in specific metrics, such as *spectral* clustering achieving a high silhouette score of 0.4670 in the sentence view, none consistently outperform others across all evaluation criteria. However, the ensemble clustering algorithm consistently demonstrates superior performance, surpassing individual methods in both sentence and keyword views as can be seen from Tables 1 and 2. By integrating insights from multiple perspectives, ensemble clustering effectively captures the intricate structure of the Hindi dataset, leading to more accurate and cohesive clusters (see Table 3). The ensemble clustering approach view achieves notable results with a silhouette score of 0.3507 and 0.4021, Diversity of 0.4729 and 0.4683, and Coverage of 0.8335 and 0.8401 for sentence and keyword views, respectively, underscoring its efficacy in leveraging diverse viewpoints for enhanced clustering outcomes. DBSCAN exhibited the poorest performance across all views. With negative silhouette scores and lower diversity and coverage metrics compared to other algorithms, DBSCAN's sensitivity to parameter settings and its density-based approach likely contributed to its suboptimal clustering results. These findings underscore the robustness and adaptability of ensemble clustering in addressing the complexities of the Hindi dataset and generating meaningful cluster representations. Additionally, by incorporating insights from multiple views, the ensemble Clustering algorithm provides a comprehensive understanding of the dataset's underlying structure, facilitating more informed decision-making and analysis. Moreover, the multi-view ensemble clustering approach consistently outperforms individual views in terms of silhouette score, diversity, and coverage for the Hindi dataset. It achieves higher silhouette scores, indicating better cluster cohesion and separation. Additionally, the ensemble method captures a broader range of topics, leading to improved diversity. Moreover, it provides recommendations that cover a wider scope of content, enhancing overall coverage compared to individual views.

5.3 Podcast Recommendation Analysis

As per the analysis done, the recommendations for the English Dataset align well with the user's input. The generated top-10 recommendations for "Straight Talk with Ross Mathews" are shown in Fig. 2. They cover diverse topics including martial arts, leadership, personal development, health, finance, and entrepreneurship, resonating with the engaging and informative nature of "Straight Talk with Ross Mathews." Podcasts like "Mind Body Musings" and "Trim Healthy Podcast" reflect themes discussed on Ross Mathews' show, such as self-improvement and relationships. Overall, the recommendations effectively suggest relevant and engaging content.

The recommendations for the "सीबीएस स्पोर्ट्स आई ऑन कॉलेज बास्केटबॉल पॉडकास्ट " in the Hindi dataset encompass diverse genres and topics as shown in Fig. 3. While some align closely with sports, others delve into literature, biography, and

Table 3. Comparison of clustering algorithms and proposed ensemble clustering algorithm on Concatenated Views of English and Hindi Datasets

Algorithm	English Dataset			Hindi Dataset		
	Sil.	Div.	Cov.	Sil.	Div.	Cov.
Affinity Propagation	0.0940	0.0190	0.8110	0.3320	0.0002	0.8310
Kmeans	0.0942	0.0760	0.7210	0.3670	0.0004	0.3600
Kmedoids	0.0840	0.0210	0.6690	0.3240	0.0005	0.3680
DBSCAN	−0.1570	0.6240	0.7980	−0.0550	0.5790	0.8190
Gaussian Mixture Model	0.1280	0.0210	0.6750	0.3290	0.0005	0.3720
Spectral	0.5120	0.6230	0.8000	0.5550	0.4540	0.8410
Birch	0.1430	0.5910	0.8130	0.5590	0.4560	0.8420
Hierarchical	0.0620	0.6230	0.8140	0.3110	0.4540	0.8370
HDBSCAN	0.6670	0.6024	0.7744	0.3516	0.4580	0.8360
Ensemble Clustering	**0.7664**	**0.6024**	**0.7742**	**0.4859**	**0.4616**	**0.8382**

Abbreviation: Sil. - Silhouette Score, Div. - Diversity, Cov. - Coverage

personal development. Comparing these recommendations to user input reveals similarities in diversity and range, offering a broad spectrum of content. Specific matches, like " ट्रेलसाइड रेडियो " for sports-related content, " जीवनी " for storytelling, and " पारिवारिक दैनिक प्रसारण पर ध्यान दें " for personal growth, indicate alignment with the target podcast's themes or audience preferences (Fig. 3).

The diverse nature of these recommendations suggests that the recommendation system considers not only direct thematic similarities but also broader content appeal and potential audience interests. This approach aims to cater to a wide range of listener preferences.

5.4 Limitations of the Approach

We have listed out the limitations of our approach:
(a) Dependency on Podcast Descriptions: Our model relies heavily on textual descriptions to capture podcast content and themes. This dependence may lead to reduced accuracy when descriptions are sparse, incomplete, or lack detailed information about the podcast's content. In such cases, the model's ability to accurately recommend podcasts may be compromised, as it may not fully capture the nuances and depth of podcast episodes.
(b) Language and Cultural Bias: While our model is designed to be adaptable to multiple languages, variations in language and cultural contexts can introduce biases that affect recommendation relevance. Cultural nuances in podcast content and listener preferences may not always be fully captured or appropriately weighted, potentially leading to less relevant recommendations for users from different cultural backgrounds.
(c) Incremental Learning for New Podcasts: Currently, the system requires

- TWIMA: This Week In Martial Arts
- 1UP.com - The Oddcast
- Mind Body Musings Podcast: Feminine Embodiment | Surrender & Trust | Relationships| Limiting Beliefs | Radical Responsibility
- Dov Baron: Leadership and Loyalty Show for Fortune 500 Executives, Family Businesses, Leadership Speaker-Consultant, Business
- Cell Podcast
- DCPL Podcast
- Trim Healthy Podcast w/Serene and Pearl
- The Commercial Investing Show
- Queen Rania's podcast channel
- Breaking Into Startups

Fig. 2. Top 10 podcast recommenndations for "Straight Talk with Ross Mathews"

- प्राइड एंड प्रीजूडिस
- ट्रेलसाइड रेडियो
- ब्लैक एजेंडा रेडियो
- खंडित – एंड्रॉइड डेवलपर पॉडकास्ट
- जीवनी
- पारिवारिक दैनिक प्रसारण पर ध्यान दें
- अल्ट्रा का विज्ञान | अल्ट्रा मैराथन और ट्रेल रनिंग विशेषज्ञता | विश्व की अग्रणी सहनशक्ति विज्ञान और कोचिंग
- अवसाद की प्रफुल्लित करने वाली दुनिया
- 1 से 70 तक
- टोरा का अर्थ है शिक्षक: बाइबिल की पहली पांच पुस्तकों से सबक: डॉ. नहूम रोमन फ्रुटनिक डेनिस प्रेगर से प्रेरित

Fig. 3. Top 10 podcast recommendations for "सीबीएस स्पोर्ट्स आई ऑन कॉलेज बास्केटबॉल पॉडकास्ट"

retraining to effectively integrate new podcasts and episodes. Implementing incremental learning mechanisms, such as online clustering algorithms or model updating techniques, would enable the system to evolve continuously with new data. This approach ensures that recommendations remain relevant and up-to-date without needing a complete retraining process. Incremental learning allows the model to adapt dynamically to new content, improving efficiency and scalability. By continually refining the system, we can maintain high-quality recommendations even as the podcast library expands.

(d) Evaluation Scope: While metrics like silhouette score, diversity, and coverage provide valuable insights into clustering performance and recommendation coverage, they may not directly reflect user satisfaction or engagement. Additional user-centric metrics, such as user feedback, interaction rates with recommended podcasts, and long-term user retention, would provide a more comprehensive evaluation of the system's effectiveness in meeting user needs and preferences.

6 Conclusion

In this research, we introduced a multi-view ensemble clustering approach to improve clustering and recommendation accuracy for podcast datasets. To show that out algorithm incorporate multi-linguasim, we have used two datasets: one in Indian language, i.e., Hindi and another in English language. Based on the

results analysis, we found that multi-view based ensemble algorithm outperforms single-view based ensemble algorithms, achieving notable results. For the English dataset, it attains a silhouette score, diversity and coverage of 0.7664, 0.6024, and 0.7742, respectively. Similarly, for the Hindi dataset, it achieves a silhouette score, diversity and coverage of 0.4859, 0.4616, and 0.8382, respectively. In terms of algorithms used for ensemble, by combining HDBSCAN and BIRCH for English dataset, and Spectral and BIRCH for Hindi dataset, our method effectively improves clustering quality and recommendation diversity. These findings mark a significant contribution to advancing ensemble clustering and recommendation systems. Furthermore, they pave the way for future improvements in clustering and recommendation strategies tailored for podcast platforms. Overall, our research highlights the potential of ensemble techniques in enhancing clustering performance and recommendation accuracy.

References

1. Ball, G.H., Hall, D.J.: Isodata, a novel method of data analysis and pattern classification (1965). https://api.semanticscholar.org/CorpusID:53887616
2. Bellini, P., Palesi, L.A.I., Nesi, P., Pantaleo, G.: Multi clustering recommendation system for fashion retail. Multimedia Tools Appl. **82**(7), 9989–10016 (2023). https://doi.org/10.1007/s11042-021-11837-5
3. Benton, G., Fazelnia, G., Wang, A., Carterette, B.: Trajectory based podcast recommendation. arXiv preprint arXiv:2009.03859 (2020)
4. Benton, G., Fazelnia, G., Wang, A., Carterette, B.: Trajectory based podcast recommendation (2020)
5. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 160–172. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_14
6. Chen, S., Guo, W.: Auto-encoders in deep learning–a review with new perspectives. Mathematics **11**(8), 1777 (2023)
7. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Association for Computational Linguistics, Online (2020).https://doi.org/10.18653/v1/2020.acl-main.747. https://aclanthology.org/2020.acl-main.747
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Royal Stat. Soc. Ser. B (Methodological) **39**(1), 1–38 (1977)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019).https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423
10. Du, G., Zhou, L., Yang, Y., Lü, K., Wang, L.: Deep multiple auto-encoder-based multi-view clustering. Data Sci. Eng. **6**(3), 323–338 (2021). https://doi.org/10.1007/s41019-021-00159-z

11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD 1996, pp. 226–231. AAAI Press (1996)
12. Fang, U., Li, M., Li, J., Gao, L., Jia, T., Zhang, Y.: A comprehensive survey on multi-view clustering. IEEE Trans. Knowl. Data Eng. **35**, 12350–12368 (2023)
13. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007). https://doi.org/10.1126/science.1136800. https://www.science.org/doi/abs/10.1126/science.1136800
14. Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Trans. Interact. Intell. Syst. **7**(1) (2016). https://doi.org/10.1145/2926720
15. Kaufman, L., Rousseeuw, P.: Finding groups in data: an introduction to cluster analysis (1990). https://doi.org/10.2307/2532178
16. Kaufmann, L., Rousseeuw, P.: Clustering by means of medoids. In: Data Analysis Based on the L1-Norm and Related Methods, pp. 405–416 (1987)
17. Kumar Sharma, A., Bajpai, B., Adhvaryu, R., Dhruvi Pankajkumar, S., Parthkumar Gordhanbhai, P., Kumar, A.: An efficient approach of product recommendation system using nlp technique. Mater. Today: Proc. **80**, 3730–3743 (2023). https://doi.org/10.1016/j.matpr.2021.07.371. https://www.sciencedirect.com/science/article/pii/S2214785321052305
18. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach (2019)
19. MacLennan, A.F., Biswas, M.: Symposium introduction: multilingual, multicultural, migrant, and diasporic radio, audio, and podcasting (2023)
20. Nazari, Z., et al.: Recommending podcasts for cold-start users based on music listening and taste. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, pp. 1041–1050. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3397271.3401101
21. Paul, D., Chakdar, D., Saha, S., Mathew, J.: Multiview deep online clustering: an application to online research topic modeling and recommendations. IEEE Trans. Comput. Social Syst. **10**(5), 2566–2578 (2023). https://doi.org/10.1109/TCSS.2022.3187342
22. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents, pp. 1 – 20 (2010). https://doi.org/10.1002/9780470689646.ch1
23. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987). https://doi.org/10.1016/0377-0427(87)90125-7. https://www.sciencedirect.com/science/article/pii/0377042787901257
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000). https://doi.org/10.1109/34.868688
25. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. **4**, 26–31 (2012)
26. Walek, B., Fajmon, P.: A hybrid recommender system for an online store using a fuzzy expert system. Expert Syst. Appl. **212**, 118565 (2023). https://doi.org/10.1016/j.eswa.2022.118565. https://www.sciencedirect.com/science/article/pii/S0957417422016293

27. Yang, L., Sobolev, M., Tsangouri, C., Estrin, D.: Understanding user interactions with podcast recommendations delivered via voice. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 190–194 (2018)
28. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. SIGMOD Rec. **25**(2), 103–114 (1996). https://doi.org/10.1145/235968.233324
29. Zhao, N., Long, Z., Wang, J., Zhao, Z.D.: AGRE: a knowledge graph recommendation algorithm based on multiple paths embeddings rnn encoder. Knowl.-Based Syst. **259**, 110078 (2023). https://doi.org/10.1016/j.knosys.2022.110078. https://www.sciencedirect.com/science/article/pii/S0950705122011741

Privacy-Preserving Ensemble Learning Using Fully Homomorphic Encryption

Tilak Sharma[1(✉)], Nalini Ratha[1], and Charanjit Jutla[2]

[1] State University of New York, Buffalo, USA
{tilaksha,nratha}@buffalo.edu
[2] IBM Research, New York, NY, USA
csjutla@us.ibm.com

Abstract. Deep learning classifiers have reached good accuracy often surpassing the conventional classifiers. To provide the robustness needed in real-world applications, classifier fusion has shown potential. Such fusion methods can involve integration at the feature (embedding) level, classifier score/confidence level, or decision level. In this paper, we explore the enhancement of data privacy in ensemble learning through the integration of Fully Homomorphic Encryption (FHE). Recognizing the potential of ensemble methods to boost performance robustly against data variations, we confront the critical challenge of adversarial attacks that could compromise classifier integrity. To this end, we introduce the Privacy-Preserving Quantile Power Transform Classifier (PPQPTC), an innovative algorithm that applies quantile transformation for score distribution adjustment and power transformation to augment linear classification, all within the FHE domain. The PPQPTC algorithm is uniquely designed to securely process data while encrypted, addressing the urgent need for stringent data privacy and security in sensitive applications. We rigorously evaluate the performance of our algorithm across a range of diverse datasets, including healthcare data and the NIST BSSR-1 dataset for biometric fusion. Our findings reveal that the PPQPTC algorithm not only effectively handles imbalanced datasets but also demonstrates the feasibility and adaptability of conducting secure data processing in encrypted domains.

Keywords: Machine Learning · Ensemble Learning · Privacy · Fully Homomorphic Encryption. · Biometrics · Classification · Deep Learning · CNN · Medical

1 Introduction

Integrating ensemble learning methods in fields that demand high precision in classification and analysis is increasingly important. Sectors like medical diagnostics, financial forecasting, and security systems benefit greatly from the collective strength of multiple classifiers. This synergy enables more reliable and comprehensive outcomes than what could be achieved with individual classifiers. [1] However, integrating data from multiple sources, as essential as it is for enhancing classification accuracy, brings significant challenges, particularly in data privacy and security. In an age where digital data

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 201–214, 2025.
https://doi.org/10.1007/978-3-031-78107-0_13

processing is ubiquitous, protecting sensitive information from various attacks, ranging from data breaches to sophisticated manipulations of classifier outputs, is paramount. Deep learning models have notably outperformed conventional classifiers in various domains due to their ability to learn complex representations. However, they often lack robustness, which is a concern that data fusion can address. By integrating outputs from multiple models, fusion methods can mitigate the weaknesses of individual models, such as susceptibility to adversarial attacks or overfitting.

Fusion methods within the ensemble learning framework [2] include simple techniques like majority voting, where the most common prediction is chosen, to more sophisticated approaches like stacking, where the outputs of various models serve as inputs to a final meta-model. Other techniques involve weighted averaging, where models are weighted based on their performance, and feature-level fusion, which combines the data before the learning process to create a richer feature set. Data fusion is essential in applications where decision-making relies on diverse and rich datasets. By consolidating information from various sources, ensemble learning methods ensure that predictions are both accurate and robust, effectively leveraging the strengths of different classifiers to compensate for their limitations.

Contributions

1. We have developed the Quantile Binning Operator/Algorithm within the encrypted domain, enabling secure and efficient data processing in Fully Homomorphic Encryption (FHE).
2. Our experiments demonstrate the efficient usage of parameters for the Quantile Power Transform Classifier (QPTC) within the FHE framework, optimizing computational performance.
3. We have successfully integrated QPTC within the FHE framework. The QPTC is effective in balancing score distributions through quantile transforms and enhancing linear classification with power transforms. This integration introduces significant challenges due to the computational intensity and complexity of FHE operations, which we address in our paper.

1.1 Threat Model

In the traditional browser security model utilized in cloud computing 1, there are several inherent threats, particularly when data must be decrypted for processing. This scenario poses risks such as data breaches, where decrypted data becomes vulnerable to unauthorized access or theft, potentially leading to exposure of sensitive information [3]. Insider threats also emerge as employees with access to the decrypted data might inadvertently or maliciously misuse it [4]. The system's configuration and authentication mechanisms, if weak, can further exacerbate these vulnerabilities, leading to security breaches. Additionally, users often face limited control and visibility over the cloud infrastructure, which complicates the effective management and response to security incidents. Challenges in maintaining service resilience and availability during system failures or attacks also pose significant risks. Finally, a heavy dependency on cloud service providers for security measures can create gaps, particularly in areas outside the

providers' direct control, such as client-side operations. Addressing these vulnerabilities is crucial and requires robust and comprehensive security measures that cover both the service provider's and the client's ends (Fig. 1).

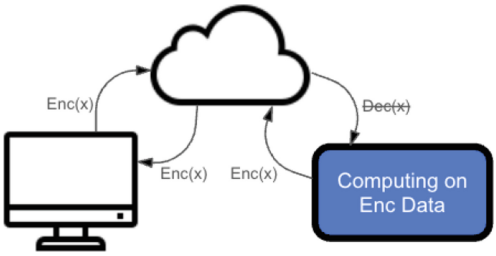


Fig. 1. Security model: In a conventional browser model, the data is encrypted only in the transmission stage and any computation on it would involve decryption. Using FHE, we can perform computations on encrypted data without the need to decrypt it.

In classification problems, various attacks can threaten the integrity of the system. Sensor-level attacks involve presenting false data, and fault injection introduces system errors. Neural networks face adversarial attacks that manipulate input data to cause misclassification and model extraction risks where the network's details are decoded [5]. Matching algorithms may be tampered with, leading to incorrect predictions. At the database level, stored data can be altered or intercepted, compromising data security. Finally, at the decision-making stage, the system can be vulnerable to identity theft, data manipulation, and synthetic data attacks, all of which aim to exploit system vulnerabilities. These threats necessitate robust security measures across all stages of the machine-learning pipeline.

1.2 Encryption and FHE Schemes

Integrating the Fully Homomorphic Encryption (FHE) framework is a critical step forward in addressing security threats. FHE represents a revolutionary advancement in the field of cryptography, allowing for executing computations on encrypted data without the need to decrypt it first. This breakthrough ensures that data remains secure throughout the processing phase. What sets FHE apart from other forms of Homomorphic Encryption, like Partial and Somewhat Homomorphic Encryption, is its ability to perform unlimited operations on encrypted data.

Encryption is crucial for protecting sensitive data, and in this paper, we explore the Fully Homomorphic Encryption (FHE) scheme, particularly the CKKS [6] scheme, which facilitates computation on encrypted complex number vectors without decryption. FHE differs from other forms of homomorphic encryption by allowing unlimited computation depths. The CKKS scheme, preferred over other FHE schemes for its floating-point operation capabilities, involves three keys: a secret key sk, a public key pk, and an evaluation key evk. Key generation, encryption, and decryption are standard

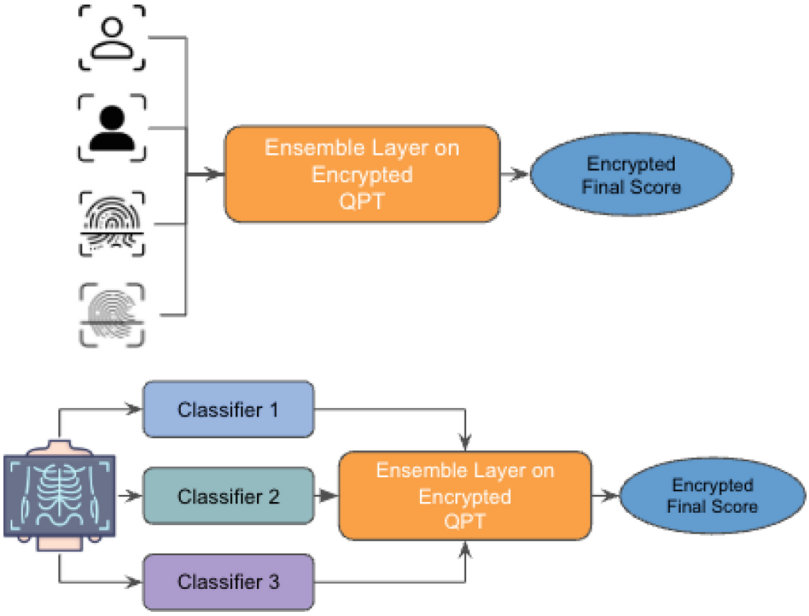


Fig. 2. Illustrative Examples of Application Domains: (a) shows the integration of biometric system scores into the Ensemble Layer in the NIST BSSR dataset. (b) presents the scoring of x-ray images by multiple classifiers in medMNIST, also combined into the Ensemble Layer.

steps in the scheme, with addition and multiplication as basic operations. The challenge of noise accumulation in FHE is addressed by a 'bootstrap' process, albeit at a high computational cost.

The effectiveness and versatility of our proposed Quantile Power Transform Classifier (QPTC) within the Fully Homomorphic Encryption (FHE) framework are demonstrated through various application domains. As illustrated in Fig. 2, our approach integrates into different biometric and medical systems.

This paper aims to tackle these challenges and demonstrate how the QPTC can be effectively implemented within the FHE domain. Doing so combines the advantages of ensemble learning in terms of enhanced classification accuracy with the crucial need for data security in digital processing environments.

2 Prior Work

[7] offered a foundational overview of ensemble learning, emphasizing the role of baseline classifiers in making collective decisions. [8] explored strategies for generating diversity among base classifiers, distinguishing between homogeneous and heterogeneous ensembles. In the realm of voting methods within ensembles, Kim et al. (2003) introduced the concept of max voting. [9,10] delved into the practicalities and limitations of max and soft voting. The effectiveness of averaging voting was discussed by

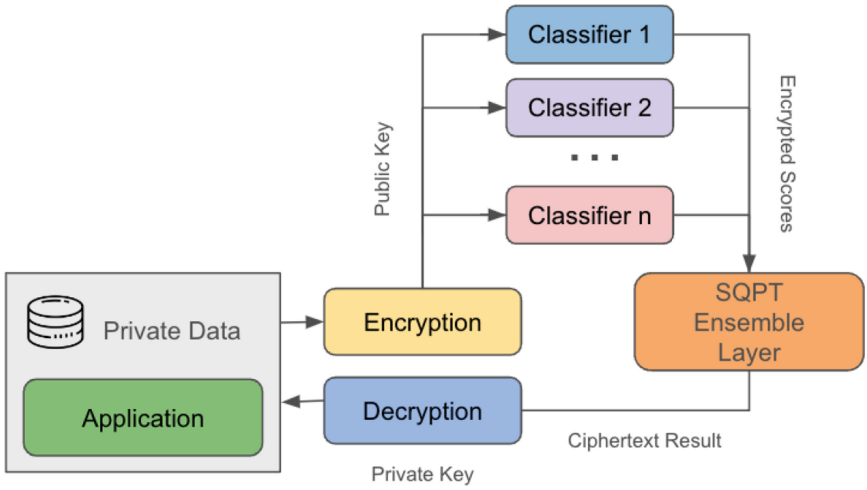


Fig. 3. Overview: The figure illustrates an encrypted ensemble learning system where private data is encrypted before being fed into a series of classifiers; the classifiers' encrypted scores are then combined by an SQPT Ensemble Layer, and the final encrypted result is decrypted for application use, all underpinned by a public-private key mechanism.

[11], while [12,13], and [14] explored the weighted average voting, highlighting its accuracy and the complexities in implementation.

The concept of meta-learning, also known as "learning to learn", has gained significant traction in ensemble learning and machine learning. As [15] described, meta-learning involves learning from previous experiences with different tasks. This method stands apart from traditional machine learning models by incorporating multiple learning stages, where the outputs of individual inducers are used as inputs for a meta-learner that generates the final output. This approach was further elucidated by [16], highlighting its increasing relevance, especially post-2017, due to the complexities of training advanced machine learning algorithms.

Meta-learning addresses several challenges in machine learning, such as operational costs and the lengthy process of finding the best model for a given dataset. By improving learning algorithms and identifying those that perform better, meta-learning can expedite learning processes, enhance adaptability to changing conditions, and optimize hyperparameters for optimal results. [17] noted that it also solves deep learning challenges, including data size, computational complexities, and generalization issues. However, implementing meta-learning is challenging, as [18] outlined. These include defining an appropriate meta-learning approach and managing the computational time complexity, which can be significant with large datasets, multiple baseline models, or multiple levels of meta-learning.

One of the most common meta-learning methods is stacking, as identified by [19], where different models are combined so that their predictions are used as inputs to a final model that makes the ultimate prediction. This method showcases the potential of meta-learning in creating sophisticated, high-performing ensemble systems. Collec-

tively, these studies provide a comprehensive understanding of the various facets of ensemble learning, from the basics of classifier aggregation to the intricacies of voting methods, contributing significantly to the field's evolution.

In [20], privacy is preserved by employing decentralized training, using unlabeled public data for one-way offline knowledge distillation, and implementing ensemble attention distillation. This approach ensures that sensitive data remains within its original location, leveraging public data and distillation techniques for model training without direct exposure to private data. Our work uses ensemble stacking and enhances privacy further by incorporating Fully Homomorphic Encryption (FHE). This method provides robust privacy protection, making it especially suitable for sensitive data applications.

3 Method

We begin by applying the simple but quite effective sum rule as a straightforward arithmetic ensemble approach to establish a baseline for comparison between classifiers. The simplicity of the sum rule allows it to be directly implemented in Fully Homomorphic Encryption (FHE), resulting in no discrepancy between its encrypted and unencrypted forms.

We then advance our model by incorporating quantile-power transformed scores for training, significantly refining the system. This refinement results in an enhanced model that surpasses the elementary sum rule method in performance. Figure 3 provides an overview of our encrypted ensemble learning system, where private data is encrypted, processed by a series of classifiers, and an SQPT Ensemble Layer combines the classifiers' encrypted scores. The final encrypted result is then decrypted for application use, ensuring data security throughout the process.

The encrypted ensemble learning system operates by first encrypting private data to maintain confidentiality. Each classifier in the series operates on the encrypted data, generating encrypted scores that are subsequently combined using the SQPT Ensemble Layer. This layer efficiently aggregates the scores, enhancing the overall accuracy and robustness of the system. Finally, the combined encrypted result is decrypted, making the final output available for secure application use.

3.1 Data Transformation

Leveraging the framework from [21] our approach implements the quantile power transform in the Fully Homomorphic Encryption (FHE) domain. The transformation of score distributions for unencrypted data proceeds as follows:

Let x be a score from classifier i. Quantiles for the score distributions of each class are computed separately, denoted as $[c1_1, c1_2, \ldots, c1_q]$ for class 1, with q representing the number of quantiles, and similarly $[c2_1, c2_2, \ldots, c2_q]$ up to $[cn_1, cn_2, \ldots, cn_q]$ for N classifiers. The transformation of scores using these quantiles is performed by binning:

$$k(x)_{c1\_class1} \text{ where } c1_k \leq x < c1_{k+1}.$$

If $x \geq c1_q$ then $k(x)_{c1} = q+1$. This process is repeated for each classifier, summing the resulting $k(x)_{c1\_class\_i}$ across all classes to compute $k(x)_{c1}$:

$$k(x)_{c1} = \sum_{i=1}^{Cn} k(x)_{c1\_class\_i}, \tag{1}$$

where Cn is the number of categories/classes, and for a binary classifier, n is 2. The normalization of $k(x)$ is given by:

$$k(x) = \frac{k(x)}{2q+2}, \tag{2}$$

since the maximum value of k is $2q + 2$. This normalization is applied to the transformed scores of the training data, preserving the quantile information for all classifiers. Test data is subsequently transformed using this quantile information derived from the training data.

Finally, to further separate classes, the scores are raised to a power $p > 1$, as in the power transform:

$$K(x) = k(x)^p. \tag{3}$$

3.2 Quantile Binning in Encrypted Domain

One of the first steps in QPLC approach is to compute the quantiles for the scores. This requires binning the scores. The Quantile Binning of Encrypted Scores algorithm 1 facilitates the categorization of scores into predefined quantile bins in an encrypted domain. Initially, the algorithm takes scores and quantiles as input and applies the *compB* 0 comparison function to compute the relationship of each score to the quantiles. It then calculates peak values by multiplying each comparison value with its complementary value, which assists in determining the most appropriate bin for each score. These peak values are normalized to maintain consistency in scale. Then we apply the *maxIdx* 3 function, which assigns each score to the relevant quantile based on the normalized peak values. The final output consists of scores categorized into their respective bins, effectively addressing the comparison limitations in Fully Homomorphic Encryption (FHE).

This series of transformations – quantile binning, normalization, and power transformation – is designed for compatibility with FHE operations, ensuring data privacy while aiming for high classification accuracy.

3.3 Polynomail-Based Approximations in FHE

In the encrypted domain, direct binning is not feasible due to the inherent limitations of Fully Homomorphic Encryption (FHE) in performing comparison operations. Given that FHE primarily enables homomorphic addition and multiplication, the need arises to approximate comparison operations using polynomial functions. For this purpose, a comparison function, denoted as $\text{comp}(a, b)$, is defined to yield 0 for $a < b$, 1 for

Algorithm 1. Quantile Binning of Encrypted Scores

Input: Scores $(s_1, s_2, \ldots, s_n)$ and quantiles $(q_1, q_2, \ldots, q_m)$
Output: Score bins $(sb_1, sb_2, \ldots, sb_n)$ where b ranges from 1 to m
$prods \leftarrow []$
for $i \leftarrow 1$ **to** m **do**
    $comp_i \leftarrow \text{COMPB}(scores, q_i, n, d, d_g, d_f)$ 0
    $one\_minus\_comp_i \leftarrow 1 - comp_i$
    $prod_i \leftarrow comp_i \times one\_minus\_comp_i$
    $\text{APPEND}(prods, prod_i)$
end for
NORMALIZE(prods)
$score\_bins \leftarrow \text{MAXIDX}(prods)$ 3
return $score\_bins$

$a > b$, and 0.5 otherwise, employing polynomial approximations to estimate such non-linear functions. Cheon et al. [6] introduced an innovative method that approximates the sign function using a composite polynomial, a technique that we incorporate in our methodology for the quantile binning process within the FHE framework. This approach is integral for transforming scores and securely implementing the Quantile Power Transform Classifier (QPTC) in an encrypted state.

To this framework, the selection of parameters is critical, especially in polynomial-based approximation algorithms. The goal is to optimize this process without reaching the bootstrap threshold. While FHE theoretically allows for an unlimited number of operations, practical constraints often render frequent bootstrapping unfeasible due to its high computational cost. In our experimentation, we work within a feasible range of parameter values, carefully selecting a set that meets our specific requirements. This careful parameter selection is pivotal to achieving an efficient balance between computational feasibility and the desired level of security and accuracy in the encrypted domain (Figs. 4, 5).

Model	Accuracy	TP@0.5	TN@0.5	FP@0.5	FN@0.5	ROC AUC
auto-sklearn	0.8589	386	150	84	4	0.94220
resnet18	0.8333	389	131	103	1	0.94815
resnet50	0.8573	386	149	85	4	0.94932
Sum Rule	0.8525	386	146	88	4	0.94700
Feed-forward neural net	0.8558	386	148	86	4	0.94500
QPNN (10 quantiles)	**0.8365**	**302**	**220**	**14**	**88**	**0.95153**
QPNN (30 quantiles)	**0.8830**	**336**	**215**	**19**	**54**	**0.95320**
QPNN (70 quantiles)	**0.8910**	**344**	**212**	**22**	**46**	**0.95333**

Fig. 4. PneumoniaMNIST on Test data

Algorithm 2. Algorithm compB [6] and its associated polynomials g_n f_n

function COMPB(a, b, n, d_g, d_f)
  $x \leftarrow a - b$
  for $i \leftarrow 1$ to d_g **do**
    $x \leftarrow g(n, x)$
  end for
  for $i \leftarrow 1$ to d_f **do**
    $x \leftarrow f(n, x)$
  end for
  return $(x + 1)/2$
end function
$f_1(x) = -\frac{1}{2}x^3 + \frac{3}{2}x$
$f_2(x) = \frac{3}{8}x^5 - \frac{10}{8}x^3 + \frac{15}{8}x$
$f_3(x) = -\frac{5}{16}x^7 + \frac{21}{16}x^5 - \frac{35}{16}x^3 + \frac{35}{16}x$
$f_4(x) = \frac{35}{128}x^9 - \frac{180}{128}x^7 + \frac{378}{128}x^5 - \frac{420}{128}x^3 + \frac{315}{128}x$
$g_1(x) = -\frac{1359}{2^{10}}x^3 + \frac{2126}{2^{10}}x$
$g_2(x) = \frac{3796}{2^{10}}x^5 - \frac{6108}{2^{10}}x^3 + \frac{3334}{2^{10}}x$
$g_3(x) = -\frac{12860}{2^{10}}x^7 + \frac{25614}{2^{10}}x^5 - \frac{16577}{2^{10}}x^3 + \frac{4589}{2^{10}}x$
$g_4(x) = \frac{46623}{2^{10}}x^9 - \frac{113492}{2^{10}}x^7 + \frac{97015}{2^{10}}x^5 - \frac{34974}{2^{10}}x^3 + \frac{5850}{2^{10}}x$

4 Experiments

4.1 Datasets

Our experiments were conducted using two distinct datasets, each with unique characteristics and applications:

NIST BSSR (Biometric Fusion): This dataset comprises biometric scores from two facial recognition systems and left and right index biometric modalities for 517 users. It includes 517 genuine cases and a significantly larger number of imposter cases (517 $\times$ 516), highlighting the dataset's imbalance.

PneumoniaMNIST [23] **:** A binary classification dataset involving pneumonia scores from five classifiers: autosklearn, resnet18, resnet50, automl-vision, and auto-keras. For our experiments, we selected three weak classifiers (autosklearn, resnet18, and resnet50) to demonstrate our method's ability to enhance accuracy. The dataset consists of 5,856 samples, divided into 4,708 training, 524 validation, and 624 test samples, with 3,494 positive and 1,214 negative samples.

The first dataset, NIST BSSR, is utilized in biometric identification problems, while the second dataset, PneumoniaMNIST, focuses on the classification of pneumonia from various classifiers. Both datasets provide a comprehensive platform to test the effectiveness of our proposed method in different scenarios.

4.2 Setup

For each classifier/modality, we computed two sets of quantiles corresponding to the binary nature of the classification task. These quantiles, q_{class1} and q_{class2}, were derived

Algorithm 3. MaxIdx Algorithm [22]

Input: n distinct numbers $(a_1, a_2, \ldots, a_n)$ with $a_i \in [\frac{1}{2}, \frac{3}{2}]$, $d, d', m, t \in \mathbb{N}$
Output: $(b_1, b_2, \ldots, b_n)$ where b_i is close to 1 if a_i is the largest among a_j's and is close to 0 otherwise.
$inv \leftarrow \text{Inv}(\sum_{j=1}^{n} a_j/n; d')$
for $j \leftarrow 1$ to $n-1$ **do**
    $b_j \leftarrow a_j/n \cdot inv$
end for
$b_n \leftarrow 1 - \sum_{k=1}^{n-1} b_j$
for $i \leftarrow 1$ to t **do**
    $inv \leftarrow \text{Inv}(\sum_{j=1}^{n} b_j^m; d)$
    for $j \leftarrow 1$ to $n-1$ **do**
        $b_j \leftarrow b_j^m \cdot inv$
    end for
    $b_n \leftarrow 1 - \sum_{k=1}^{n-1} b_k$
end for
return $\left(\sum_{i=0}^{n} b_i \times i\right)$ ▷ Dot product of $(b_1, b_2, \ldots, b_n)$ with indices $(0, 1, \ldots, n)$

from the unencrypted training data. Utilizing these quantiles, we performed binning of each score/result as outlined in our algorithm, executed within the encrypted domain. This process involves binning scores into their respective bins for each class quantile, followed by aggregation as per Eq. 1 Subsequent normalization of these aggregated scores was conducted and the scores were raised as specified in 2 and 3. The transformed scores were then inputted into the ensemble layer classifier, which was trained on encrypted data and adapted for use in the encrypted domain.

4.3 Results

After conducting experiments on the NIST BSSR and the pneumonia datasets, we observed significant improvements in accuracy using the Quantile Power Transform (QPT) method compared to individual classifiers/modalities and the baseline sum rule, a basic form of ensemble learning.

- The enhancement in accuracy and the AUC-ROC (Area Under the Receiver Operating Characteristic Curve) with the application of QPT is noteworthy. As can be observed in 4, the accuracy and ROC-AUC increase with a higher number of quantiles. However, it is crucial to select an optimal number of quantiles for practicality in the encrypted domain. This is because computations in Fully Homomorphic Encryption (FHE) are inherently slow, and an excessive number of quantiles can lead to impractical computation times.
- As mentioned earlier, the selection of parameters is pivotal. Using higher values for these parameters yields sharper **compval** transitions (the plot of different values of a against its **compval**). The **compval** represents the output of the comparison algorithm for different values being compared against a threshold (i.e., if $a > b$, then **compval** is close to 1; if $a < b$, it is close to 0; and approximately 0.5 otherwise). We observed that lower values of the parameters n, d_g, and d_f in the comparison

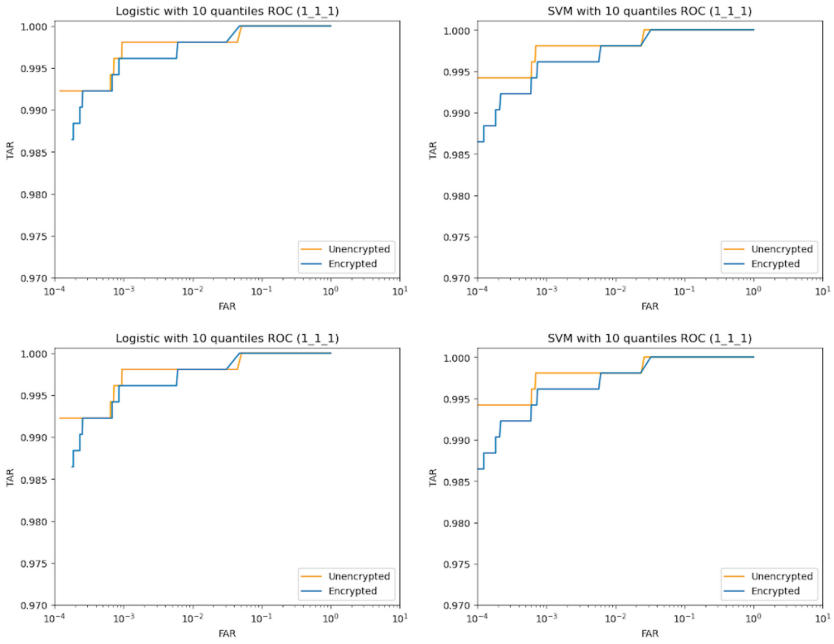


Fig. 5. ROC curves that contrast the performance of logistic regression and SVM classifiers, both encrypted and unencrypted, on NIST-BSSR biometric scores that have been modified by a quantile-power transformation process.

algorithm yield better results for peak and maxIdx calculations due to smoother transitions, which are desirable. This is advantageous because these parameter settings also expedite the computation process, a significant consideration in the context of Fully Homomorphic Encryption (FHE). While higher values of n, d_g, and d_f result in a sharper plot, which is beneficial for comparison algorithms as it leads to a more distinct differentiation between values greater than and less than the comparison value, lower values of these parameters produce smoother transitions. Smoother transitions are preferable for peak and maxIdx calculations as they enhance the stability and accuracy of the algorithm. This can be seen in 6

– There is a slight variation in the ROC between the encrypted and unencrypted implementations. This variation is anticipated due to the approximation-based approach inherent in FHE. Despite this, the trade-off between accuracy and privacy is acceptable, particularly in scenarios where security is a paramount concern.
– A key hurdle we tackled in our study involves the computational intensity associated with Fully Homomorphic Encryption (FHE). Specifically, our quantile binning algorithm exhibited an average processing time of 66 s for 2^n (where $n = 14$ in our experiments, though it is scalable with available computational resources) scores, utilizing SIMD techniques for each quantile. This equates to an approximate processing time of 0.004 s for each score during the binning process, which we found to be practically feasible. Consequently, the processing times for binning into 10,

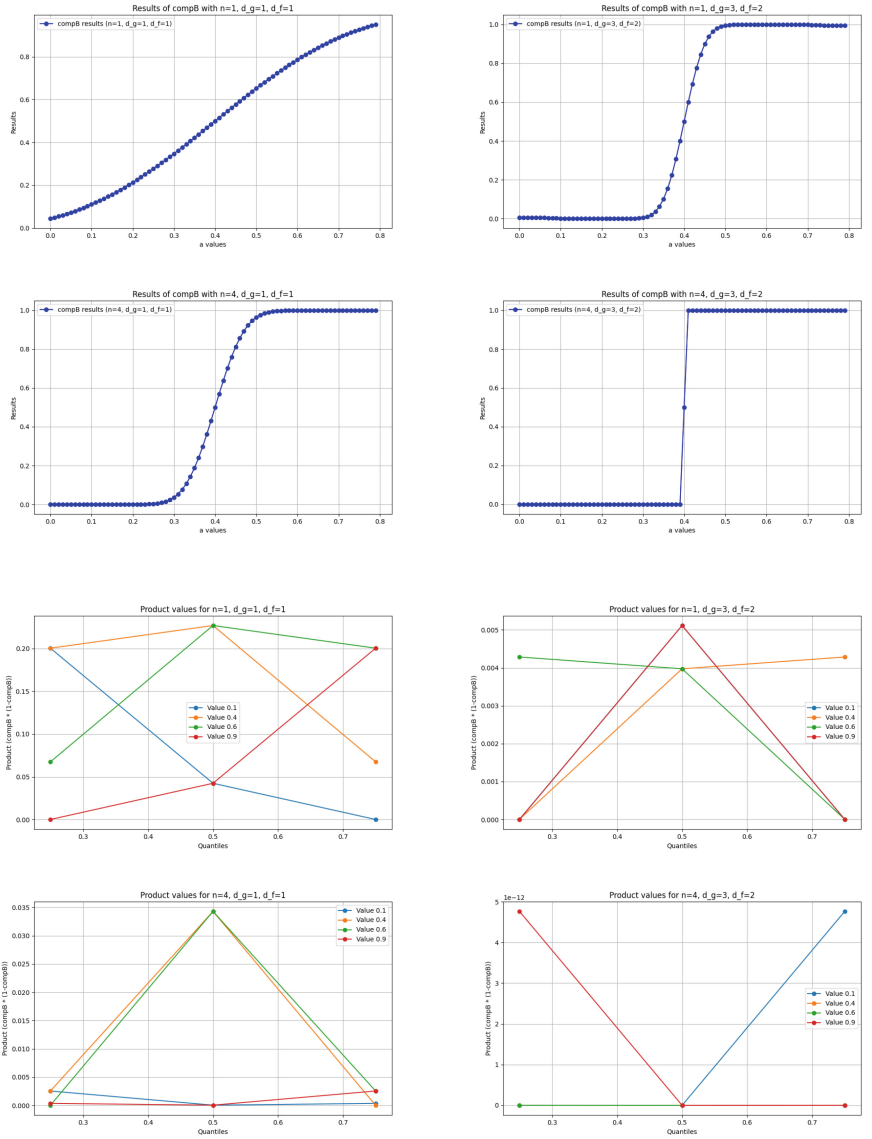


Fig. 6. The first four figures show the **compB** values against different values of a compared to $b = 0.4$ for various combinations of parameters n, d_g, and d_f. Specifically: top-left: $n = 1$, $d_g = 1$, $d_f = 1$; top-right: $n = 1$, $d_g = 3$, $d_f = 2$; middle-left: $n = 4$, $d_g = 1$, $d_f = 1$; middle-right: $n = 4$, $d_g = 3$, $d_f = 2$. The subsequent four figures display the product values (i.e., **compB** $\times$ (1 - **compB**)) to identify the peaks. Each score peaks at the respective quantile it belongs to.

30, and 70 quantiles, according to our experimental setup, are roughly 660 s, 1980 seconds, and 4620 s, respectively.

These findings underscore the efficacy of QPT in enhancing classification accuracy within an FHE framework, while also highlighting the need to balance computational feasibility with the desired level of security (Fig. 6).

5 Conclusion

In conclusion, our research presents a significant advancement in the field of machine learning, particularly in the secure and efficient processing of sensitive data. We have demonstrated that the integration of Quantile Power Transform (QPT) methods with Fully Homomorphic Encryption (FHE) enhances classification accuracy in encrypted domains. This is evidenced by our experiments conducted on the NIST BSSR and pneumonia datasets. Our findings reveal that the QPT method, especially when coupled with the CKKS encryption scheme, offers a robust approach to handling encrypted data. The method shows notable improvements in accuracy compared to traditional classifiers and baseline ensemble methods like the sum rule. Furthermore, selecting an optimal number of quantiles and appropriate parameters in the comparison algorithm (compB) is crucial in balancing computational feasibility with accuracy in the FHE context.

Minor variations observed between the encrypted and unencrypted implementations highlight the trade-offs between accuracy and privacy. However, in applications where data security is paramount, such as in healthcare and biometrics, this trade-off is justifiable and necessary. Overall, our study underscores the potential of combining advanced cryptographic techniques with machine learning to process sensitive data securely and efficiently, opening new avenues for research and application in fields where data privacy is a critical concern.

References

1. Dietterich, T.G.: Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems, ser. MCS '00. Berlin, Heidelberg: Springer-Verlag, pp. 1–15 (2000)
2. Kumar, J., Singh, A.K., Mohan, A., Buyya, R.: Ensemble learning. In: Encyclopedia of Biometrics (2019). https://api.semanticscholar.org/CorpusID:9963037
3. Sabahi, F.: Cloud computing security threats and responses. In: 2011 IEEE 3rd International Conference on Communication Software and Networks, pp. 245–249 (2011)
4. Modi, C., Patel, D., Borisaniya, B., Patel, A., Rajarajan, M.: A survey on security issues and solutions at different layers of cloud computing. J. Supercomput. **63**, 02 (2013)
5. Papernot, N., McDaniel, P., Sinha, A., Wellman, M.: Towards the science of security and privacy in machine learning (2016). https://arxiv.org/abs/1611.03814
6. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
7. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems, Guyon, I. (eds.) vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf

8. Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., Alonso-Betanzos, A.: Ensemble feature selection: homogeneous and heterogeneous approaches. Knowl. Based Syst. **118**, 124–139 (2017). https://www.sciencedirect.com/science/article/pii/S0950705116304749
9. Delgado, R.: A semi-hard voting combiner scheme to ensemble multi-class probabilistic classifiers. Appl. Intell. **52**, 3653–3677 (2022)
10. Nti, I.K., Adekoya, A.F., Weyori, B.A.: A comprehensive evaluation of ensemble learning for stock-market prediction. J. Big Data **7**(1), 1–40 (2020). https://doi.org/10.1186/s40537-020-00299-5
11. Montgomery, J.M., Hollenbach, F.M., Ward, M.D.: Improving predictions using ensemble bayesian model averaging. Polit. Anal. **20**(3), 271–291 (2012)
12. Latif-Shabgahi, G.-R.: A novel algorithm for weighted average voting used in fault tolerant computing systems. Microprocess. Microsyst. **28**(7), 357–361 (2004)
13. Hopkinson, B.M., King, A.C., Owen, D.P., Johnson-Roberson, M., Long, M.H., Bhandarkar, S.M.: Automated classification of three-dimensional reconstructions of coral reefs using convolutional neural networks. PLoS ONE **15**(3), e0230671 (2020)
14. Khan, W., Ghazanfar, M.A., Azam, M.A., Karami, A., Alyoubi, K.H., Alfakeeh, A.S.: Stock market prediction using machine learning classifiers and social media, news. J. Ambient Intell. Humanized Comput. **13**, 1–24 (2020)
15. Soares, C., Brazdil, P.B., Kuba, P.: A meta-learning method to select the kernel width in support vector regression. Mach. Learn. **54**, 195–209 (2004)
16. Kuruvayil, S., Palaniswamy, S.: Emotion recognition from facial images with simultaneous occlusion, pose and illumination variations using meta-learning. J. King Saud Univ. Comput. Inf. Sci. **34**(9), 7271–7282 (2022)
17. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5149–5169 (2021)
18. Monteiro, J.P., Ramos, D., Carneiro, D., Duarte, F., Fernandes, J.M., Novais, P.: Meta-learning and the new challenges of machine learning. Int. J. Intell. Syst. **36**(11), 6240–6272 (2021)
19. Haghighi, F., Omranpour, H.: Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition. Knowl. Based Syst. **220**, 106940 (2021)
20. Gong, X., et al.: Federated learning with privacy-preserving ensemble attention distillation (2022)
21. Basak, J., Kate, K., Tyagi, V., Ratha, N.: QPLC: a novel multimodal biometric score fusion method. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, vol. 2010, pp. 46–52 (2010)
22. Cheon, J.H., Kim, D., Kim, D., Lee, H.H., Lee, K.: Numerical method for comparison on homomorphically encrypted numbers. Cryptology ePrint Archive, Paper 2019/417 (2019). https://eprint.iacr.org/2019/417, https://eprint.iacr.org/2019/417
23. Yang, J., et al.: MedMNIST V2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification. Sci. Data **10**(1), 41 (2023)

Capturing Temporal Components for Time Series Classification

Venkata Ragavendra Vavilthota[1], Ranjith Ramanathan[2], and Sathyanarayanan N. Aakur[3(✉)]

[1] Department of Computer Science, Oklahoma State University, Stillwater, OK 74078, USA
[2] Department of Animal and Food Sciences, Oklahoma State University, Stillwater, OK 74078, USA
[3] Department of Computer Science and Software Engineering, Auburn University, Auburn 36849, USA
san0028@auburn.edu

Abstract. Analyzing sequential data is crucial in many domains, particularly due to the abundance of data collected from the Internet of Things paradigm. Time series classification, the task of categorizing sequential data, has gained prominence, with machine learning approaches demonstrating remarkable performance on public benchmark datasets. However, progress has primarily been in designing architectures for learning representations from raw data at fixed (or ideal) time scales, which can fail to generalize to longer sequences. This work introduces a *compositional representation learning* approach trained on statistically coherent components extracted from sequential data. Based on a multi-scale change space, an unsupervised approach is proposed to segment the sequential data into chunks with similar statistical properties. A sequence-based encoder model is trained in a multi-task setting to learn compositional representations from these temporal components for time series classification. We demonstrate its effectiveness through extensive experiments on publicly available time series classification benchmarks. Evaluating the coherence of segmented components shows its competitive performance on the unsupervised segmentation task.

Keywords: Time-series classification · Temporal Compositionality · Time Series Segmentation

1 Introduction

Time series data is ubiquitous in many domains, such as healthcare [30] and robotics [39]. Given the widespread presence of sensors and smart devices, abundant sequential (time series) data across different domains has been collected, giving rise to several important tasks in time series analysis, such as classification, segmentation, and anomaly detection. Time series classification is one task that has received significant attention in recent years. The goal is to learn robust features from sequential data to classify them into their respective categories. Machine learning approaches [29,31], particularly deep learning

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 215–230, 2025.
https://doi.org/10.1007/978-3-031-78107-0_14

approaches [36,44], have shown tremendous progress in learning models for time series data classification and have resulted in interesting applications such as sleep state segmentation [30] and pandemic modeling [9], to name a view.

The sequential nature of the time series data offers several challenges for classification. First, the sequence length can vary across samples within categories, which requires learning representations robust to such intra-class variations. Second, understanding the ideal time scale for extracting meaningful patterns is challenging, primarily caused by measurement errors and phase/amplitude changes across samples. Finally, long-duration sequences can have dependencies that span different time scales and pose a significant challenge to representation learning approaches. While driving tremendous progress, learning from raw signals relies heavily on representation learning mechanisms to capture intricate, compositional properties for tackling these challenges. Explicitly capturing the underlying temporal structure of signals in the representation can help alleviate this dependency and lead to more robust performance on downstream tasks. Such representations have shown tremendous potential in scene recognition tasks [28] by considering objects as atomic components that combine to compose the overall scene. However, time series may not have such clear distinctions for recognizing boundaries between components, requiring a novel paradigm for defining and detecting temporal components in sequential data.

In this work, we propose to capture the different atomic components that combine to form these signals in a compositional representation. We consider a time series data point, or signal, to be a sequence of data points ordered by some condition and can be segmented into chunks that share semantic or statistical properties. These chunks, or sub-series, are called *components* of the overall signal. Rather than learning representations over the raw sequential data, representations from this sequence of *components* can result in a compositional feature that can span longer durations with reduced computational complexity. The overall approach is illustrated in Fig. 1. We first establish a multi-scale change space (Sect. 3.1) to segment (or tokenize) the signal into components at different temporal scales. Then, we learn compositional representations (Sect. 3.2) from these segments in a multi-task learning setting. Extensive evaluation (Sect. 4) on publicly available benchmark datasets shows that the approach performs competitively with state-of-the-art approaches and scales well to longer duration time series data. These components are remarkably similar to natural segments found in time series data (Sect. 5), and the approach can naturally be extended to unsupervised time series segmentation. Without bells and whistles, the approach performs competitively to state-of-the-art techniques designed explicitly for segmentation and outperforms other non-learning-based methods.

The **contributions** of our approach are four-fold: (i) we are, to the best of our knowledge, to introduce a multi-scale change space for time series data to segment them into statistically atomic components, (ii) we introduce the notion of compositional feature learning from temporally segmented components in time series data rather than modeling the raw data points, (iii) we show that the temporal components detected by the algorithm are highly

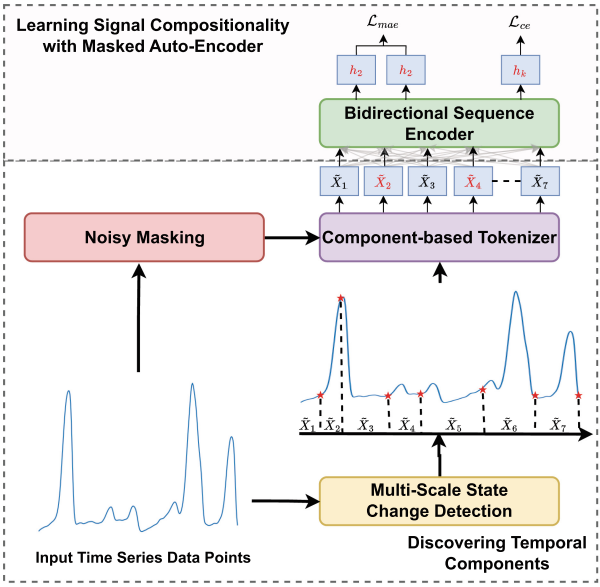


Fig. 1. Overall architecture of the proposed approach is illustrated here. First, we introduce a multi-scale state change detection model to segment sequential data into components and then use a sequence-based encoder to learn compositional representations for time series classification.

correlated with natural boundaries in time series data by evaluating it on the time series segmentation task, achieving state-of-the-art performance compared with other non-learning-based approaches, and (iv) we establish a competitive baseline that provides competitive performance with the state-of-the-art approaches on benchmark datasets for both time series classification and segmentation with limited training needs and without explicit handcrafting.

We structure the paper as follows. We review the relevant literature and techniques used in this work in Sect. 2, followed by an overview and detailed explanation of the proposed approach in Sect. 3. We present and analyze the quantitative results in Sect. 4 and demonstrate how it can be expanded to tackle other time series analysis tasks such as unsupervised segmentation in Sect. 5. Finally, in Sect. 6, we discuss its limitations and future directions.

2 Related Work

Time series classification has been tackled through three major types of approaches. *Classical* approaches, such as those based on handcrafted feature learning [19,27,29,31], have attempted to learn discriminative features from modeling the time series at different scales through techniques such as shapelet transforms [19,27], distance-based transforms [7,29], and bag-of-symbols [31,35],

to name a few. However, their computational complexity increases almost exponentially as the duration of the time series increases, and hence, they appear to hit a wall of scalability. *Deep learning*-based approaches, using architectures such as convolutional neural networks (CNNs) [24] and transformers [41], have opened a wave of large models pre-trained on significant amounts of data [22]. Deep learning models have focused on modeling the data at the ideal time scale [8,36] for capturing robust representations using different backbones such as CNNs [15,34,42,44,47,48], recurrent neural networks (RNNs) [37,38], and transformers [14,46]. *Ensemble-based* approaches [12,27,35,36], i.e., using multiple predictions from the different aspects of the same time series data, have made significant strides in establishing the state-of-the-art performance on several benchmark datasets [2,3,10]. Our work, however, offers a novel framework to capture multi-scale representations by detecting temporal components at different time scales and integrating them in a unified representation without the need for ensembling and additional overhead in the form of annotations.

Approaches to **time series segmentation** have primarily focused on detecting boundaries in sequential data through heuristic-based, domain-specific approaches. Broadly categorized into three categories [40], the time series is segmented by comparing the features of consecutive fixed-size windows using their likelihood of belonging to the same segment [21], assessing homogeneity using kernels [18] or mapping them into graph-based representation for extracting subgraphs (segments) through heuristics such as pairwise similarity [5]. Search-based approaches [1,13,33] and learning-based approaches [11,17,32] have offered a way forward to domain-agnostic segmentation by learning sequence-level representations and segmenting them based on similarity measures. The former assigns costs to plausible boundaries and finds optimal segments by minimizing these costs, while the latter focuses on learning boundaries through pre-text tasks as self-supervision. Many of these approaches require the number of segments to be pre-defined, with learning-based approaches such as ClaSP [32] being a notable exception. Our approach built on BIC-based tokenization (Sect. 3.1) belongs to the search-based approach category and performs *domain-agnostic* segmentation by comparing statistical similarity measures without training.

3 Proposed Framework

In this section, we outline the proposed framework to extract temporal components from sequential data and learn a robust, compositional representation in a multi-task setting. We first outline the problem formulation to provide an overview of the approach and then introduce the multi-scale change space used to discover temporal components in signals. Finally, we introduce the representation learning mechanism used to combine these temporal components into a robust representation.

Problem Formulation. We address the task of classifying univariate time series data by decomposing the signal into its constituent parts. We aim to characterize and build a rich signal representation by detecting parts (sub-series) that

compose the overall signal. Inspired by theories of compositional event understanding [45], we consider these parts atomic, i.e., each sub-series cannot be broken down into smaller components. To this end, we consider a multi-scale approach to identify these components at different time scales to account for the unique challenges inherent in time series data, such as variations that are introduced during data collection [3,10] (i.e., sampling rate and record length) and unavoidable intra-class variations (such as amplitude offset and warping). Following prior works on state-spaces [25,26], we define a signal-dependent change scale-space that captures the multi-scale structure of the signal based on its temporal change points. The overall architecture is illustrated in Fig. 1. First, we identify the temporal change points in the signal using a statistics-based multi-scale organization (Sect. 3.1), which allows us to break the signal down into its components. Second, we learn compositional relationships from these signal components using a bidirectional sequence learning model (Sect. 3.2) trained in a multi-task setting. Combined, these two steps help identify atomic components in time series signals and help capture their temporal structure in a purely bottom-up fashion without auxiliary data.

3.1 Discovering Temporal Components of Signals

The first step in our approach is to discover temporal sub-components that compose time series signals. These sub-components are temporal chunks whose statistics (mean, variance, etc.) are consistent within the sub-series yet vary significantly with neighboring chunks. Hence, detecting the change in statistics at multiple time scales allows us to discover these temporal components in univariate signals. We use the premise from statistics-based speaker-turn detection approaches [6,25] to define a function TSCS (Time Series Change Space) to capture the temporal change space in time series data ($X_{0,N} = \{x_1, x_2, x_3, \ldots, x_N\}$). It is a two-dimensional function over time (t) and temporal scale (δ) that characterizes the varying statistics between two sub-series $t-\delta$ and $t+\delta$ to detect a possible temporal change point (time series component) at time t, given a temporal scale δ. We cast this formulation as a hypothesis-testing problem. The null hypothesis is that two consecutive chunks are different and thus require two different models to represent them individually. The alternative hypothesis is that they are very similar and belong to a single, longer chunk one model can represent. We evaluate each hypothesis by fitting a single Gaussian model [6] for the chunks from each hypothesis. Hence, the difference in the Bayesian Information Criterion (BIC) between the two models at time t provides a measure of their separability based on their statistics. Formally, we define the state space ($TSCS(t, \delta)$) as a function of BIC given by

$$\begin{aligned} TSCS(t,\delta) = \frac{\delta}{2}(log|\sigma_{X_{t-\delta,t}}| + log|\sigma_{X_{t,t+\delta}}|) \\ - \delta(log|\Sigma_{X_{t-\delta,t+\delta}}|) + \delta P \end{aligned} \tag{1}$$

where $log|\sigma_{X_{t-\delta,t}}|$ and $log|\sigma_{X_{t,t+\delta}}|$ refer to the BIC of the single Gaussian representation for the subseries from time $t-\delta$ to t and from t to $t+\delta$, respectively;

$log|\Sigma_{X_{t-\delta,t+\delta}}|$ refers to the BIC of a multivariate jointly considering both sub-series; and P is a penalty term to account for the size of the subseries considered and is typically set to $log(T)$ where T is the length of the subseries considered. Higher values of TSCS indicate that the two sub-series are separate components, i.e., a change in statistics is likely and indicates the presence of a change point.

Given this change space, we can build a multi-scale representation by varying the time scale δ over a range and summing up the resulting BIC curves. Formally, this can be defined as

$$MS{-}TSCS(t) = \sum_{\delta \in \Delta} TSCS(t, \delta) \tag{2}$$

where Δ is the set of all time scales for detecting time series components. In practice, we consider Δ to range from 10 time steps to 500 time steps. We then pass the curve from $MS{-}TSCS(t)$ through a low pass filter to extract peaks that provide possible time steps to segment the time series. We select peaks with high *saliency*, i.e., if it is more than two standard deviations from its neighbors. This is a common approach in statistics-based outlier detection literature [6,25] and provides a good measure of temporal saliency for this problem. Given the temporal change locations, the *ideal* number of segments per dataset is computed as the average number of components across classes in the training set. We find that considering too few (or smaller) values in Δ will result in fewer segments and poor representations. Note that not all time series will have such components that are statistically separable. We use a uniform sampling approach to split the series into 15 equal segments in these cases. Empirically, segmenting chunks into more than 50 segments is not ideal and could degrade the performance, particularly on smaller datasets.

3.2 Capturing Signal Compositionality

The second step in our approach is to learn robust representations from the multi-scale components extracted using the MS-TSCS function defined in Sect. 3.1. Given the ideal number of segments K, the input sequence is *tokenized* $X_N{=}\{x_1, x_2, x_3, \ldots, x_N\}$ into its constituent segments $\tilde{X}_K{=}\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3, \ldots, \tilde{X}_k\}$. For capturing *compositional* representations, we then use a masked auto-encoding loss function [4] to train the encoding model (with parameters Θ). The masked auto-encoding loss randomly masks $M < k$ components and forces the encoder to independently predict the masked components by conditioning on the context provided by the unmasked components. Given the tokenized time series data $\tilde{X}_K = \{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3, \ldots, \tilde{X}_k\}$ and masked components $M = \{m_1, m_2, \ldots, m_{|M|}\}$, the masked auto-encoding loss is

$$\mathcal{L}_{mae} = -\sum_{X_i \in \mathcal{C}} log \prod_{m \in M} p(\tilde{X}_m | \tilde{X}_{K \setminus M}) \tag{3}$$

where $p(\tilde{X}_m | \tilde{X}_{K \setminus M})$ is the probability of predicting the randomly masked components in set $\{M\}$. This probability is computed as the mean squared error

over the masked component's values. We use a bidirectional LSTM [20] as our encoder and ensure that the mask is bidirectional, i.e., the context for predicting the masked component is present on both sides of the mask. This masking procedure has successfully been used to train text-based [22] and image-based [43] encoders. We extend the formulation to univariate time series data. The hidden states of the forward and backward LSTM cells, h_t^f and h_t^b, respectively, are concatenated and used as the feature representation for time series classification, optimized by the cross-entropy loss ($\mathcal{L}_{CE}$). Hence, the overall objective function is given by

$$\mathcal{L}_{tot} = \lambda_1 \mathcal{L}_{mae} + \lambda_2 \mathcal{L}_{CE} \tag{4}$$

where λ_1 and λ_2 are tunable parameters that trade-off between the two losses. The values of λ_1 and λ_2 are varied according to a pre-set schedule to balance the representation learning capabilities from the self-supervised masked auto-encoding loss ($\mathcal{L}_{mae}$) and the discriminative, class-specific properties imbued by the supervised cross-entropy loss ($\mathcal{L}_{ce}$).

Implementation Details. We use a bidirectional LSTM model with a hidden size of 160 neurons, followed by a dense layer with 320 neurons, as our encoder architecture. The ReLU activation is used for all layers. All segmented components are padded as necessary to be equal in length. We use 5% of the training data for validation. We use the same pre-processing as previous work [44]. λ_1 and λ_2 are varied as follows: for the first 100 epochs, $\lambda_1 = 1$ and $\lambda_2 = 0$, then $\lambda_1 = 2$ and $\lambda_2 = 1$. The network is trained for 250 epochs or until convergence, i.e., the loss does not improve on the validation set. All experiments were conducted on a workstation server with a 32-core AMD ThreadRipper CPU, 128 GB RAM, and an NVIDIA RTX 3060.

4 Experimental Evaluation

In this section, we present the results from the experimental evaluation of the proposed approach. We begin with a discussion on the experimental setup, followed by the quantitative results, and conclude with a qualitative discussion on the representations learned by the approach.

4.1 Experimental Setup

Data. We evaluate the proposed approach on 85 datasets collated in the UCR time series archive [3]. It consists of univariate time series datasets collected from different sensors and domains such as health care, speech reorganization, and spectrum analysis, to name a few. The archive provides a comprehensive benchmark for evaluating time series classification models [15,36,44] across diverse datasets with varying characteristics. The number of classes in each dataset ranges from 2 to 6, the number of time steps per sample varies from 24 to 2709, and the number of training samples per dataset from 16 to 8926. Additionally,

we evaluate the approach on 15 datasets with the longest timesteps from the UCR-85 [3] and the UCR-128 [10] datasets to evaluate its ability to capture robust representations from time series with longer duration. We use the official train and test splits on all datasets for a fair comparison with prior works. Average accuracy across all datasets is used to quantify the performance on the UCR time series archive. Code and performance for baselines are obtained from publicly available implementations of prior works [36,44].

Baselines. We compare against state-of-the-art univariate time series classification models, which use different representation learning backbones and propose robust learning methods to account for high intra-class variation common in time series data. Chiefly, we compare against models with CNN backbones [15,34,42,44,47,48], transformer backbones [14,46], RNN backbones [37,38], and other hand-crafted features such as shapelet transforms [19], distance-based metrics [7,29], and bag-of-symbols [31]. We also compare against ensembles [12,27,35,36], which explicitly capture representations at multiple time scales, which can require additional overhead for training.

Table 1. Performance evaluation of the proposed approach with state-of-the-art approaches on 85 datasets from the UCR time series archive [3,10]

Approach	Ensemble?	Backbone	Accuracy
TST [46]	✗	Transformer	64.901
MCDCNN [48]	✗	CNN	68.551
TWIESN [37]	✗	RNN	68.636
TS-Encoder [34]	✗	CNN	71.909
Time-CNN [47]	✗	CNN	72.284
DTW [7]	✗	Distance	74.040
TS-TCC [14]	✗	CNN-Transformer	77.764
TNC [38]	✗	Bi-RNN	77.896
PF [29]	✗	Distance	80.419
T-Loss [15]	✗	Dilated CNN	80.482
BOSS [31]	✗	Bag of Symbols	81.019
FCN [42]	✗	CNN	81.634
ResNet [42]	✗	CNN	82.201
ST [19]	✗	Shapelets	82.236
TS2Vec [44]	✗	Dilated CNN	82.934
Ours	✗	Bi-RNN	**83.309**
TS-CHIEF [35]	✓	Bag of Symbols	84.641
HIVE-COTE [27]	✓	Multiple	84.714
OS-CNN [36]	✓	CNN	84.774
ROCKET [12]	✓	CNN	**85.077**

4.2 Quantitative Evaluation

We present the performance of the approach on the UCR-85 archive in Table 1. We outperform other approaches on the benchmark while offering competitive performance to those designed to work in an ensemble. Interestingly, most state-of-the-art techniques are based on CNNs, with much effort spent finding optimal receptive field sizes for learning robust features at multiple timescales. Sequence-based approaches, such as those based on Transformers and RNNs, have struggled in this benchmark, mostly due to the limited training examples in many datasets. We, however, significantly outperform other sequence-based approaches and provide improvements of almost 5.5% in absolute accuracy points over the closest RNN-based approach (TNC [38]). It also provides the best performance (out of non-ensemble approaches) on 17 datasets (also called *wins* in prior literature [36]) out of the 85 benchmark datasets. Additionally, it has an average rank of 5.35, performing competitively with other non-ensemble approaches. Ensemble models outperform all non-ensemble models by explicitly modeling sequential data by representing the sequential data at different time scales. However, they introduce additional overhead for handcrafting and fine-tuning multiple models.

Table 2. Performance on 15 longest sequence time series data from the UCR Archives [3], compared against state-of-the-art models with different backbones.

Backbone→	CNN		Transformer		Bi-RNN	
Dataset ↓	TS2Vec [44]	OS-CNN [36]	TS-TCC [14]	TST [46]	TNC [38]	Ours
Rock	**70.00**	55.00	60.00	68.00	58.00	**70.00**
HandOutlines	92.20	92.95	72.40	73.50	93.00	**94.05**
HouseTwenty	91.60	**94.87**	79.00	81.50	78.20	92.44
InlineSkate	41.50	**42.92**	34.70	28.70	37.80	41.09
EthanolLevel	46.80	73.08	48.60	26.00	42.40	**87.00**
SemgHandSubjectCh2	**95.10**	71.84	75.30	48.40	77.10	91.56
SemgHandGenderCh2	**96.30**	85.61	83.70	72.50	88.20	89.33
SemgHandMovementCh2	**86.00**	56.62	61.30	42.00	59.30	78.22
EOGHorizontalSignal	53.90	**63.97**	40.10	37.30	44.20	57.73
EOGVerticalSignal	50.30	47.76	37.60	29.80	39.20	**51.10**
Haptics	**52.60**	51.01	39.60	35.70	47.40	50.32
Mallat	91.40	96.38	92.20	71.30	87.10	**97.10**
MixedShapesRegularTrain	91.70	**96.09**	85.50	87.90	91.10	93.69
MixedShapesSmallTrain	86.10	**91.79**	73.50	82.80	81.30	87.96
StarLightCurves	96.90	97.51	96.70	94.90	96.80	**97.78**
Average	76.16	74.49	65.35	58.69	68.07	**78.62**

Performance on Longer Sequence Data. While the overall UCR-85 archive performance is excellent, we also examine the ability of the proposed approach to capture long-range dependencies when presented with time series data of longer durations. We select a subset of the UCR-128 archive, which contains additional datasets of longer duration. Specifically, we select 15 datasets with more than 1000 timesteps per sample without incomplete data. Table 2 presents a summary of the results. As can be seen, we provide competitive performance with top-performing baselines with different backbone architectures. We have an average accuracy of 78.62%, an average rank of 1.72, and provide "wins" in 6 out of the 15 long sequence datasets. It significantly improves over transformer-based (TS-TCC and TST) and RNN-based (TNC) baselines, which are trained to specifically model longer sequences through specialized training procedures such as contrastive learning. These results indicate the approach can capture robust representations from long sequences without complex ensemble processing.

Table 3. Ablation studies on the UCR-85 archive [3] to assess the impact of each component on the overall performance.

Backbone	MS-TSCS	$\mathcal{L}_{mae}$	$\mathcal{L}_{CE}$	Accuracy
Bi-LSTM	✓	✓	✓	**83.31**
Bi-LSTM	✗	✓	✓	73.68
Bi-LSTM	✓	✗	✓	75.31
Bi-LSTM	✓	✓	✗	74.28
Bi-LSTM	✗	✗	✓	68.33
Bi-RNN	✓	✓	✓	81.54
Uni-LSTM	✓	✓	✓	79.55

Constrained Hardware Requirements. Our approach is designed to be simple and lightweight for use in settings with constrained training requirements, such as time and space budgets (i.e., limited training time, constrained hardware requirements, and limiting the number of parameters). Our model achieves competitive performance with 440k parameters and converges training on all datasets in 4 h (on average over ten runs). For comparison, the current non-ensemble state-of-the-art approaches, TS2Vec (637k parameters) and ResNet (479k parameters), have more parameters and take longer to converge on a constrained hardware setup (32-core AMD ThreadRipper and NVIDIA RTX 3060). Similarly, on average, the BIC-based tokenization process (Sect. 1) takes 500 ms for a sequence of 1000 data points, running in a single-threaded CPU-only application while having significantly less overhead for storing the components compared with other approaches.

Ablation Studies. We systematically examine the impact of each module and summarize the results in Table 3. Specifically, we assess the effects of the multi-scale component discovery module (Sect. 3.1) and the choice of encoder model (Sect. 3.2). Removing the component discovery model and using a fixed number of components for all datasets (set to 25, the median number of components across datasets) significantly hurts the performance. We also evaluate the strength of the learned representations by using a kNN instead of end-to-end training by removing $\mathcal{L}_{CE}$ from Eq. 4. While the loss in performance is expected, it does perform decently, indicating that the unsupervised loss function helps learn robust features. Removing $\mathcal{L}_{mae}$ results in significantly worse performance. Using bidirectional LSTMs instead of unidirectional LSTMs helps capture context and provides a more robust performance across all 85 datasets in the UCR archive.

5 Extension to Unsupervised Time Series Segmentation

In addition to evaluating the performance of our approach on time series classification, we assess the quality of the components obtained through the BIC-based segmentation (Sect. 3.1) by evaluating it on the time series segmentation task [16]. The goal of time series segmentation is to identify natural segments caused by *change points* in sequential data where there are sudden changes in statistical properties of the time series due to changes in events captured by the data. For example, these changes could point to transitions between actions performed by a subject. The UTSA benchmark [16] introduces a set of 32 datasets derived from the UCR archive [3] and provides human-annotated segments of datasets across 16 different use cases from biological, mechanical, and synthetic processes. Each use case in the benchmark contains, on average, 2 to 3

Table 4. Evaluation of the BIC-based tokenization approach on the **time series segmentation** task [16].

Approach	Learning Phase?	Pre-Defined Window?	Mean Covering
BinSeg	✗	✓	52.4 ± 30.6
PELT	✗	✓	50.4 ± 30.0
Window	✗	✓	53.8 ± 12.9
BOCD	✗	✓	55.5 ± 14.4
ESPRESSO	✗	✓	58.0 ± 15.8
Ours	✗	✗	$\mathbf{72.7 \pm 12.5}$
FLOSS	✓	✓	79.0 ± 17.2
ClaSP	✓	✗	$\mathbf{79.8 \pm 20.4}$
Ours	✗	✓	$\mathbf{78.3 \pm 12.9}$

segments derived from real, semi-synthetic, and artificial changes and provides a considerable challenge for unsupervised time series segmentation.

We use the components discovered using the multi-scale change space model as segments and assess the quality of the segmentations on the UTSA benchmark. We compare against a variety of baselines such as BinSeg [33], PELT [23], Window [40], BOCD [1], ESPRESSO [11], FLOSS [17], and ClaSP [32], which represent the commonly used state-of-the-art unsupervised segmentation approaches. We use the mean covering with standard deviation as a metric to quantify the performance of the approaches. Based on the Jaccard index, the covering score provides a weighted overlap between the ground truth and the predicted segments. Higher values indicate better alignment between the predicted and the ground truth segments. We report results from the implementations from ClaSP [32] for a fair comparison and consistent experimental setup.

Table 4 summarizes the results. We significantly outperform other non-learning-based approaches that require a pre-defined period size (temporal window) corresponding to the ideal time scale at which the change points can be detected reliably. This value is often domain-dependent and requires extensive handcrafting (of architecture or features) to capture, especially in time series classification and segmentation. Our approach can automatically search for this using the multi-scale change space and considers change points at different temporal granularities. When given this optimal window, we establish the change space at this time scale and perform segmentation. As can be seen, we perform competitively with learning-based approaches and further widen the gap with the non-learning-based approaches. Interestingly, we perform exceptionally well without the optimal time scale, indicating that the multi-scale change space captures the change points at time scales approaching the ideal scale. Some example segmentations are shown in Fig. 2, where it can be seen that our

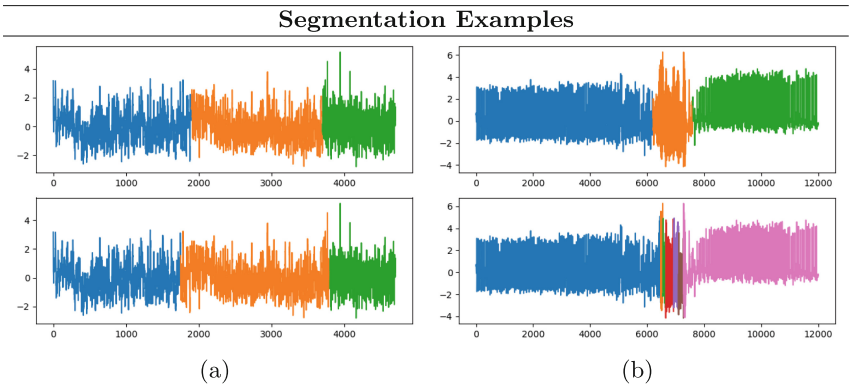


Fig. 2. Qualitative visalization of (a) a successful segmentation and (b) unsuccessful segmentation on the GreatBarbet2 and SuddenCardianDeath1 datasets, respectively. The first row shows ground truth segments, and the second shows predicted segments.

approach can segment signals into their components without training and supervision. Although it over segments in some instances, the segments are statistically meaningful, are captured at multiple time scales, and do not always correspond to the ground change points extracted at a single time scale. For example, in Fig. 2(b), we see that over-segmentation occurs during periods of intense changes and captures fine-grained change points but has excellent coverage during stable regions on either side of this rapidly changing segment. Note that our approach detects the temporal components in a time-scale and class-agnostic manner and does not have access to the ideal time scale at which the ground truth is annotated. Despite this over-segmentation, it allows us to capture robust features for classification.

6 Discussion and Future Work

In this work, we presented a novel multi-scale change-space approach to discover temporal components in univariate time series data and provide an intuitive way to tokenize time series data using statistical measures. Given these components, we learn compositional representations using sequence-based encoders by training the model as a masked, denoising auto-encoder. Evaluation on 85 publicly available datasets on the benchmark UCR-85 archive demonstrates its effectiveness in learning robust representations. Additional experiments on segmentation benchmarks demonstrate that the detected components are highly correlated with naturally occurring segments found in time series data. We aim to extend this formulation to capture part-whole hierarchies for learning *hierarchical* compositional representations from multi-modal and multi-variate time series data with longer temporal durations.

Acknowledgements. This work was supported by the U.S. National Science Foundation Grant IIS 2348689 and IIS 2348690 and U.S. Department of Agriculture Grant 2023-69014-39716-1030191. We thank the authors of the UCR and UTSA datasets for their service by making the data publicly available.

References

1. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742 (2007)
2. Bagnall, A., et al.: The uea multivariate time series classification archive. arXiv preprint arXiv:1811.00075 (2018)
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **31**, 606–660 (2017)
4. Bao, H., et al.: Unilmv2: pseudo-masked language models for unified language model pre-training. In: International Conference on Machine Learning, pp. 642–652. PMLR (2020)
5. Chen, H., Chu, L.: Graph-based change-point analysis. Ann. Rev. Stat. Appl. **10**, 475–499 (2023)

6. Chen, S., Gopalakrishnan, P., et al.: Speaker, environment and channel change detection and clustering via the bayesian information criterion. In: DARPA Broadcast News Transcription and Understanding Workshop, vol. 8, pp. 127–132. Citeseer (1998)
7. Chen, Y., Hu, B., Keogh, E., Batista, G.E.: Dtw-d: time series semi-supervised learning from a single example. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 383–391 (2013)
8. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995 (2016)
9. Dash, S., Chakraborty, C., Giri, S.K., Pani, S.K.: Intelligent computing on time-series data analysis and prediction of covid-19 pandemics. Pattern Recogn. Lett. **151**, 69–75 (2021)
10. Dau, H.A., et al.: The ucr time series archive. IEEE/CAA J. Automatica Sinica **6**(6), 1293–1305 (2019)
11. Deldari, S., Smith, D.V., Sadri, A., Salim, F.: Espresso: Entropy and shape aware time-series segmentation for processing heterogeneous sensor data. Proc. ACM Interact. Mobile Wearable Ubiq. Technol. **4**(3), 1–24 (2020)
12. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. **34**(5), 1454–1495 (2020)
13. Draayer, E., Cao, H., Hao, Y.: Reevaluating the change point detection problem with segment-based bayesian online detection. In: ACM International Conference on Information & Knowledge Management, pp. 2989–2993 (2021)
14. Eldele, E., et al.: Time-series representation learning via temporal and contextual contrasting. arXiv preprint arXiv:2106.14112 (2021)
15. Franceschi, J.Y., Dieuleveut, A., Jaggi, M.: Unsupervised scalable representation learning for multivariate time series. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
16. Gharghabi, S., Ding, Y., Yeh, C.C.M., Kamgar, K., Ulanova, L., Keogh, E.: Matrix profile viii: domain agnostic online semantic segmentation at superhuman performance levels. In: IEEE International Conference on Data Mining (ICDM), pp. 117–126. IEEE (2017)
17. Gharghabi, S., et al.: Domain agnostic online semantic segmentation for multi-dimensional time series. Data Min. Knowl. Disc. **33**, 96–130 (2019)
18. Harchaoui, Z., Vallet, F., Lung-Yut-Fong, A., Cappé, O.: A regularized kernel-based approach to unsupervised audio segmentation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1665–1668. IEEE (2009)
19. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Min. Knowl. Disc. **28**, 851–881 (2014)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
21. Kawahara, Y., Sugiyama, M.: Sequential change-point detection based on direct density-ratio estimation. Stat. Anal. Data Mining: ASA Data Sci. J. **5**(2), 114–127 (2012)
22. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
23. Killick, R., Fearnhead, P., Eckley, I.A.: Optimal detection of changepoints with a linear computational cost. J. Am. Stat. Assoc. **107**(500), 1590–1598 (2012)
24. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1d convolutional neural networks and applications: a survey. Mech. Syst. Signal Process. **151**, 107398 (2021)

25. Krishnan, R., Sarkar, S.: Detecting group turn patterns in conversations using audio-video change scale-space. In: International Conference on Pattern Recognition, pp. 137–140. IEEE (2010)
26. Laptev, I., Lindeberg, T.: A multi-scale feature likelihood map for direct evaluation of object hypotheses*. In: Kerckhove, M. (ed.) Scale-Space 2001. LNCS, vol. 2106, pp. 98–110. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-47778-0_9
27. Lines, J., Taylor, S., Bagnall, A.: Time series classification with hive-cote: the hierarchical vote collective of transformation-based ensembles. ACM Trans. Knowl. Discov. Data **12**(5), 1–35 (2018)
28. Locatello, F., et al.: Object-centric learning with slot attention. In: Advances on Neural Information Processing System, vol. 33, pp. 11525–11538 (2020)
29. Lucas, B., et al.: Proximity forest: an effective and scalable distance-based classifier for time series. Data Min. Knowl. Disc. **33**(3), 607–635 (2019)
30. Ramnath, V.L., Katkoori, S.: A smart iot system for continuous sleep state monitoring. In: 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 241–244. IEEE (2020)
31. Schäfer, P.: The boss is concerned with time series classification in the presence of noise. Data Min. Knowl. Disc. **29**, 1505–1530 (2015)
32. Schäfer, P., Ermshaus, A., Leser, U.: Clasp-time series segmentation. In: ACM International Conference on Information & Knowledge Management, pp. 1578–1587 (2021)
33. Sen, A., Srivastava, M.S.: On tests for detecting change in mean. Ann. Stat. 98–108 (1975)
34. Serra, J., Pascual, S., Karatzoglou, A.: Towards a universal neural network encoder for time series. In: International Conference of the Catalan Association for Artificial Intelligence, pp. 120–129 (2018)
35. Shifaz, A., Pelletier, C., Petitjean, F., Webb, G.I.: Ts-chief: a scalable and accurate forest algorithm for time series classification. Data Min. Knowl. Disc. **34**(3), 742–775 (2020)
36. Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., Jiang, J.: Omni-scale cnns: a simple and effective kernel size configuration for time series classification. In: International Conference on Learning Representations (2021)
37. Tanisaro, P., Heidemann, G.: Time series classification using time warping invariant echo state networks. In: IEEE International Conference on Machine Learning and Applications, pp. 831–836. IEEE (2016)
38. Tonekaboni, S., Eytan, D., Goldenberg, A.: Unsupervised representation learning for time series with temporal neighborhood coding. In: International Conference on Learning Representations (2020)
39. Trehan, S., Aakur, S.N.: Towards active vision for action localization with reactive control and predictive learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 783–792 (2022)
40. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Process. **167**, 107299 (2020)
41. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
42. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: International Joint Conference on Neural Networks, pp. 1578–1585. IEEE (2017)
43. Xie, Z., et al.: On data scaling in masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10365–10374 (2023)

44. Yue, Z., et al.: Ts2vec: towards universal representation of time series. In: AAAI Conference on Artificial Intelligence, vol. 36, pp. 8980–8987 (2022)
45. Zacks, J.M., Tversky, B.: Event structure in perception and conception. Psychol. Bull. **127**(1), 3 (2001)
46. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2114–2124 (2021)
47. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. J. Syst. Eng. Electron. **28**(1), 162–169 (2017)
48. Zheng, Y., Liu, Q., Chen, E., Ge, Y., Zhao, J.L.: Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. Front. Comp. Sci. **10**(1), 96–112 (2016). https://doi.org/10.1007/s11704-015-4478-2

Hierarchical Transfer Multi-task Learning Approach for Scene Classification

Reza Khoshkangini[1(✉)], Mohsen Tajgardan[2], Mahtab Jamali[1], Martin Georg Ljungqvist[3], Radu-Casian Mihailescu[1], and Paul Davidsson[1]

[1] Internet of Things and People Research Center, Department of Computer Science and Media Technology, Malmö University, Malmö, Sweden
{reza.khoshkangini,mahtab.jamali,radu.c.mihailescu,paul.davidsson}@mau.se
[2] Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran
tajgardan.m@qut.ac.ir
[3] Axis Communications AB, Lund, Sweden
martin.ljungqvist@axis.com

Abstract. This paper presents a novel Hierarchical Transfer and Multi-task Learning (HTMTL) approach designed to substantially improve the performance of scene classification networks by leveraging the collective influence of diverse scene types. HTMTL is distinguished by its ability to capture the interaction between various scene types, recognizing how context information from one scene category can enhance the classification performance of another. Our method, when applied to the Places365 dataset, demonstrates a significant improvement in the network's ability to accurately identify scene types. By exploiting these inter-scene interactions, HTMTL significantly enhances scene classification performance, making it a potent tool for advancing scene understanding and classification. Additionally, this study explores the contribution of individual tasks and task groupings on the performance of other tasks. To further validate the generality of HTMTL, we applied it to the Cityscapes dataset, where the results also show promise. This indicates the broad applicability and effectiveness of our approach across different datasets and scene types.

Keywords: Scene Classification · Multi-task Learning · Transfer Learning

1 Introduction

Over recent years, significant progress has been made in the field of computer vision, enabling smart systems to interact with their surroundings and recognize environments. In scene classification images are categorized according to their ambient content, layout, and object into predefined scene categories (such as bedroom, park, and jungle). It aims to help computers understand the environment around them by providing a valuable and important description of the

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 231–248, 2025.
https://doi.org/10.1007/978-3-031-78107-0_15

content of an image. The ability to recognize scenes holds significant importance in several applications, including but not limited to security cameras [21] and autonomous driving systems [11].

As scenes tend to have a variety of objects, different layouts, and semantic ambiguity, the task of scene classification can be quite challenging for a network. Inter-class similarity, and intra-class diversity are two crucial factors in scene classification that contribute to the overall effectiveness and robustness of the classification model [36,41].

Indoor and outdoor scene classification is a basic step in order to classify scenes. Indoor and outdoor images are generally different in terms of color, size, layout, light, and other factors for categorizing them into indoor and outdoor categories. This detection helps the network to get a better understanding of the entire environment and objects in the scene. Extensive research has been done on classifying indoor or outdoor environments [34]. This issue was investigated by Simer et al. in 1998 [33], and many researchers have attempted to enhance the accuracy of models in scene classification in recent years, such as [7,22,27,47]. The categorization of indoor and outdoor environments has broader implications for a range of imaging applications such as content-based image retrieval [8,35], robotic [3,6,24], and map depth creation [2].

Although the ability to differentiate between indoor and outdoor environments is undoubtedly advantageous in numerous computer vision applications, it frequently fails to comprehensively comprehend intricate surroundings. A higher level of scene classification involves precisely identifying specific places, such as a forest or a home, which demands a more intricate and concerted effort. This heightened degree of scene comprehension presents novel prospects for a more profound understanding of scenes. Therefore, detecting different scene levels could be crucial for environment detection systems.

Multi-task learning (MTL) is a suitable approach for categorizing scenes into different environmental levels. The objective of MTL is to enhance the performance of individual tasks by utilizing information contained in multiple related tasks [44]. Its application spans a multitude of domains, including but not limited to computer vision [5,19], speech recognition [23], natural language processing [4], and automotive [16,17]. For instance, Lu et al. in [19], enhanced scene recognition by combining a multitask model for different image resolutions with a Sparse Feature Selection-based Manifold Regularization (SFSMR). This semi-supervised learning approach demonstrated substantial improvements in accuracy compared to prior methods. Another study has been investigated by Zheng et al. in [45], where a multi-task metric multi-kernel learning algorithm is used to learn linear transformations of RGB and depth features, making full use of inter-modal relations [42]. Another work [1] presents a deep learning approach utilizing SqueezeNet [13] to enhance classification accuracy by learning from multiple datasets and considering each dataset as a separate learning task. MTL with transformers has been used for scene classification [30].

Task grouping has previously been used for other computer vision tasks [10, 29,31,37] but not for scene classification or using hierarchies.

Although these papers have achieved satisfactory scene classification results with multi-task methods, the impact of different scene levels on each other remains an unsolved challenge. These methods often do not consider or infer the links between the different tasks. For example, to what extent the classification of outdoor natural scenes like a forest can affect the classification of outdoor human-made scenes?

To achieve this objective, we have designed a Hierarchical Multi-task approach based on hard parameter sharing and transfer learning methods to assess the interplay between various scenery levels regarding their positive or negative influence to enhance the predictive performance. We have proposed a hierarchical transfer multi-task approach capable of predicting different levels of scenes at the same time. Two main modules construct the proposed approach. In the former, the hidden layers from the InceptionV3 network [32] are used as the shared layers, which were trained with ImageNet dataset [9]. This will extract more general representations which may help the tasks. The latter is specific layers for each individual task; we train multiple tasks simultaneously by exploiting shared representation obtained from InceptionV3 to learn similar knowledge within a set of each task [44]. In this design, at the head, each high-level task is hierarchically concatenated to the low-level task to transfer knowledge from high to low-level tasks to increase the performance of each task and, consequently, the predictive model. In this learning process, the knowledge in each task can be transferred and taken by other related tasks to improve the generalization performance. This enabled us to study extracting the transferred knowledge between the tasks and quantifying the effect of each and multiple tasks on each other over the training process. Furthermore, at the same time, the shared representation will yield shorter training time, allowing us to add more individual tasks by constructing only the specific layers and training only those layers.

The following research questions (RQs) further elaborate the investigative objectives of our proposed approach:

- **RQ1- Hierarchical Transfer Multi-task Learning:** To what extent could the task of scene classification be performed using transfer and multi-task learning approach?
- **RQ2- Scene Transference Quantification:** To what extent could the scene-level task transference be quantified?

Taken together with these research questions, our work aims to counter the practice noted above by utilizing a multi-task learning approach to improve the scene classification performance in surveillance systems. To answer RQ1, we utilize the share representation collected from the base model InceptionV3 network and develop a predictive multi-task model for different levels of scenes. The output of each task will be injected into the next task to provide the knowledge from one task to other tasks in the late layers. The reported figures revealed how this approach could positively impact the accuracy of the predictive model in both individual and overall prediction. To answer RQ2, we adapted the inter-task-affinity (AF) technique introduced in [15] to quantify the knowledge transfer

through the hierarchy of the tasks. The result of RQ2 allows us to understand which tasks (or sets of tasks) have positive or possibly negative contributions to different levels of scenes. The outcome of this study does not provide a conclusive approach to scene understanding, but they suggest quantifying the interaction between tasks can be a potential approach to scene understanding.

In addition, the result we achieved by employing our hierarchical multi-task system outperforms the existing approaches such as SWAG [28], Hiera-H [25], MAE [12], WaveMix-240/12 [14], InternImage-H [38], and MixMIM-L [18]. Overall, we believe our study presents a considerable improvement compared to the existing scene classification approaches. Further explanation of our approach is given in Sect. 3.

2 Data Representation

This section describes the dataset used for the proposed multi-task scene classification approach:

– **Places365**[46]**:** This dataset contains the following scene levels:
  - Level 1 (Task 1): The highest hierarchical level has three categories: indoor, outdoor natural, outdoor man-made.
  - Level 2 (Task 2): Six categories of different indoor places, including 1) shopping and dining; 2) workplace; 3) home or hotel; 4) transportation; 5) sports, leisure; and 6) cultural.
  - Level 2 (Task 3): Four natural outdoor categories: 1) water, ice, snow; 2) mountains, hills, desert, sky; 3) forest, field, jungle; and 4) man-made elements.
  - Level 2 (Task 4): Six categories of outdoor man-made places: 1) transportation; 2) cultural or historical buildings; 3) sports fields, parks, and leisure spaces; 4) industrial and construction; 5) houses, cabins, gardens, farms; and 6) commercial buildings, shops, markets, cities, towns.
  - New level (Task 5): The specific scene category, for example airfield or tennis-court.

We used a subset of Places365 dataset with 8508 images for training and 1385 images for testing. Furthermore, ten classes were chosen from this dataset, namely: airfield, alley, arena-rodeo, factory, forest, labyrinth, outhouse, podium, squash-court, and tennis-court.

3 Approach

The conceptual view of the proposed HTMLT is illustrated in Fig. 1, where our approach constructed by InceptionV3 as the base model–holding the shared layers trained with ImageNet–and a multi-task network with multiple heads for each specific task.

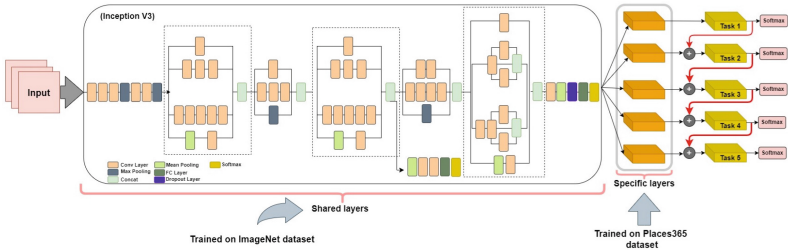


Fig. 1. The conceptual version of the proposed approach.

3.1 Multi-task Learning

Multi-task Learning (MTL) points to learning several related tasks simultaneously with the same network to learn feature representations, leading to improved generalization and more immediate learning [44]. In our specific case of modeling scene, the different scenes are indicated τ_i and $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_m\}$ is the set of all tasks. The total loss vector is:

$$min_\theta L(\theta) = (L_{\tau_1}(\theta), L_{\tau_2}(\theta), \ldots, L_{\tau_m}(\theta)) \tag{1}$$

where $L_{\tau_i}(\theta)$ is the loss function of the i_{th} task (each task refers to a particular scene level mentioned in Sect. 2). A multi-task learning aims to perform joint learning at the same time and optimize all the tasks by utilizing $DS = \{x_j^i, y_j^i\}_{j=1}^{n_i}$ The models learn from the data samples DS, and take advantage of θ_{sh} as the shared layers to compute a loss for each level/task $L(\tau_i|\theta_{sh}, \theta_i)$. Let's assume a multi-task loss function parameterized by $\{\theta_{sh}\} \cup \{\theta_i|\tau_i \in \mathcal{T}\}$, where "$\theta_{sh}$" shows the shared parameters in the shared layers and τ_i is the task i. Thus, given a set of data samples X, the total loss function of the MTL for scene prediction is calculated by Eq. 2.

$$L_{all}(X, \theta_{sh}, \theta_{i=1,..,m}) = \sum_{i=1}^{m} L(\mathcal{T}_i; \theta_{sh}, \theta_i) \tag{2}$$

3.2 Training Hierarchical Multi-task Learning

Given the shared parameters θ_{sh} from the base model and set of tasks defined in $\mathcal{T}$, we concatenate the output of each task to the non-linearity function of the next task placed in its specific layer. This concatenation allows transferring the knowledge of the high-level tasks to the low-level tasks during the course of training.

$$L_{all}(X, \theta_{sh}, \theta_{i=1,..,m}) = \sum_{i=1}^{m} L(\mathcal{T}_i; \theta_{sh}, \theta_i, e^{z_{\tau_{i-1}}}) \tag{3}$$

In Eq. 3, $e^{z_{i-1}}$ refers to the output of τ_{i-1}, which is concatenated to the τ_i at the latest stage in the specific layer. The knowledge transferred between these tasks in the latest stage supports increasing the predictive model's performance.

3.3 Individual and Group Task Transference Quantification

To achieve perspicuity into the influence of the task (scene level) on each other without and with a hierarchical structure, as well as a group of tasks onto a specific task during the training process, we adapt the inter-task affinity (AF) to scene level impact, introduced in this study [15]. Hence, taking into account the deep multi-task learner parameterized by θ at the time t of training the scene levels without hierarchical structure, given a batch of $X_t \in DS$. We define the quantity $\theta^{t+1}_{sh|i}$ to indicate the model with the updated shared parameters towards the task τ_i.

$$\theta^{t+1}_{sh|i} = \theta^t_{sh} - \zeta \Delta_{\theta^t_{sh}} L_{\tau_i}(X^t, \theta^t_i, \theta^t_{\tau_i}) \tag{4}$$

$$IM_{\tau_i \to \tau_j} = 1 - \frac{L_{\tau_j}(X^t, \theta^{t+1}_{s|i}, \theta_j)}{L_{\tau_j}(X^t, \theta^t_s, \theta_j)} \cdot W_c \tag{5}$$

Using $\theta^{t+1}_{sh|i}$ in Eq. 4, we could measure the impact of task τ_i on the performance of the other individual tasks defined in $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_m\}$ in Eq. 5.

$$\theta^{t+1}_{sh|(S_\tau)} = \theta^t_{sh} - \zeta \Delta_{\theta^t_{sh}} L_{S_\tau}(X^t, \theta^t_i, \theta^t_{S_\tau}) \tag{6}$$

By extending Eq. 4 and Eq. 5, into Eq. 6 and Eq. 7 –respectively–, we are capable of calculating the impact of a group of tasks τ_S on the performance of the other individual task represented in $\mathcal{T}$. Thus, given the input $X^t \in DS$ we can measure the loss for each task by taking the updated shared parameter θ_{sh} and the specific task parameters θ_i. Indeed, we assess the impact IM of the gradient update of task τ_i or set of tasks S_τ on a given task τ_j for the individual transference. For the group transference, we assess the impact IM of the gradient update of the set of tasks (S_τ), which in our case are chosen to be tuple, triple, and quadruple, on a specific, e.g., task τ_k. Then, we can compare the ratio between the average loss value of tasks in a given set (τ_k) before and after conducting the gradient update from a set of tasks S_τ towards the shared parameters as follows:

$$IM_{S_\tau \to \tau_i} = 1 - \frac{L_{\tau_i}(X^t, \theta^{t+1}_{sh|(S_\tau)}, \theta_i)}{L_{\tau_i}(X^t, \theta^t_{sh}, \theta_i)} \cdot W_c \tag{7}$$

$$W_c = \frac{t}{T} \quad t = 1, 2, \ldots, T \tag{8}$$

where $\theta_{S\tau}$ represent parameters of the tasks in a given set. Thus, we translate $IM_{LS\tau \to \tau_i}$ as a measure of transference from meta-train individual or set of tasks S_τ to task τ_i. A positive value of $IM_{LS\tau \to \tau_i}$ shows the update on the

shard parameters θ_{sh} led to a lower loss value on the set of tasks with respect to the original parameters. This basically expresses the positive effect of task i (or S_τ) to generalize the predictive model on the individual or set of tasks (S_τ), while the negative value (if any) of S_τ describes the destructive impact of task i on that individual/set over the training process. We adapted the overall inter-task affinity measure onto the individual from the group of tasks by incrementally adding certain weights, defined in Eq. 8 on each iteration over the training process. t refers to the current epoch number, and T is the maximum number of epochs that the multi-task network should be iterated. This is because at the beginning of the training, the weights are randomly generated, so it is not expected the earlier loss have the same impact w.r.t to the parameters at the end of the training.

Thus, in Eq. 9, we calculate the transference over all epochs from task i to task j, and extend to Eq. 10 to measure the transference from a group of tasks to a specific task (e.g., task i).

$$\hat{IM}_{\tau_i \to \tau_j} = \frac{1}{T} \sum_{e=1}^{T} IM^e_{\tau_i \to \tau_j} \tag{9}$$

$$\hat{IM}_{L_{S\tau} \to \tau_i} = \frac{1}{T} \sum_{e=1}^{T} IM^e_{L_{S\tau} \to \tau_i} \tag{10}$$

$\hat{IM}_{\tau_i \to \tau_j}$ and $\hat{IM}_{L_{S\tau} \to \tau_i}$ can be conducted in different levels of granularity, such as per-epoch level or even micro-level/batch level. In this study, we measure the transference between individual tasks and groups of tasks onto individual tasks (Eq. 5) at the epoch level, where T refers to the number of epochs. Given the above problem formulation, we could then quantify how much a high-level scene can transfer the knowledge to a lower level in a later stage by concatenating the output of each scene to the lower level using Eq. 11.

$$IM_{S_\tau \to \tau_i} = 1 - \frac{L_{\tau_i}(X^t, \theta^{t+1}_{sh|(S_\tau)}, \theta_i, , e^{z_{\tau_{i-1}}})}{L_{\tau_i}(X^t, \theta^t_{sh}, \theta_i, , e^{z_{\tau_{i-1}}})} \cdot W_c \tag{11}$$

4 Experimental Evaluation and Results

The results and evaluations are presented with respect to the two research questions introduced in Sect. 1.

RQ1) To what extent could the task of scene classification be predicted using the Hierarchical Transfer Multi-tasks approach?; RQ2) To what extent could the scene transference be quantified?

4.1 RQ1: Hierarchical Multi-task Learning Results

To answer RQ1, we conducted several experiments. First, we build multiple predictive models for each task/scene level single task (STL) to quantify the

predictive model's performance. Then, we designed a multi-task network (MTL) wherein we simultaneously trained all the tasks in one single training process, having the shared and their specific parameters using Eq. 2. Thus the backbone of STL and MTL was trained from scratch on the Places365 subset. Indeed, these experiments will show whether we could gain any performance by formulating the task as multi-task learning. In addition, these practices have been accomplished to have baselines – and assess our proposed approach.

The second round of experiments was accomplished using the transfer multi-task learning (TMTL) and our proposed Hierarchical Transfer and Multi-task Learning approach (HTMTL) and compared with the baselines and other approaches to assess how much we could gain with the hierarchical transference between the tasks.

Table 1. The comparison between HTMTL, and different approaches for RQ1. For each model, 5×2 cv paired t-test was used to test the pairwise significance between the model and other models for each task. "**" refers to the alpha level at 0.05 to reject the null hypothesis, e.g., the "two sigma" level. Significant differences are denoted by ✓, and insignificant differences are denoted by $\times$ ✗. "TSTL-T1" refers to the single scene prediction, where the model is constructed based on transfer learning and only a specific layer. A similar construction has been used for other individual tasks.

Methods	Task1		Task 2		Task 3		Task 4		Task 5		Avg
	f-score	t-test **	f-score	t-test **	f-score	t-test **	f-score	t-test **	f-score	t-test **	f-score
STL-T1	0.62 ± 0.05	8.71 ✓	– – –	– – –	– – –	– – –	– – –	– – –	– – –	—	0.62
STL-T2	– – –	– – –	0.72 ± 0.2	15.29 ✓	– – –	– – –	– – –	– – –	– – –	—	0.72
STL-T3	– – –	– – –	– – –	– – –	0.85 ± 0.01	9.366 ✓	– – –	– – –	– – –	—	0.85
STL-T4	– – –	– – –	– – –	– – –	– – –	– – –	0.56 ± 0.04	−11.82 ✓	– – –	—	0.56
STL-T5	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	0.45 ± 0.01	46.90 ✓	0.45
MTL	$\mathbf{0.91 \pm 0.035}$	−3.41 ✓	0.41 ± 0.021	51.30 ✓	0.45 ± 0.03	27.86 ✓	0.16 ± 0.021	60.83 ✓	$\mathbf{0.88 \pm 0.03}$	7.31 ✓	0.57
TSTL-T1	0.78 ± 0.003	−19.5 ✓	– – –	– – –	– – –	– – –	– – –	– – –	– – –	—	0.78
TSTL-T2	– – –	– – –	0.91 ± 0.01	2.284 ✓	– – –	– – –	– – –	– – –	– – –	—	0.91
TSTL-T3	– – –	– – –	– – –	– – –	0.92 ± 0.006	0.99 ✗	– – –	– – –	– – –	—	0.92
TSTL-T4	– – –	– – –	– – –	– – –	– – –	– – –	0.80 ± 0.01	2.347 ✓	– – –	—	0.80
TSTL-T5	– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –	0.71 ± 0.01	6.793 ✓	0.71
TMTL	0.83 ± 0.014	3.438 ✓	0.92 ± 0.017	0.996 ✗	0.92 ± 0.21	0.87 ✗	0.76 ± 0.027	4.766 ✓	0.73 ± 0.019	4.079 ✓	0.83
HTMTL	0.86 ± 0.010	– – –	$\mathbf{0.93 \pm 0.007}$	– – –	$\mathbf{0.93 \pm 0.011}$	– – – –	$\mathbf{0.83 \pm 0.01}$	– – –	0.77 ± 0.004	—	**0.86**

Table 1 shows the results obtained by Eq. 2 for TMTL and Eq. 3 for HTMTL and compare them with single and other models. We have iterated the training and testing phases 5 times to get a reliable result and reported the overall average.

Results obtained from the models trained for the single tasks (STL) show that single Task3 (STL-T3) gave the highest f-score of the single tasks, with an f-score of 0.85. Utilizing MTL we could obtain an average f-score value of 0.57, where the shared layers were trained only on the Places365 dataset. In MTL, Task2, Task3, and Task4 got a lower f-score than their single-task counterparts (STL-2, STL-3, STL-4). However, there was an improvement of 0.29 in Task 1, and an improvement of 0.43 for Task 5, compared to their single-task models (STL-T1 and STL-T5), respectively.

Considering the average f-scores of 0.83 and 0.86 obtained by TMTL and HTMTL, respectively, we could see a considerable improvement when transfer learning and hierarchical structure were used - compared to MTL. The improvement can also be seen for the TSTL for each individual task, where we utilized the shared parameters from InceptionV3 trained on ImageNet and the specific layer trained on Places365.

Comparing TMTL and HTMTL, HTMTL gains improvement in all tasks by 0.3, 0.1, 0.1, 0.7, and 0.4 for Task 1 to Task 5, respectively, as well as 0.3 improvement of F-score in average, see Table 1.

As a part of the experiment, we conducted the statistical t-test and compared the results obtained for HTMTL and all other experiments to quantify whether the results differed significantly. Selecting $\alpha = 0.05$ as the critical value, we could see in most cases, the test could reject the null hypothesis and conclude that the proposed approach performed best on average and on the majority of the tasks. However, the test fails to reject the null hypothesis where Task2 and Task3 express that the small difference in individual task comparison between HTMTL and TMTL is not significant. It is also fair to highlight that MTL performed well for Task1 and Task5 compared to TMTL and HTMTL structures. This might be because MTL was trained from scratch on Places365, where TMTL and HTMTL used a pretrained backbone as shared parameters for Task1 and Task5. However, the average f-score obtained by TMTL and HTMTL indicates the proposed approach's superiority.

Table 2 provides a comprehensive comparison between the HTMTL model and other studies that utilized the Places365 dataset. As indicated in the table, the proposed HTMTL approach significantly outperforms the existing methods, achieving an impressive accuracy of 73.2%. This notable improvement highlights the effectiveness and robustness of the HTMTL model in handling the complex features and diverse categories present in the Places365 dataset. The enhanced performance can be attributed to the innovative techniques and optimizations incorporated into the HTMTL framework, setting a new benchmark for future research in this domain.

Table 2. The comparison between HTMTL and other approaches tested on Places365 dataset.

Models	Method	Extra data	Accuracy
SWAG [28]	Weakly-supervised	Yes	60.7%
Hiera-H [25]	Vision transformer	Yes	60.6%
MAE [12]	Self-supervised	Yes	60.3%
WaveMix [14]	Wavelet transform	No	56.4%
InternImage [38]	Large CNN foundation	No	61.2%
MixMIM-L [18]	AutoEncoder	–	60.3%
OmniVec2 [30]	Transformer, Multi-task	–	65.1%
HTMTL (ours)	Multi-task, Hierarchical	No	73.2%

To assess the generality of our approach, we utilized the Cityscapes dataset, which includes multiple tasks. Table 3 presents the Intersection over Union (IoU) metrics obtained from our HTMTL model and compares these results with those of existing methodologies applied to the same dataset. The results indicate that both BiSeNet and HTMTL achieve comparable performance, significantly surpassing other current approaches. This demonstrates the robustness and effectiveness of our HTMTL model in handling diverse tasks within the Cityscapes dataset. In more detail, the Cityscapes dataset is known for its complexity, containing a variety of urban scenes with numerous object classes, making it a rigorous benchmark for evaluating multi-task learning models. Our HTMTL model's performance on this dataset underscores its capability to generalize well across different tasks. Specifically, the IoU metric, which measures the overlap between the predicted segmentation and the ground truth, is a critical indicator of the model's precision and accuracy.

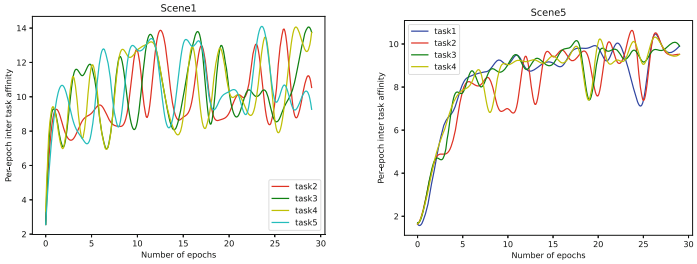
The superior performance of HTMTL and BiSeNet can be attributed to their advanced architectures, which effectively integrate spatial and contextual information. This integration allows for more accurate predictions across different object classes, contributing to the higher IoU scores observed. Consequently, the results from Table 3 validate our approach, highlighting its potential for application in real-world scenarios where multi-task learning is essential.

Table 3. The comparison between HTMTL and other approaches tested on Cityscapes dataset.

Models	Method	Extra data	IoU
MTL [26]	Multi-task, Multi-Objective	No	66.6%
Cross-CBAM [39]	Cross Convolutional Block Attention	No	73.4%
GUNet [20]	Guided Upsampling Network	Yes	70.4%
CAS-GT [43]	Directed Acyclic Graph	No	72.3%
BiSeNet [40]	Feature Fusion Module	No	74.7%
HTMTL (ours)	Multi-task, Hierarchical	No	73.6%

4.2 RQ2: Scene Transference Quantification Results

To answer RQ2, we carried out two kinds of implementations. The first refers to the individual task transference practice. We aim to quantify how much one task can positively or negatively affect another task over the training process by utilizing Eq. 5.



(a) Task1. (b) Task5.

Fig. 2. The task transference of different scene levels onto each other over the training process (Only Taks1 and Task5 are depicted due to the page limit).

Table 4. The inter-task affinity values obtained by Eq. 5. Each column shows the average effect of each task on other tasks over the 100 epochs using Eq. 9.

Task1	AF Value	Task2	AF Value	Task3	AF Value	Task4	AF Value	Task5	AF Value
T1 → T2	8.756	T2 → T1	9.359	T3 → T1	20.579	T4 → T1	8.807	T5 → T1	7.368
T1 → T3	9.193	T2 → T3	13.531	T3 → T2	21.339	T4 → T2	8.694	T5 → T2	6.999
T1 → T4	9.400	T2 → T4	12.328	T3 → T4	22.643	T4 → T3	9.294	T5 → T3	7.390
T1 → T5	9.359	T2 → T5	12.867	T3 → T5	18.458	T4 → T5	8.452	T5 → T4	7.334

Figure 2 shows the transference from individual tasks (or scene levels) to the performance of other tasks/scene levels during the 30 epochs (due to the page limit, we just plotted the transference onto two tasks; Task1 and Task5). The subplots illustrate the constructive impact of training on an individual task on the performance of another task. The two sub-plots show that the individual tasks at the beginning have less impact on other tasks, but the impact increases over the training time. Concerning the transference onto Task1 and Task5 (see Fig. 2a and Fig. 2b), which is the highest and lowest scene levels, we observed that over the 30 epoch training process, the impact on Task1 is more than on Task5 (the more AF value the more impact). This suggests that lower-level scenes can have a more positive effect on higher-level scenes when they are trained together in a multi-task fashion. The interaction between tasks and Task1 (depicted in Fig. 2a) fluctuated a lot, showing that weights are not well trained over the first 30 epochs, while interaction stabilized much faster for Task5 with a lower interaction value. This might be due to the nature of the scene levels to transfer the feature representation that positively supports the next level/task classification performance, which takes a bit of time to train to stabilize its positive effect. Although we observed these fluctuations over the training process, the impact stabilized in the later stages due to the learning improvement onto all tasks.

Table 4 shows the overall impact of each individual task on others over 100 epochs. From the table, we could clearly observe the constructive impact of the tasks on each other over the training process. In addition, we acknowledged that

Table 5. The inter-task affinity values obtained by Eq. 11 with the hierarchical structure. Each column shows the average effect of each task on other tasks over the 100 epochs.

Task1	AF Value	Task2	AF Value	Task3	AF Value	Task4	AF Value	Task5	AF Value
T1 → T2	8.533	T2 → T1	15.594	T3 → T1	22.282	T4 → T1	9.395	T5 → T1	7.607
T1 → T3	9.103	T2 → T3	15.458	T3 → T2	23.965	T4 → T2	9.201	T5 → T2	7.523
T1 → T4	8.928	T2 → T4	15.220	T3 → T4	23.860	T4 → T3	9.562	T5 → T3	7.421
T1 → T5	8.783	T2 → T5	15.147	T3 → T5	22.892	T4 → T5	8.646	T5 → T4	7.622

tasks have different contributions to the performance of the other tasks. The figures illustrated that Task5 has contributed less (compared with others) to the performance of the other tasks (highlighted in orange). The low-level nature of the Task5 could be the reason for this less contribution within the other tasks. In contrast, AF values obtained for Task3 show how important the role of Task3 is in the other tasks' performance in the multi-task structure. A similar contribution has been captured by counting the hierarchical design into multi-task learning (see Table 5). This increase in the effect of each task on others also confirms the usability of the hierarchical structure in the multi-task predictive model.

Table 6. The combination of different tasks located in different groups that affect 'Task3'. The impacts of the groups on 'Task3' are illustrated in Fig. 3.

Group #	Individual Task in Each Group
G1	Task4 & Task5
G2	Task2 & Task4 & Task5
G3	Task2 & Task5
G4	Task1 & Task2 & Task4
G5	Task2 & Task4
G6	Task1 & Task4
G7	Task1 & Task2
G8	Task1 & Task4 & T5
G9	Task1 & Task2 & Task5
G10	Task1 & Task5
G11	Task1 & Task2 & Task4 & Task5

In the second evaluation in RQ2, we aim to understand and quantify the set of tasks (group of tasks) transference onto individual tasks or the group's contribution to the individual Task/scene level over the training process by utilizing the shared and specific layers. To acknowledge this, we utilized Eq. 6 and adapted to the below equations.

$$b = avg(L_{S_\tau}(X^t_{val}, X^t_{tr}, \theta^{t+1}_{s|i}, \theta_{S_\tau})) \tag{12}$$

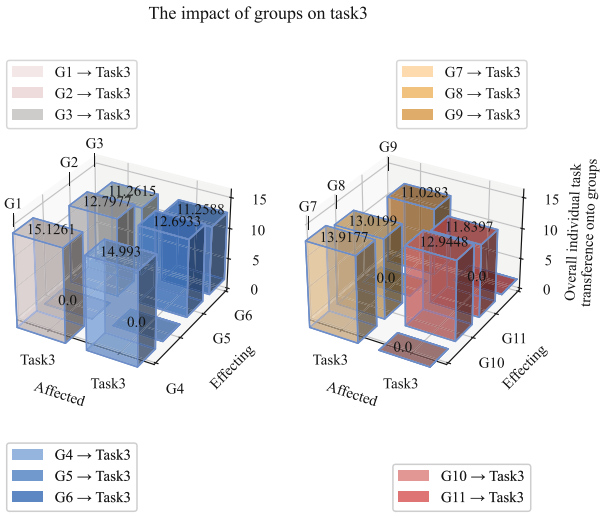


Fig. 3. The overall group knowledge transferred onto Task3 obtained by Eq. 6 and Eq. 10.

Table 7. The combination of different tasks located in different groups. The impacts of the groups on Task5 are illustrated in Fig. 4.

Group #	Individual Task in Each Group
G1	Task1 & Task4
G2	Task3 & Task4
G3	Task1 & Task2
G4	Task2 & Task4
G5	Task1 & Task3 & Task4
G6	Task1 & Task3
G7	Task1 & Task2 & Task3 & Task4
G8	Task2 & Task3
G9	Task1 & Task2 & Task3
G10	Task1 & Task2 & Task4
G11	Task2 & Task3 & Task4

$$a = avg(L_{\tau_j,\tau_k}(X^t_{val}, X^t_{tr}, \theta^t_s, \theta_{S_\tau}) \tag{13}$$

$$IM_{\tau_i \to L_{S_\tau}} = \frac{b-a}{b}.W_c \tag{14}$$

where X^t_{val} and X^t_{tr} are the independent feature predictors for training and validating, respectively. $avg(L_{S_\tau}(X^t_{val}, X^t_{tr}, \theta^{t+1}_{s|i}, \theta_{S_\tau}))$ refers to the average loss

value of tasks in the set S_τ after the network was updated and trained with the shared parameters for specific task τ_i ($\theta^{t+1}_{s|i}$), and $avg(L_{S_\tau}(X^t_{val}, X^t_{tr}, \theta^t_s, \theta_{S_\tau})$ points to the average loss value before the update. This intuitively reveals to what extent a group of tasks can transfer positive or even negative knowledge onto the performance of the individual tasks. This means different tasks might behave differently when they are in different groups.

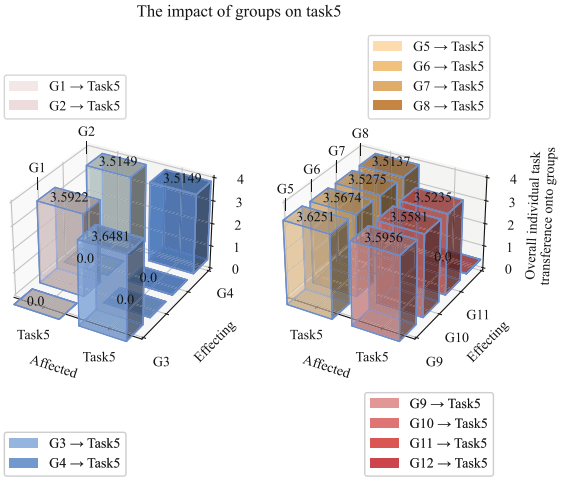


Fig. 4. The overall group knowledge transferred onto Task5 obtained by Eq. 6 and Eq. 10.

Considering the five different tasks, we could set up 11 different group combinations (we have removed groups with only one task) that can impact on each task. In total, we could generate 55 different combinations (groups of tasks) and measure their contributions to the performance of each task. For illustration purposes, we reported only two sets of combinations that can affect Task3 and Task5 in Table 6 and Table 7. The figures obtained from these groups are plotted in Fig. 3 and Fig. 4, and show all 11 groups, both for Task3 and Task5, have constructive contributions. We acknowledged that G1 is an optimal group with the most positive effect by $AF = 15.1$ on Task3 (highlighted in green), and G3 is the optimal group for Task5 by $AF = 3.62$. Given the numbers, we could then translate groups with two tasks as the optimal (or sub-optimal) group for the task grouping.

5 Ablation Studies

5.1 Contribution of Different Hierarchy and Base Model to Knowledge Transfer

Table 8 presents the results of an ablation study on HTMTL using the Cityscapes and Places365 datasets with different base models and hierarchical strategies. We

investigated the impact of top-down hierarchy (knowledge transfer from Task1 to Task2, Task2 to Task3, etc.) versus bottom-up hierarchy (knowledge transfer from Task8 to Task7, Task7 to Task6, etc.). In our experiments, we varied the baseline models and hyperparameters to thoroughly evaluate performance differences. The results indicate that the top-down hierarchy strategy consistently outperforms the bottom-up strategy. Interestingly, when we compared the performance of different base models, specifically ResNet and Xception, we found no significant difference. This suggests that both base models contribute similarly to the system's performance. Our findings imply that further improvements might be achieved by focusing on the specific layers within the network rather than the choice of the base model itself. Additionally, performance might improve with more training iterations, highlighting the potential for optimization through extended training and fine-tuning. Further investigation is needed to explore these aspects in more detail.

Table 8. Performance comparison of HTMTL on Cityscapes and Places365 with different base models and regularization components. V1 in HTMTL_V1 refers to the version of the HTMTL with different components detailed in the table.

Methods	Acc & std Cityscapes	Acc & std Places365
HTMTL_V1 ResNet101V2+up-to-down+30 epochs	**0.84±0.07**	0.72±0.02
HTMTL_V2 ResNet101V2+down-to-up+30 epochs	0.83±0.06	0.71±0.04
HTMTL_V3 Xception+up-to-down+30 epochs	0.83±0.02	**0.74±0.06**
HTMTL_V4 Xception+down-to-up+30 epochs	0.79±0.10	0.71±0.03

6 Conclusion

This study presents a hierarchical transfer multi-task deep neural network approach for multiple scene classification and quantifying knowledge transference among the scenes. The experimental evaluation of different levels of scene category classification illustrates a significant difference between the performance of the HTMLT approach and the baselines and other studies (see Table 2). Extensive experiments on quantifying the knowledge transferred between tasks and groups of tasks onto single tasks suggest a constructive contribution of the tasks on the performance of each other. Considering the overall impact and the results for individual tasks transference illustrated in plots and tables, it is evident that

high-level tasks can have a higher impact on the performance of the other tasks. In contrast, the low-level task/scene, or more specific tasks (e.g., Task5), has a less constructive impact on the performance of the higher-level scenes (see Fig. 2). This work motivated us, to extend our inter-task affinity approach to find the relationship between tasks (e.g., objects) to understand the scene, particularly in problematic conditions.

Acknowledgment. This study is supported by the'Synergy' project at Malmö University which was funded by the Knowledge Foundation in Sweden.

References

1. Alhichri, H.: Multitask classification of remote sensing scenes using deep neural networks. In: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 1195–1198. IEEE (2018)
2. Battiato, S., Curti, S., La Cascia, M., Tortora, M., Scordato, E.: Depth map generation by image classification. In: Three-Dimensional Image Capture and Applications VI, vol. 5302, pp. 95–104. SPIE (2004)
3. Chen, B.X., Sahdev, R., Wu, D., Zhao, X., Papagelis, M., Tsotsos, J.K.: Scene classification in indoor environments for robots using context based word embeddings. arXiv preprint arXiv:1908.06422 (2019)
4. Chen, S., Zhang, Y., Yang, Q.: Multi-task learning in natural language processing: an overview. arXiv preprint arXiv:2109.09138 (2021)
5. Chen, Y., Zhao, D., Lv, L., Zhang, Q.: Multi-task learning for dangerous object detection in autonomous driving. Inf. Sci. **432**, 559–571 (2018)
6. Collier, J., Ramirez-Serrano, A.: Environment classification for indoor/outdoor robotic mapping. In: 2009 Canadian Conference on Computer and Robot Vision, pp. 276–283. IEEE (2009)
7. Dandotiya, M., Dubey, M.: A VGG-16 framework for an efficient indoor-outdoor
8. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: ideas, influences, and trends of the new age. ACM Comput. Surv. (Csur) **40**(2), 1–60 (2008)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
10. Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., Finn, C.: Efficiently identifying task groupings for multi-task learning. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
11. Guo, Z., Huang, Y., Hu, X., Wei, H., Zhao, B.: A survey on deep learning based approaches for scene understanding in autonomous driving. Electronics **10**(4), 471 (2021)
12. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16000–16009 (2022)
13. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size (2016)
14. Jeevan, P., Viswanathan, K., Sethi, A.: Wavemix-lite: a resource-efficient neural network for image analysis. arXiv preprint arXiv:2205.14375 (2022)

15. Khoshkangini, R., Mashhadi, P., Tegnered, D., Lundström, J., Rögnvaldsson, T.: Predicting vehicle behavior using multi-task ensemble learning. Expert Syst. Appl. **212**, 118716 (2023)
16. Khoshkangini, R., Tajgardan, M., Lundström, J., Rabbani, M., Tegnered, D.: A snapshot-stacked ensemble and optimization approach for vehicle breakdown prediction. Sensors **23**(12) (2023). https://www.mdpi.com/1424-8220/23/12/5621
17. Khoshkangini, R., Tajgardan, M., Mashhadi, P., Rögnvaldsson, T., Tegnered, D.: Optimal task grouping approach in multitask learning. In: Luo, B., Cheng, L., Wu, Z.G., Li, H., Li, C. (eds.) Neural Information Processing, pp. 206–225. Springer, Singapore (2024)
18. Liu, J., Huang, X., Zheng, J., Liu, Y., Li, H.: Mixmae: mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6252–6261 (2023)
19. Lu, X., Li, X., Mou, L.: Semi-supervised multitask learning for scene recognition. IEEE Trans. Cybern. **45**(9), 1967–1976 (2014)
20. Mazzini, D.: Guided upsampling network for real-time semantic segmentation. arXiv preprint arXiv:1807.07466 (2018)
21. Miyahara, A., Shimabukuro, K., Nagayama, I.: A study on intelligent security camera for crime scene detection. In: Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications, vol. 2014, pp. 34–40. The ISCIE Symposium on Stochastic Systems Theory and Its Applications (2014)
22. Pereira, R., Barros, T., Garrote, L., Lopes, A., Nunes, U.J.: A deep learning-based global and segmentation-based semantic feature fusion approach for indoor scene classification. Pattern Recogn. Lett. **179**, 24–30 (2024)
23. Pironkov, G., Dupont, S., Dutoit, T.: Multi-task learning for speech recognition: an overview. In: ESANN (2016)
24. Ran, T., Yuan, L., Zhang, J.: Scene perception based visual navigation of mobile robot in indoor environment. ISA Trans. **109**, 389–400 (2021)
25. Ryali, C., et al.: Hiera: a hierarchical vision transformer without the bells-and-whistles. arXiv preprint arXiv:2306.00989 (2023)
26. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
27. Shahriari, M., Bergevin, R.: A two-stage outdoor-indoor scene classification framework: experimental study for the outdoor stage. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8. IEEE (2016)
28. Singh, M., et al.: Revisiting weakly supervised pre-training of visual perception models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 804–814 (2022)
29. Song, X., et al.: Efficient and effective multi-task grouping via meta learning on task combinations. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems, vol. 35, pp. 37647–37659. Curran Associates, Inc. (2022)
30. Srivastava, S., Sharma, G.: Omnivec2 - a novel transformer based network for large scale multimodal and multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 27412–27424 (2024)

31. Standley, T., Zamir, A., Chen, D., Guibas, L., Malik, J., Savarese, S.: Which tasks should be learned together in multi-task learning? In: International Conference on Machine Learning, pp. 9120–9132. PMLR (2020)
32. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
33. Szummer, M., Picard, R.W.: Indoor-outdoor image classification. In: Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database, pp. 42–51. IEEE (1998)
34. Tong, Z., Shi, D., Yan, B., Wei, J.: A review of indoor-outdoor scene classification. In: 2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017), pp. 469–474. Atlantis Press (2017)
35. Vailaya, A., Figueiredo, M.A., Jain, A.K., Zhang, H.J.: Image classification for content-based indexing. IEEE Trans. Image Process. **10**(1), 117–130 (2001)
36. Venkataramanan, A., Laviale, M., Figus, C., Usseglio-Polatera, P., Pradalier, C.: Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In: International Conference on Computer Vision Systems, pp. 93–103. Springer (2021)
37. Wang, C., Pan, X., Yu, T.: Towards principled task grouping for multi-task learning (2024). https://arxiv.org/abs/2402.15328
38. Wang, W., et al.: Internimage: exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14408–14419 (2023)
39. Xu, Z., Xiong, J., et al.: Cross-cbam: a lightweight network for scene segmentation
40. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer vision (ECCV), pp. 325–341 (2018)
41. Yu, Q., Anzawa, M., Amano, S., Ogawa, M., Aizawa, K.: Food image recognition by personalized classifier. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 171–175. IEEE (2018)
42. Zeng, D., et al.: Deep learning for scene classification: a survey. arXiv preprint arXiv:2101.10531 (2021)
43. Zhang, Y., Qiu, Z., Liu, J., Yao, T., Liu, D., Mei, T.: Customizable architecture search for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11641–11650 (2019)
44. Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Trans. Knowl. Data Eng. (2021)
45. Zheng, Y., Gao, X.: Indoor scene recognition via multi-task metric multi-kernel learning from RGB-D images. Multimedia Tools Appl. **76**, 4427–4443 (2017)
46. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. (2017)
47. Zhu, B., Fan, X., Gao, X., Xu, G., Xie, J.: A heterogeneous attention fusion mechanism for the cross-environment scene classification of the home service robot. Robot. Auton. Syst. **173**, 104619 (2024)

Deep Prompt Multi-task Network for Abuse Language Detection

Jian Zhu, Yuping Ruan, Jingfei Chang, Wenhui Sun, Hui Wan, Jian Long, and Cheng Luo(✉)

Zhejiang Lab, Hangzhou, China
{qijian.zhu,ypruan,cjf_chang,sunwh,wanhui,longjian,
luo_cheng}@zhejianglab.com

Abstract. The detection of abuse language remains a long-standing challenge with the extensive use of social networks. The detection task of abuse language suffers from limited accuracy. We argue that the existing detection methods utilize the fine-tuning technique of the pre-trained language models (PLMs) to handle downstream tasks. Hence, these methods fail to stimulate the general knowledge of the PLMs. To address the problem, we propose a novel Deep Prompt Multi-task Network (DPMN) for abuse language detection. Specifically, DPMN first attempts to design two forms of deep prompt tuning and light prompt tuning for the PLMs. The effects of different prompt lengths, tuning strategies, and prompt initialization methods on detecting abuse language are studied. In addition, we propose a Task Head based on Bi-LSTM and FFN, which can be used as a short text classifier. Eventually, DPMN utilizes multi-task learning to improve detection metrics further. The multi-task network has the function of transferring effective knowledge. The proposed DPMN is evaluated against eight typical methods on three public datasets: OLID, SOLID, and AbuseAnalyzer. The experimental results show that our DPMN outperforms the state-of-the-art methods.

Keywords: Abuse Language Detection · Prompt-based Learning · Deep Prompt Tuning · Multi-task Network

1 Introduction

1.1 Background

The abuse language has spread throughout social media and become a significant issue. On social network sites like Facebook, Twitter, and Instagram, some groups become targets of online bullying activities. Any expression that denigrates or offends a person or group of people is referred to as abuse, and examples include sexism, harassment, cyberbullying, personal insults, racism, etc. Abuse

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 249–263, 2025.
https://doi.org/10.1007/978-3-031-78107-0_16

language can be directed at particular people or groups. Abuse language can have serious psychological consequences for the victim and hinder freedom of expression. Intelligent detection algorithms can identify abuse content in a significant volume of social media. It is essential to minimize the psychological toll on victims to stop hate crimes. As a result, it is important to intelligently detect and govern abuse language before it spreads on social networks. Past research has examined various abuse language issues, including abuse and hate speech.

Abuse language detection can be seen as short text classification. As natural language processing evolves, detecting abuse language can be roughly divided into three periods. Early detection algorithms adopt conventional machine learning methods, and the performance of model detection results largely depends on the features of manual design. These hand-designed features mainly include character features, word features, n-gram features, syntactic features, and linguistic features. The second phase of the detection algorithm uses the deep learning method. Typical deep networks are CNN [1], and RNN [2]. The advantage of the deep learning method is that it does not need to design features manually. It can automatically generate context features of short text through the deep neural network. Third, the large PLMs, such as BERT [3] and GPT [4], improve the metric of detecting abuse language. Because these models are trained on a large-scale corpus, general knowledge of natural language can be obtained.

Due to the complicated of natural language laws, automatically identifying abuse language is still exceedingly challenging. For example, abuse language generally occurs in two cases, explicit and indirect linguistic phenomena. The earlier type of abuse language is more overt, perhaps taking the form of specific harsh phrases. However, the latter type could contain metaphors or analogies, which might cause certain algorithmic identification mistakes. In addition, the existing methods focus on fine-tuning the PLMs to adapt to the downstream tasks so that the training and application of the PLMs are not under a unified paradigm, and the knowledge contained in the PLMs cannot be better utilized. Therefore, the primary goal of our work is to do this research: whether prompt-based learning helps detect abuse language. To this end, we propose a new end-to-end multi-task detection network for abuse language, which combines prompt tuning and multi-task learning.

1.2 Motivations and Contributions

The task of abuse language detection suffers from limited accuracy. Current detection methods [5–9] fine-tune the PLMs to adapt to downstream tasks. NULI [5] adapts and fine-tunes the BERT-base model to detect abuse language. AbuseAnalyzer [6] uses a two-layer feed-forward network with BERT for detecting abuse language. Kungfupanda [7] develops a method for detecting abuse language that blends multi-task learning with BERT-based models. With regard to their effectiveness in detecting abuse language, UHH-LT [8] uses the MLM method to compare the performance of different PLMs. An architecture called CTF [9] combines various textual elements to find abuse or hostile tweets on Twitter, which generates contextual 768-dimensional word vectors for each

input character using a pre-trained BERT model. Therefore, these methods fail to stimulate the general knowledge of the PLMs, leading to limited accuracy. To address the mentioned problem, we introduce prompt-based learning and explore how prompt-based learning can be correctly used in detecting abuse language.

Prompt tuning has been a great success for most natural language processing tasks. By including new texts in the input, prompt-based learning is a method for better using the knowledge from the PLMs. As shown in Fig. 1, a prompt with a mask token is added to the tweet text, and we predict that the mask would point to the corresponding word in the vocabulary through PLMs. Then, based on the corresponding label of the word, it can determine whether the tweet is abuse language.

We propose a novel *Deep Prompt Multi-task Network* termed DPMN. First, it uses prompt-based learning to acquire knowledge of the PLMs. Second, to use the supervisory signals from other related tasks, we employ multi-task learning. Eventually, we design a task head based on the synthesis of Bi-LSTM [10], and feed-forward network (FFN) [11] to aggregate all the shared representations of the final output layer of the BERT model.

Experimental results indicate that deep prompt tuning is a very effective method. Specifically, the Macro F1 scores of DPMN are 0.8384, 0.9218, and 0.8165 on the OLID, SOLID, and AbuseAnalyzer datasets. We prove that the proposed DPMN achieves excellent results in detecting abuse language.

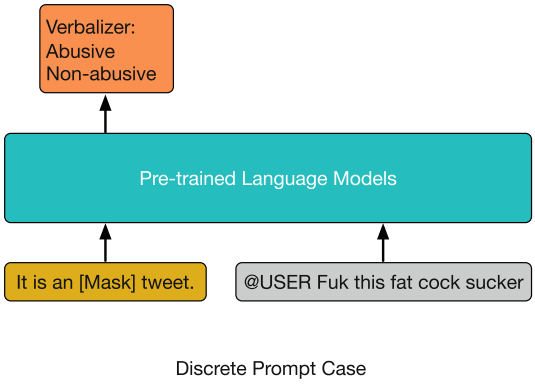


Fig. 1. An example of applying prompt-based learning to identify whether a tweet is an abuse language. The mask character is a word to be predicted. It can have two-word choices in the verbalizer, abuse and non-abuse.

Our main contributions are as follows:

- We propose a novel *Deep Prompt Multi-task Network*, which achieves state-of-the-art results in detecting abuse language.
- Deep prompt tuning is first applied to abuse language detection. To improve the metric of detecting abuse language, we compare the forms of deep prompt

tuning and light prompt tuning. Then the effects of different prompt lengths, tuning strategies, and prompt initialization methods are studied.
– We present a task head based on Bi-LSTM and FFN, and we prove that the effect of the task head is very significant through experiments.

2 Related Works

2.1 Abuse Language Detection

Much past work has explored the problem of detecting different types of abuse language. On the whole, the detection of abuse language mainly includes the following types, aggression detection [12], bullying recognition [13], hate speech recognition [14], abuse language, and toxic comments.

– Aggression detection: The developer can access a dataset of 15,000 tagged Facebook short texts as part of the shared challenge on aggression identification [12] in TRAC-2018. It is applied to the model training and validation. In the performance test portion of the model, there are two distinct datasets used, one from Twitter and the other from Facebook. The detection task aims to distinguish three categories: non-aggressive, covertly aggressive, and overtly aggressive.
– Bullying recognition: There are currently several works on cyberbullying detection methods. For example, Jun-Ming Xu [15] uses text classification, role labeling, sentiment recognition, and LDA to recognize related topics.
– Hate speech recognition: Hate speech detection tasks have a long history of research [16–18]. Davidson [19] proposes a dataset for detecting hate speech that includes more than 24,000 tweets in English.
– offensive language: LSF [20] applies concepts from the theory of natural language processing to exploit the linguistic syntactic representations of sentences to detect offensive language. Zampieri [21] introduces an offensive language recognition dataset OLID, which seeks to identify the class and the objective of offensive content in social networks. Rosenthal [22] extend the OLID into the multilingual edition SOLID, which promotes multilingual research in detecting abuse language. MTL [7] uses multi-task learning and the BERT-base model to detect offensive language.
– Toxic comments: On Kaggle, there is a free contest called the Toxic Comment Classification Challenge. It provides the developer with short comments from Wikipedia. The dataset is divided into six groups: insult, obscene, threat, toxic, severe toxic, and identity hate. Through thorough trials on prompt engineering, Generative Cls [23] investigates the generative variation of zero-shot prompt-based toxicity detection.

Although each task involves specific types of abuse or offense, many things are in common. For instance, insults against individuals are often called cyberbullying, and insults against groups are called hate speech.

MTL is the baseline model of our network. Compared with the MTL model, we first add prompt-based learning. Secondly, the task head is optimized. Our

structure is Bi-LSTM + FFN, which is simpler and more effective than the LSTM + FFN + Softmax of MTL.

In contrast to Generative Cls, DPMN does not need to design manual prompts. Making a good prompt is very time-consuming and tough. In general, artificial design is not an intelligent solution.

2.2 Prompt-Based Learning for PLMs

To improve the output embedding from the PLMs, prompt-based learning entails adding instructions to the input text. With the development of GPT-3 [24], prompt-based techniques have excelled in many common natural language processing applications. Many researches [25–30] have been put forth to show how prompt-based learning has advanced by the appropriate manual prompt. Knowledgeable prompt-tuning [31] suggests calibrating the verbalizer to accommodate outside knowledge. Automatic generating for discrete prompt has been thoroughly investigated as a way to prevent time-consuming prompt design. LM-BFF [32] first explores the creation of label words and templates automatically. Additionally, Autoprompt [33] suggests using gradient-guided search to create the vocabulary template and identify terms automatically. Continuous prompts have recently been proposed [34–37], which emphasize the use of learnable continuous representations rather than label words as prompt templates. In a word, prompt-based learning is applied in natural language processing to improve the understanding and generation of PLMs.

In view of the rapid rise and development of the above prompt tuning, we first design two continuous prompt forms, namely, deep prompt tuning and light prompt tuning. Then we apply the two prompt forms to abuse language detection. Eventually, we propose a network termed DPMN, which combines prompt-based learning and multi-task learning.

3 The Proposed Methodology

Our DPMN architecture is shown in Fig. 2. The three sub-tasks share PLMs in the section at the bottom. Each sub-task has its unique module in the upper parts. A task head based on the Bi-LSTM and FFN neural network topology is present in each module. DPMN sets the number, form, and initialization strategy of continuous prompt tokens. Then it generates the learnable embedding through the prompt encoder module. The tokenizer encoder module encodes the short text to generate input embedding. Splicing it with the input embedding generated by the PLMs. DPMN inputs them into the PLMs. The task head obtains the shared representations produced by the PLMs and generates a predicted category for short text. We calculate the multi-task loss function and train the entire architecture.

3.1 Continuous Prompt Tuning Parameters

Prompts can be divided into Discrete Prompts and Continuous Prompts.

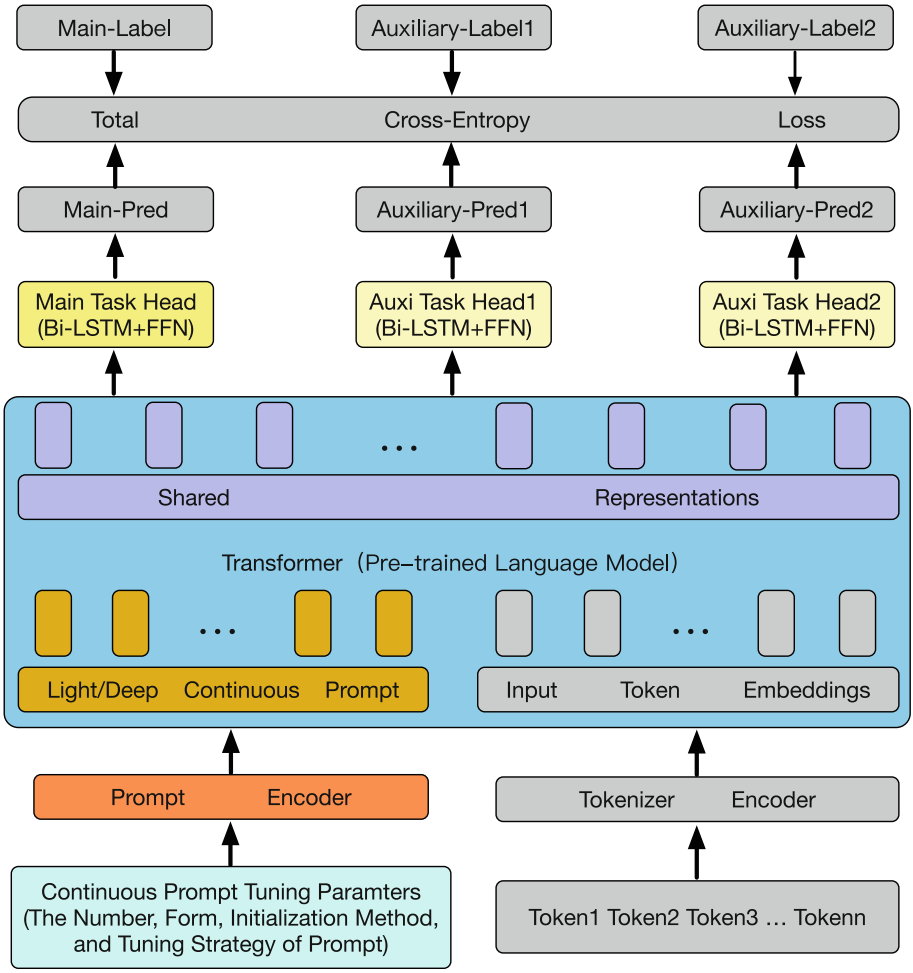


Fig. 2. Our DPMN Architecture. According to the number of continuous prompt tokens, initialization strategy of continuous prompt, prompt form, and tuning strategy, the prompt encoder module generates the learnable embedding. The tokenizer encoder module encodes the short text to generate input embedding. We produce a representation matrix by combining the learnable embedding and input embedding. The representation matrix is input into PLMs. It outputs the shared embedding. The shared embedding is input to the task heads. The task heads output the probability value of the prediction classification. The total loss function is calculated to train the entire DPMN architecture.

– **Discrete Prompts.** Discrete prompts are essentially a natural language. Based on fixed prompt word templates, that is, adding fixed prompt word templates and masking words after inputting statements.
– **Continuous Prompts.** Continuous prompts are no longer natural language. It replaces the fixed prompt word template with n learnable vectors.

Continuous prompt tuning parameters are made up of the number, form, initialization method, and tuning strategy of prompt tokens.

The number of prompt tokens plays a critical role in DPMN. We verify that abuse language detection usually achieves different performances with different prompt lengths. Specifically, we choose the number of prompt tokens according to the metric of detecting abuse language.

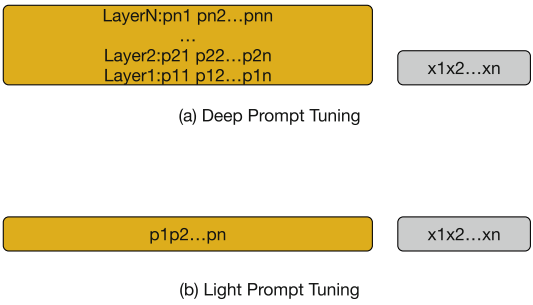


Fig. 3. We design two continuous prompt forms, namely deep prompt tuning and light prompt tuning. The deep prompt tuning is to add trainable continuous prompt embedding to each layer of the PLMs. The light prompt tuning is to add trainable continuous prompt embedding to the first layer of the PLMs.

Considering the instability of discrete prompt performance, we adopt the form of continuous prompt tuning. Because the template of a discrete prompt requires a lot of manual design work, the predicted performance of the prompt-based model is relatively volatile. As shown in Fig. 3, we use two continuous prompt forms: deep prompt tuning and light prompt tuning. We prove the effectiveness of these two prompt forms through experiments. The deep prompt tuning is to add trainable continuous prompt embedding to each layer of the PLMs. The light prompt tuning is to add trainable continuous prompt embedding to the first layer of the PLMs. For two continuous prompt forms, we study the most suitable prompt form.

In the architecture of our network, prompt initiation is a significant research challenge. It has two kinds of parameter initialization methods, which are random parameter initialization and BERT token initialization.

Tuning strategy is also an important research issue in our network design. It contains the two strategies of the Fixed LM Prompt Tuning strategy and LM + Prompt Tuning strategy.

3.2 Prompt Encoder

According to the number, form, initialization method, and tuning strategy of the prompt, the prompt encoder module generates a continuous prompt. Specifically, first define n trainable embedding vectors, and then initialize them. Secondly, the input text is generated into a word ID sequence through a word splitter, and then an embedding sequence is generated based on the BERT model vocabulary. Finally, we concatenate these two vectors as inputs to the BERT model.

The PLMs are used as the backbone for our DPMN. The continuous prompt is applied to the encoder of the PLMs. To prepend continuous prompts for the encoder, DPMN initializes a trainable prefix matrix p_e of dimension $p_n \cdot d$ for each layer of the PLMs, where p_n is the prompt length, and d is the hidden size of the PLMs. Because the prompt is located in the deep layers of the PLMs, it has the total capacity to guide the PLMs in the right direction and output a semantic shared representation for abuse language detection.

The continuous prompt stimulates the general knowledge of the PLMs. It performs better than fine-tuning in a range of natural language processing applications. The learnable embeddings are trained for abuse language detection.

3.3 Task Head

The central for prompt tuning is that the PLMs use a head to predict verbalizers. Because this requires manual design and even searches for all possible words, which is very labor-intensive. Therefore, the DPMN directly replaces the design of the verbalizer module with the classified label. The task head of the DPMN adopts the neural network architecture of Bi-LSTM + FFN. Compared to the linear classification head, its predicted performance is better. LSTM comprises an input gate, forget gate, output gate, and cell state. Bi-LSTM contained two sub-networks to model a text sequence in both directions. The outputs of short text are integrated in the following way:

$$H_{F,B} = [F\_H_t, B\_H_t], \tag{1}$$

where F_H_t is output value of the LSTM at the last time t in forward direction. B_H_t represents the output value of the LSTM at the last time t in the backward direction. Here $H_{F,B}$ is the output embedding of the Bi-LSTM result.

FFN is made up of two linear layers. Its activation function is ReLU.

$$y = W_{f2} ReLU(W_{f1} H_{F,B} + b_{f1}) + b_{f2}, \tag{2}$$

we let $\Theta_{FFN} = \{W_{f1}, W_{f2}, b_{f1}, b_{f2}\}$, where Θ_{FFN} is the learnable FFN parameter. The task head takes the feed-forward network as the classifier. Multi-task classification is achieved by setting different output numbers of PLMs.

3.4 Multi-task Network

The proposed DPMN is a multi-task network, which is divided into the main task and two auxiliary tasks. Our main task is to detect whether the text is

abuse language. The auxiliary tasks are to improve the feature representation ability of the output layer of the PLMs, thereby improving the detection ability of the main task and playing the role of transfer learning.

We train our DPMN on train datasets and mainly verify the model performance metrics on sub-task A. The goal of multi-task learning is to deliver useful information in tasks B and C to boost task A.

3.5 DPMN Loss

$$Loss_{sub-task} = -\sum_{i} y_i \log\left(y_i'\right) \tag{3}$$

y_i' is the probability predicted by the proposed DPMN. y_i is the category information of the dataset. $Loss_{sub-task}$ represents the loss of the sub-task, using the cross-entropy loss function.

$$c_{main} + c_{auxi1} + c_{auxi2} = 1 \tag{4}$$

c_{main} is the loss coefficient for the main task. c_{auxi1} is the loss coefficient for the auxiliary task. We set the sum of the coefficients of all current sub-task losses to 1.

$$\begin{aligned} Loss_{\text{total}} = c_{main} Loss_{main} + c_{auxi1} Loss_{auxi1} \\ + c_{auxi2} Loss_{auxi2} \end{aligned} \tag{5}$$

$Loss_{total}$ is the total loss of the DPMN. It is equal to the weighted sum of the losses of each sub-task.

4 Experiments

Table 1. Four short tweets from the OLID, their corresponding labels are hierarchical. In task A, the aim is to discriminate between offensive and non-offensive posts. In task B, the goal is to predict the type of offense: Targeted Insult (TIN) and Untargeted (UNT). Task C focuses on the target of offenses: Individual (IND), Group (GRP), and Other (OTH).

Short Text	Task A	Task B	Task C
@USER With his offers, he is extremely kind.	NOT	—	—
Liberated! THE WORST ACTIVITY OF MY FUCKING LIFE	OFF	UNT	—
@USER This big cocksucker is fucked	OFF	TIN	IND
@USER Figures! Why are these people such idiots? Praise God for @USER	OFF	TIN	GRP

We assess the proposed DPMN for detecting abuse language in experiments. Three public datasets are adopted: the OLID [38], SOLID [22], and AbuseAnalyzer dataset [39]. These datasets have been widely used for evaluating detection metrics of abuse language. Macro F1 score is used as the evaluation metric. DPMN achieves excellent performance in abuse language detection (Table 1).

Table 2. Detecting abuse language results on three public datasets, the evaluation metric is the Macro F1 score.

Dataset	Model	Main Task	Tuning Strategy	Prompt Length	Prompt Initialization
OLID	Logistic Regression	0.7501	*	*	*
	Bagging	0.7558	*	*	*
	MTL	0.8244	LM Tuning	*	*
	NULI	0.8290	LM Tuning	*	*
	DPMN(light prompt)	0.8279	LM + Prompt Tuning	1	Random
	DPMN(deep prompt)	**0.8384**	**LM + Prompt Tuning**	**1**	**BERT Token**
SOLID	MTL	0.9139	LM Tuning	*	*
	MTL(Ensemble)	0.9151	LM Tuning	*	*
	UHH-LT	0.9204	LM Tuning	*	*
	DPMN(light prompt)	0.9208	LM + Prompt Tuning	1	Random
	DPMN(deep prompt)	**0.9218**	**LM + Prompt Tuning**	**2**	**BERT Token**
AbuseAnalyzer	SVM	0.7277	*	*	*
	XGBoost	0.7157	*	*	*
	Logistic Regression	0.7235	*	*	*
	BERT + Linear Head	0.7985	LM Tuning	*	*
	DPMN(light prompt)	0.8107	LM + Prompt Tuning	1	BERT Token
	DPMN(deep prompt)	**0.8165**	**LM + Prompt Tuning**	**1**	**Random**

4.1 Baseline

To evaluate the detection metric, the proposed network DPMN is compared with eight comparable supervised methods, containing four shallow supervised methods (e.g., Logistic Regression [40], XGBoost [41], Bagging [42], SVM [43]) and four deep supervised methods (e.g., BERT + Linear Head [39], MTL [7], NULI [5], UHH-LT [8]).

4.2 Analysis of Experimental Results

The results in Table 2 show detection metrics. The proposed DPMN is overall better than all the compared methods in three public datasets. For example, compared with the current state-of-the-art method, the Macro F1 score of our DPMN has been increased by 0.94%, 0.14%, and 1.80% on the OLID, SOLID, and AbuseAnalyzer, respectively.

The main reasons for these superior results come from three aspects:

- Deep prompt tuning can better use the general knowledge of the PLMs.
- We propose an effective task head based on Bi-LSTM and FFN, and it improves the detection of abuse language.
- DPMN utilizes multi-task learning, which can obtain more useful information from the other tasks.

In addition, it can be seen that the detection effect of the deep models is better than that of the shallow models.

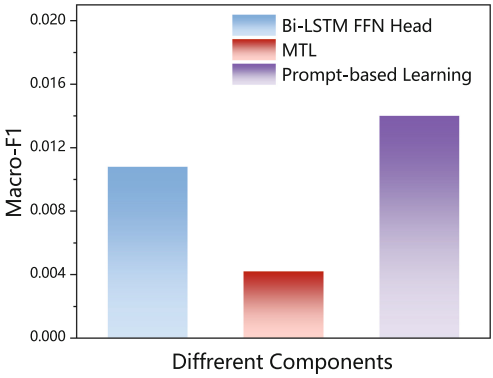


Fig. 4. Through ablation experiments on the OLID dataset, the Macro F1 score contributions of different neural network modules of the DPMN algorithm are determined. A histogram of the structural lifting value is drawn in the network, which contains the Bi-LSTM + FFN module, the multi-task learning (MTL) module, and the prompt-based learning module.

Table 3. The Ablation Experiments of DPMN Components.

Model	Architecture	Macro F1
BERT Base	BERT + Linear Head	0.8037
BERT LSTM	BERT + Bi-LSTM FFN Head	0.8202
DPMN-P	BERT + Bi-LSTM FFN Head + Multi-task Learning	0.8244
DPMN-M	BERT + Bi-LSTM FFN Head + Prompt	0.8342
DPMN-B	BERT + Multi-task Learning + Prompt	0.8276
DPMN	BERT + Bi-LSTM FFN Head + Multi-task Learning + Prompt	0.8384

4.2.1 Ablation Experiment

The results in Table 3 show the ablation experiment. We design the ablation experiment for the DPMN components. *BERT Base* model adopts the BERT + Linear Head structure, where the Linear Head is a classifier of multi-layer perceptron structure. The Macro F1 score of the *BERT Base* model to detect abuse language is 0.8037. The *BERT LSTM* model utilizes the BERT + Bi-LSTM FFN Head structure, where the Bi-LSTM FFN Head is a classifier of the Bi-LSTM + FFN structure. The Macro F1 score of the *BERT LSTM* model to detect abuse language is 0.8202. We experimentally prove that the classification head based on Bi-LSTM + FFN is better than the classification head based on Linear Head.

DPMN-P adds a multi-task learning architecture based on the *BERT LSTM* model. It removes the prompt tuning module compared to DPMN. The Macro F1 score of the *DPMN-P* model to detect abuse language is 0.8244. DPMN optimizes the *DPMN-P* model and designs the architecture of prompt-based learning. The Macro F1 score of the DPMN model to detect abuse language is

0.8384. Comparing the detection performance of *DPMN-P* and DPMN models, we can conclude that deep continuous prompt learning is effective.

In contrast to DPMN, *DPMN-M* gets rid of the multi-task learning module. Its Macro F1 score is 0.8342.

Compared to DPMN, *DPMN-B* does away with the Bi-LSTM + FFN Head. Its metric value is 0.8276.

To express the validity of the ablation experiment, we draw a histogram of the structural lifting value in the network. From the above Fig. 4, in the evaluation value of Macro F1, the contribution of the Bi-LSTM + FFN module is 0.0108, the contribution of the multi-task learning (MTL) module is 0.0042, and the contribution of the prompt-based learning module is 0.0140. Therefore, it can be seen that the proposed DPMN is effective.

4.2.2 The Convergence of DPMN

From Fig. 5, the training loss curve shows a downward trend with the increase in the number of epochs. As the number of epochs increases, the test loss curve first decreases and then increases. With the deepening of DPMN training, the test loss shows an upward trend, indicating that the detection performance of DPMN is declining. Through the changes in the two curves, it can be concluded that the DPMN has the best performance at $epoch = 5$.

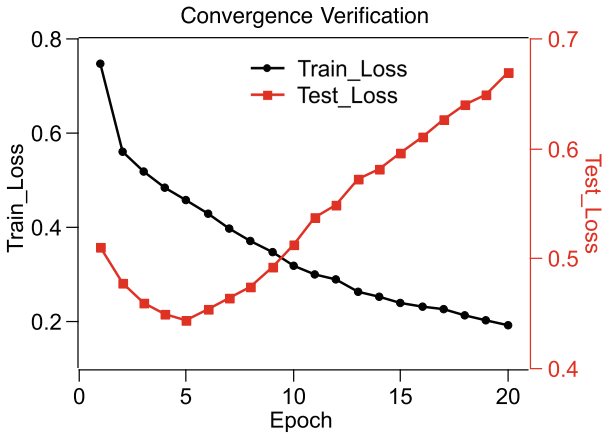


Fig. 5. The convergence of the DPMN is verified on the OLID dataset.

4.3 Implementation Details

We chose a learning rate of 3e–6 and a batch size of 32 for our best DPMN. The loss coefficients for sub-tasks A, B, and C are 0.4, 0.3, and 0.3, respectively. We

use an early stop method to stop tuning the model if the validation Macro F1 does not rise in four consecutive epochs. We train the DPMN with a maximum of 30 epochs. The DPMN is implemented in PyTorch, and a single GPU-V100 is used for each experiment.

5 Conclusion and Future Work

We propose an innovative *Deep Prompt Multi-task Network* termed DPMN. It introduces deep prompt tuning in abuse language detection for the first time. It can better motivate the knowledge of PLMs. We design a task head based on Bi-LSTM and FFN, which improves the performance in detecting abuse language. We attempt two prompt forms and verify the effects of different prompt lengths, tuning strategies, and prompt initialization methods. The proposed DPMN achieves state-of-the-art results in three abuse datasets. The follow-up work is to optimize and adaptively adjust the sub-task loss weight in multi-tasks and reasonably design the algorithm of the whole model loss function.

Acknowledgment. This work was supported by the National Key Research and Development Program of China (Grant No. 2021ZD0201501), the Youth Foundation Project of Zhejiang Province (Grant No. LQ22F020035), the National Natural Science Foundation of China (No. 32200860), and the Youth Foundation Project of Zhejiang Province (Grant No. LQ22F020035).

References

1. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
2. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech, vol. 2, pp. 1045–1048. Makuhari (2010)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Radford, A., Narasimhan, K., Sutskever, I., Salimans, T., et al.: Improving language understanding by generative pre-training (2018)
5. Liu, P., Li, W., Zou, L.: Nuli at semeval-2019 task 6: transfer learning for offensive language detection using bidirectional transformers. In: SemEval@ NAACL-HLT, pp. 87–91 (2019)
6. Chandra, M., et al.: AbuseAnalyzer: abuse detection, severity and target prediction for gab posts. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6277–6283. International Committee on Computational Linguistics (2020)
7. Dai, W., Yu, T., Liu, Z., Fung, P.: Kungfupanda at semeval-2020 task 12: bert-based multi-task learning for offensive language detection. arXiv preprint arXiv:2004.13432 (2020)
8. Wiedemann, G., Yimam, S.M., Biemann, C.: Uhh-lt at semeval-2020 task 12: fine-tuning of pre-trained transformer networks for offensive language detection. arXiv preprint arXiv:2004.11493 (2020)

9. Hakimov, S., Ewerth, R.: Combining textual features for the detection of hateful and offensive language. arXiv preprint arXiv:2112.04803 (2021)
10. Zhang, S., Zheng, D., Hu, X., Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, pp. 73–78 (2015)
11. Bebis, G., Georgiopoulos, M.: Feed-forward neural networks. IEEE Pot. **13**(4), 27–31 (1994)
12. Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M.: Benchmarking aggression identification in social media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 1–11 (2018)
13. Huang, Q., Singh, V.K., Atrey, P.K.: Cyber bullying detection using social and textual analysis. In: Proceedings of the 3rd International Workshop on Socially-aware Multimedia, pp. 3–6 (2014)
14. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on twitter. arXiv preprint arXiv:1706.01206 (2017)
15. Xu, J.M., Jun, K.S., Zhu, X., Bellmore, A.: Learning from bullying traces in social media. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 656–666 (2012)
16. Kwok, I., Wang, Y.: Locate the hate: detecting tweets against blacks. In: Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
17. Burnap, P., Williams, M.L.: Cyber hate speech on twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet **7**(2), 223–242 (2015)
18. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N.: Hate speech detection with comment embeddings. In: Proceedings of the 24th International Conference on World Wide Web, pp. 29–30 (2015)
19. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 11, pp. 512–515 (2017)
20. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, pp. 71–80. IEEE (2012)
21. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. arXiv preprint arXiv:1902.09666 (2019)
22. Rosenthal, S., Atanasova, P., Karadzhov, G., Zampieri, M., Nakov, P.: A large-scale semi-supervised dataset for offensive language identification. arXiv preprint arXiv:2004.14454 (2020)
23. Wang, Y.S., Chang, Y.: Toxicity detection with generative prompt-based inference. arXiv preprint arXiv:2205.12390 (2022)
24. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
25. Ben-David, E., Oved, N., Reichart, R.: Pada: a prompt-based autoregressive approach for adaptation to unseen domains. arXiv preprint arXiv:2102.12206 (2021)
26. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
27. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)

28. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786 (2021)
29. Reynolds, L., McDonell, K.: Prompt programming for large language models: beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–7 (2021)
30. Le Scao, T., Rush, A.M.: How many data points is a prompt worth? arXiv preprint arXiv:2103.08493 (2021)
31. Hu, S., Ding, N., Wang, H., Liu, Z., Li, J., Sun, M.: Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. arXiv preprint arXiv:2108.02035 (2021)
32. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723 (2020)
33. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: Autoprompt: eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980 (2020)
34. Hambardzumyan, K., Khachatrian, H., May, J.: Warp: word-level adversarial reprogramming. arXiv preprint arXiv:2101.00121 (2021)
35. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366 (2021)
36. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
37. Liu, X., et al.: Gpt understands, too. arXiv preprint arXiv:2103.10385 (2021)
38. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Semeval-2019 task 6: identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983 (2019)
39. Chandra, M., et al. Abuseanalyzer: abuse detection, severity and target prediction for gab posts. arXiv preprint arXiv:2010.00038 (2020)
40. Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M.: Logistic Regression. Springer, Heidelberg (2002)
41. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
42. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
43. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)

All Mistakes are not Equal: Comprehensive Hierarchy Aware Multilabel Predictions (CHAMP)

Ashwin Vaswani, Yashas Samaga, Gaurav Aggarwal, Praneeth Netrapalli, and Narayan Hegde(✉)

Google Inc., Menlo Park, USA
{ashwinvaswani,syashas,gauravaggarwal,
pnetrapalli,hegde}@google.com

Abstract. This paper considers the problem of Hierarchical Multilabel Classification (HMC), where (i) several labels can be present for each example, and (ii) labels are related via a domain-specific hierarchy tree. Guided by the intuition that all mistakes are not equal, we present Comprehensive Hierarchy Aware Multilabel Predictions (CHAMP), a framework that penalizes a misprediction depending on its severity as per the hierarchy tree. While such an approach has been applied to single-label classification, to the best of our knowledge, there is no work extending it to the general multilabel setting. A key challenge here is to identify a metric that quantifies the discrepancy between the *sets* of predicted labels and ground truth labels as per the given hierarchy tree. We address this challenge by proposing an assignment strategy for each predicted label to the set of ground truth labels, which naturally leads to CHAMP. Extensive experiments on six public HMC datasets across modalities (image, audio, and text) show that CHAMP significantly reduces the severity of mistakes, as measured by hierarchical metrics, by 18% on average across datasets. This improvement is achieved without a drop in classification performance over the standard (hierarchy-free) multilabel classification. We provide insights into the gains in hierarchical metrics by attributing them to hierarchy and co-occurrence information in the data.

1 Introduction

Many real-world prediction tasks have relationships among labels that can be encoded by a hierarchical structure over labels. Hierarchical relationships cover a wide spectrum of possible ways to express complex relations (is-a, part-of) among labels, which can provide domain-specific semantics. Movies organized by genres, blogs structured by subjects, species within a biological taxonomy, and e-commerce products arranged into categories and subcategories all exemplify

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_17.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 264–282, 2025.
https://doi.org/10.1007/978-3-031-78107-0_17

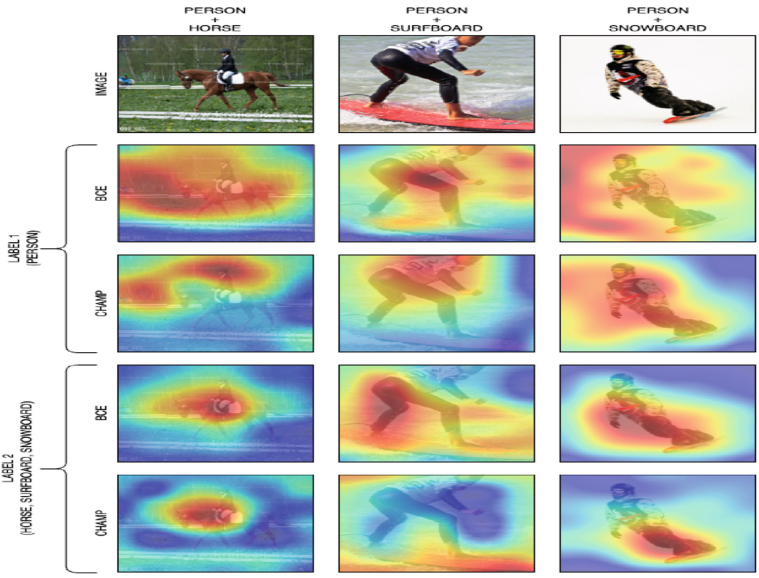


Fig. 1. Comparing CHAMP vs. Binary Cross Entropy (BCE) across highly co-occurring classes demonstrates the overreliance of baseline multilabel classification models on co-occurrence and background information. In Example 1, the BCE model uses information from the background to predict a person and horse. In Examples 2 and 3, the baseline model uses information leaked about the person to predict surfboard and snowboard respectively. The activation maps are computed using Grad-CAM.

types of data that lend themselves well to a hierarchical organization. Incorporating such hierarchical information in building classification models provides two distinct advantages. First, such an approach accommodates a range of complexities present in real-world data and incentivizes the model to learn concepts more comprehensively, capturing both overarching themes and finer details. Secondly, the hierarchical organization of labels facilitates graceful performance degradation by ensuring higher accuracy as you move up the categorization levels, especially in scenarios where the inputs are difficult to classify. Overall, the utilization of hierarchical classification not only strengthens the system's robustness but also improves utility by providing means to gracefully respond to inputs of varying difficulty.

Real-world data frequently exhibit multilabel nature as well. Movies, for instance, can be assigned multiple genres, like "romantic comedy." Similarly, blog posts can be tagged with several topics, and plant species can be classified as both "flowering" and "medicinal" simultaneously. A significant portion of the existing research on multilabel classification relies on label co-occurrence information, which is often fraught with spurious correlations. This reliance on such correlations can lead to predictions that are not based on meaningful semantics, potentially undermining the robustness of trained models (see Fig. 1). The label

horse, for example, may be associated with trees or a ranch in the background. A test image featuring a horse carriage on urban roads could be misidentified as a convertible, given that horses are seldom seen alongside roads, unlike other labels such as a person or wheels in the image. Such mispredictions, like labeling an animal as an automobile, could carry repercussions, say for automated driving assistance systems. Here, incorporating hierarchical information may help alleviate these problems by requiring labels to be coupled based on semantics rather than purely based co-occurrence information.

This intuition has spurred a significant amount of work on incorporating hierarchical information in single-label classification settings. While there have been several approaches centered on label embeddings and hierarchical model architectures Park, Kim, and Paik (2021), one of the most successful approaches has been via *hierarchy aware* loss functions (Bertinetto et al. (2019). In particular, Bertinetto et al. demonstrate that optimizing a *hierarchy-aware* loss function, in addition to moderate improvements in standard accuracy metrics Brust and Denzler (2018), can provide drastic improvements in *mistake severity metrics*, where the severity of a mistake depends on the distance in the hierarchy between the predicted and ground truth label.

Motivated by the prevalence of multilabel classification and the high practical relevance of making better mistakes, in this work, we extend the idea of hierarchy-aware loss functions from *single-label* to *multilabel* settings. In the single label setting, the gravity of a mistake depends on the distance $\text{dist}\,(i, j)$, according to the hierarchy tree, between the predicted label i and the ground truth label j. However, in the multilabel setting, it depends on the distance between the predicted label *set* $S \subseteq [L]$ and target label *set* $T \subseteq [L]$, where L is the number of labels. The question is how do we measure the hierarchical distance between two sets of labels?

Our key intuition is that given any *predicted* label $\ell \in [L]$, we can think of a probability distribution $p_\ell(\cdot)$ over the ground truth labels $T \subseteq [L]$ such that $p_\ell(j)$ denotes the probability that label ℓ is being mistakenly predicted for ground truth label $j \in T$ – we denote this event by $j \rightarrow \ell$. Importantly, $p_\ell(j)$ depends on the hierarchical distance between ℓ and j, i.e., $\text{dist}\,(\ell, j)$, with larger distance leading to smaller $p_\ell(j)$, and vice versa. Furthermore, the *cost* of a misprediction $j \rightarrow \ell$ is high if the $\text{dist}\,(\ell, j)$ is high.

More concretely, we consider the simplest technique for mistake assignment probabilities $p_\ell(j)$, wherein $p_\ell(j)$ adopts the form of a Dirac delta distribution supported on $argmin_{j \in T} d(\ell, j)$, where T is the set of ground truth labels for the given example. Given a predicted label ℓ and its corresponding mistake assignment probability vector $p_\ell(\cdot)$, we define the severity of this misprediction to be $\sum_{j \in T} p(\ell, j) \cdot d(\ell, j)$. With this notion of severity, we introduce an algorithm named *Comprehensive Hierarchy Aware Multilabel Prediction (CHAMP)*, that optimizes the standard binary cross entropy loss for each label weighed according to the severity of the misprediction.

To the best of our knowledge, ours is the first work for multilabel classification that proposes a notion of the severity of a misprediction. The generality of our

framework provides wider applicability across a range of Hierarchical Multilabel Classification (HMC) domains and datasets.

The key contributions of this work are as follows:

- We address the primary challenge of attributing predictions to the correct ground-truth labels when adapting single-label hierarchy-aware loss functions to the multilabel setting. We propose a simple but effective metric to quantify the severity of a mistake in HMC (Hierarchical Multilabel Classification), naturally leading to CHAMP.
- Our method augments existing multilabel classification algorithms to incorporate hierarchy to make better mistakes.
- Our experiments on six HMC datasets spanning several modalities like text, image, and audio demonstrate that CHAMP consistently makes better mistakes than standard multilabel classification without much loss in overall classification performance. We also identify reasons behind these performance gains by analyzing the contribution of hierarchy and co-occurrence information in data.

The rest of this paper is organized as follows. In Sect. 2, we summarize related work. In Sect. 3, we introduce the notation and terminology. In Sect. 4, we present our approach. Experimental results are presented in Sect. 5, followed by discussion in Sect. 6.

2 Related Work

Multilabel Classification: A significant body of literature on multilabel classification focuses on exploiting label correlations via graph neural networks Chen et al. (2019); Durand, Mehrasa, and Mori (2019); Chen et al. (2019), and label embeddings Chen et al. (2019); Wang et al. (2019). They also explore modeling image parts and attentional regions You et al. (2019); Wang et al. (2017); Ye et al. (2020) as well as using recurrent neural networks Nam et al. (2017); Wang et al. (2016), embedding space constraints Qu et al. (2021), region sampling Zhu et al. (2017) and cross-attention Ridnik et al. (2021). There has been recent interest in multilabel text classification Liu et al. (2017); Pal, Selvakumar, and Sankarasubbu (2020); Cerri, Barros, and de Carvalho (2011); Kumar et al. (2018). However, these approaches generally do not consider a hierarchy over labels.

Hierarchical Multilabel Classification: There has been a extensive amount of work directed towards HMC in various domains such as protein function prediction, online advertising systems Agrawal et al. (2013), and text classification Chatterjee et al. (2021); Mao et al. (2019); Chen et al. (2019); Daisey and Brown (2020). However, the primary focus of these studies has been on achieving better performance on standard metrics like AUPRC, but not on making better mistakes – which constitutes the primary focus of our work. Approaches to HMC

can be broadly classified as local or global. Local methods Cesa-Bianchi, Gentile, and Zaniboni (2006) train classifiers at each level of the hierarchy. Local methods can be further be categorized as local classifier per level (LCL) Cerri, Barros, and de Carvalho (2011), local classifier per node (LCN) Valentini (2009), and local classifier per parent node (LCP). It has also been argued Cerri, Barros, and de Carvalho (2011) that it is impractical to train separate classifiers at each level due to the several assumptions involved in semantics across siblings or nodes at the same level. On the other hand, global approaches Silla Jr. and Freitas (2009) employ a single classifier that factors in the complete hierarchy, mitigating the issue of error propagation that local methods suffer from. Local methods are better at capturing label correlations, whereas global methods are computationally cheaper. Peng et al. recently tried to use a hybrid loss function associated with specifically designed neural networks. HMCN-F Wehrmann, Cerri, and Barros (2018) uses a cascade of neural networks, each corresponding to a hierarchy level. Such architectures generally require all the paths in the label hierarchy to have the same length which limits their use. Giunchiglia and Lukasiewicz approach HMC by using parent and child probabilities constraints.

Our contribution, CHAMP, focuses on making better mistakes as well rather than solely imposing logical constraints on predictions. Moreover, because our framework of assigning mispredicted labels to the ground-truth labels is domain independent, CHAMP can be used with global, local, hybrid and many other HMC algorithms Cerri, Barros, and de Carvalho (2011); Silla Jr. and Freitas (2009); Wehrmann, Cerri, and Barros (2018). For example, CHAMP does not require explicit co-occurrence information among labels, does not restrict the semantics associated with nodes at the same level in the hierarchy and ground-truth nodes can be both leaf and non-leaf nodes.

Importance of Making Better Mistakes: Making better mistakes as per the hierarchy is important in several applications such as food recognition Mao et al. (2020); Wu et al. (2016), protein function prediction Cesa-Bianchi and Valentini (2009); Bi and Kwok (2011); Barros et al. (2013); Feng, Fu, and Zheng (2018), image annotation Dimitrovski et al. (2011) and text classification Mao et al. (2019); Rousu et al. (2006); Shen et al. (2021).

Making Better Mistakes in Single-Label Hierarchical Classification: Recently, there has been progress in incorporating hierarchy into single-label classification with the goal of making better mistakes. Some of the approaches include imposing logical constraints Giunchiglia and Lukasiewicz (2020), using hyperbolic embeddings Dhall et al. (2020), prototype learning Garnot and Landrieu (2021), label smearing and soft labels, multiple learning heads for different levels of the hierarchy Park, Kim, and Paik (2021), hierarchical post-processing Karthik et al. (2021) and others Mo et al. (2019); Deng et al. (2011). In particular, hierarchy-aware loss functions Bertinetto et al. (2019) have shown significant performance improvements. We extend this approach to the multilabel scenario.

3 Preliminaries and Problem Setting

We are given a set of labeled training examples $\{(x_i, y_i) : i = 1, \cdots, n\}$, where $x_i \in \mathbb{R}^d$ is the input example and $y_i \in \{0,1\}^L$ is the associated label vector, each (x_i, y_i) drawn from an underlying distribution $\mathcal{D}$ on $\mathbb{R}^d \times \{0,1\}^L$. Here L denotes the total number of labels/classes. We use the term label/class interchangeably. In addition to the training dataset, we are given a hierarchy tree $\mathcal{T}$ with L nodes where each node corresponds to one of the classes. We denote the distance between two nodes j and j' on the hierarchy tree $\mathcal{T}$ using dist (j, j'). Our goal is to train a prediction model $\mathcal{M}$ that takes x as input and outputs an L-dimensional real-valued score vector $\bar{y} \in [0,1]^L$. This real-valued score vector $\bar{y}$ is converted to a Boolean prediction vector $\hat{y} \in \{0,1\}^L$ using a scalar threshold τ. So, given a scalar threshold $\tau \in [0,1]$, the final prediction for class $j \in [L]$ is $\hat{y}_j = 1$ if $\bar{y}_j \geq \tau$ and $\hat{y}_j = 0$ otherwise.

3.1 Metrics

We now present the metrics we use to evaluate our method and existing methods. Given a model $\mathcal{M}$ and threshold τ, the precision and recall of class j is given by:

$$\text{Precision}(j) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y_j = 1 \,|\hat{y}_j = 1\right] \text{ and}$$
$$\text{Recall}(j) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\hat{y}_j = 1 \,|y_j = 1\right].$$

The overall precision and recall are then given by taking an average over all the L labels.

As our evaluation metric, we use **Area Under the Precision-Recall Curve (AUPRC)**, which, as the name suggests, is given by the area under the curve traced by precision vs. recall, as the threshold τ changes. AUPRC is a standard, popular metric to measure the quality of predictions in multilabel settings, and does not depend on the hierarchy tree.

4 Method

In this section, we will present the key insights of this work, leading to the main algorithm CHAMP.

4.1 Assignment of Predicted to Ground Truth Labels

Given a predicted label set $S \subseteq [L]$ and a ground truth label set $T \subseteq [L]$, we wish to quantify the severity of mispredictions in S. We do so by first defining the severity of misprediction for every label $\ell \in S$. The key intuition is that the severity of misprediction $\ell \in S$ depends on which element $j \in T$, that ℓ was confused with – we denote this by $j \leftarrow \ell$. To formalize this, we posit that every predicted label ℓ has an associated probability distribution $p_\ell(\cdot)$ over the target label set T, where $p_\ell(j)$ is the probability that ℓ was confused with j. Furthermore, $p_\ell(\cdot)$

depends on the hierarchy tree. We consider an assignment model for $p_\ell(\cdot)$: (i) where $p_\ell(j) = 0$ for $j \notin argmin_{j'} \text{dist}(\ell, j')$ and $\sum_{j \in argmin_{j'} \text{dist}(\ell,j')} p_\ell(j) = 1$. Furthermore, the severity of misprediction $j \leftarrow \ell$ is directly proportional to $\text{dist}(\ell, j)$.

4.2 The CHAMP Loss Functions

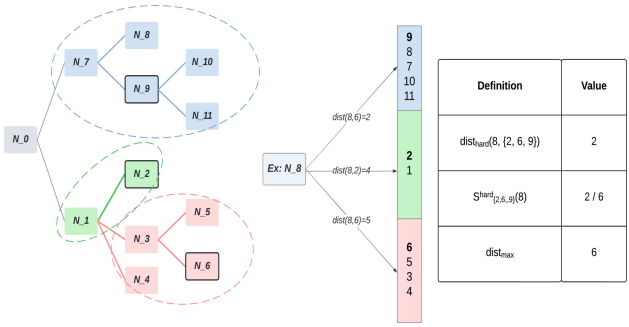


Fig. 2. An example illustrating the severity metrics (3) used in CHAMP. The ground truth labels are $T = \{N_2, N_6, N_9\}$. For each ground truth label, we represent all other labels that are closest to it, compared to any other ground truth label using the same color. The set of all such labels is referred to as the sphere of influence of the corresponding ground truth label. For the specific label $j = N_8$, we further compute all the relevant quantities appearing in (3). The distances of N_8 to N_2, N_6 and N_9 are $4, 5$ and 2 respectively. We set the scaling parameter $\beta = 1$ for illustration.

The most classical approach to multilabel classification is to solve L binary classification problems, one for each label using the binary cross entropy (BCE) loss given by (Fig. 2):

$$\ell_{(x,y)}(\mathcal{M}) = -\sum_{j=1}^{L} \{y_j \log \hat{y}_j + (1 - y_j) \log (1 - \hat{y}_j)\}, \tag{1}$$

where $\hat{y}$ is the score output by model $\mathcal{M}$ on input x. We now wish to introduce weight different terms according to the severity of misprediction, as discussed in the above section. All false negatives are equally severe since they are ground truth labels while the severity of a false positive depends on which ground truth label it was confused with, and the distance between this ground truth label and the predicted label. So, we consider the following modified BCE loss function:

$$\tilde{\ell}_{(x,y)}(\mathcal{M}) = -\sum_{j=1}^{L} \{y_j \log \hat{y}_j + (1 + s_T(j))\,(1 - y_j) \log (1 - \hat{y}_j)\}, \tag{2}$$

where T denotes the ground truth labels i.e., $T = \{\ell : y_\ell = 1\}$ and $s_T(j)$ denotes the severity of a false positive prediction on label j, defined as: $s_T(j) = \mathbb{E}_{\ell \sim p_j(\cdot)}[\text{dist}\,(\ell, j)]$. Thus, we have:

$$s_T(j) \triangleq \beta \cdot \frac{\min_{\ell \in T} \text{dist}\,(\ell, j)}{\text{dist}_{\max}} \text{and} \tag{3}$$

Here, $\text{dist}_{\max} \triangleq \max_{j,j' \in [L]} \text{dist}\,(j, j')$ denotes the maximum distance between any two labels on the hierarchy tree, and β is a scaling parameter (a hyperparameter).

4.3 Hierarchical Metrics

Based on the above discussion, we now extend a popular single-label hierarchical metric to the multilabel setting, which we will use for evaluating CHAMP.

Depth of Lowest Common Ancestor (LCA): Given a predicted label j, and the set of ground truth labels T, we first compute ℓ such that $\ell \in argmin_{j' \in T} \text{dist}\,(j', j)$, and compute the depth, computed from the root, of the least common ancestor for j and ℓ on the hierarchy tree $\mathcal{T}$ Deng et al. (2010; 2009). As remarked in Deng et al. (2009), this should be thought of in logarithmic terms, as the number of confounded classes is exponential in the height.

5 Results

In this section, we present our main results evaluating CHAMP on six datasets spanning across vision, audio, and text. Our results demonstrate that:

- CHAMP provides substantial improvements in the hierarchical metrics over the standard baseline models and multilabel classification methods without loss in overall classification performance.
- CHAMP's loss function can be used along with other multi-label loss functions to augment the model to make less severe mistakes.
- CHAMP gain in hierarchy aware metric performance is reasoned by analyzing localized improvements on labels across multiple levels in the hierarchy tree.
- Improvement in classification performance can be distinctly attributed to hierarchy and co-occurrence information in the dataset.

5.1 Datasets and Baseline

We evaluate our approach on six public HMC datasets across image, text, and audio modalities. We perform:

- multilabel image classification on OpenImages V4 Kuznetsova et al. (2020), Food201 Myers et al. (2015), and MS-COCO 2017 Lin et al. (2014) datasets;

Table 1. Detailed information about the six public datasets used in our experiments. The Train and Test columns denote the number of train and test samples in the datasets, respectively. μ_s denotes the mean number of samples per label. μ_l and $\max_l$ denote the mean number of labels per sample and the maximum number of labels per sample, respectively. BF denotes the branching factor, which is the average number of children per node. D, Leaf, and NL denote the depth, and the number of leaf and non-leaf nodes in the tree, respectively.

Type	Dataset	Train	Test	μ_s	μ_l	max_l	BF	D	Leaf	NL
Image	Food201	35242	15132	334	1.91	9	6.4	4	201	35
	COCO	1,18,287	40,670	4287	2.89	18	6.57	3	80	14
	OpenImages V4	17,43,042	1,25,436	7384	2.53	19	6.81	5	526	73
Text	RCV1	23,149	7,81,265	729	3.24	17	4.71	4	82	21
	NYT	12,79,092	5,47,863	69,314	2.52	14	4.17	4	91	27
Audio	FSDK Audio	4970	4480	72	1.4	6	2.53	5	80	49

– multilabel text classification on Reuters Corpus Volume 1 (RCV1) Lewis et al. (2004) and New York Times articles (NYT)Sandhaus (2008); and
– multilabel audio classification on FSDKaggle2019 dataset. Fonseca et al. (2019)

RCV1 is a newswire dataset of the articles collected between 1996–1997 from Reuters. New York Times contains articles from New York Times published between January 1st, 1987, and June 19th, 2007. The hierarchies for OpenImages, COCO, RCV1, NYT, and FSDK are provided by their respective authors and the hierarchy for Food201 was adapted based on the hierarchy given in Wu et al. (2016). Detailed labels and hierarchical information of each dataset can be found in Table 1. These six datasets cover diverse settings of data distributions and types of hierarchy.

Baseline: Table 2 demonstrates how CHAMP can augment state-of-the-art multilabel loss functions. Since our main contribution is to extend hierarchy aware loss functions from a single label to multilabel settings, we compare CHAMP primarily with vanilla binary cross entropy loss i.e., (1). We also demonstrate the importance of the mistake severity assignment step in CHAMP by comparing it with a naive extension of the single label hierarchical approach to HMC as follows. For each sample x_i with label set T_i, we construct a new dataset where sample x_i is repeated in the dataset $|T_i|$ times each with a different label in T_i. Thus, we convert the multilabel problem into a single-label problem where sample-label pairs include (x_i, j) where $j \in T_i$. We denote this experiment as M2S. Finally, the generality of CHAMP framework enables us to easily augment state-of-the-art multi-label loss functions with hierarchy information, thereby encouraging better mistakes. In particular, we consider the asymmetric loss function introduced in Baruch et al. (2020) and augment it with CHAMP to obtain CHAMP-Asymmetric (see Supplementary material for the precise formulation).

Architecture: Our model consists of L different classifiers on top of a standard feature extractor backbone, where L denotes the number of labels.

5.2 Training Configuration

Image Classification: We conduct experiments using two different backbones, Efficientnetv2S Tan and Le (2021) and Mobilenetv2 Sandler et al. (2018) namely, that are initialized with ImageNet pre-trained weights. They are followed by a dropout layer, linear layer, and sigmoid activation in order. We use an image size of 224×224 and a standard set of data augmentations methods. Note that state-of-the-art approaches Ben-Baruch et al. (2020) use larger image sizes for training, autoaugment Cubuk et al. (2018) and cutout DeVries and Taylor (2017) for augmentations, one-cycle learning rate schedulers Smith and Topin (2017) amongst many others. We do not intend to compete with the state-of-the-art approaches to multilabel classification and limit ourselves to demonstrating the value of adding hierarchical knowledge.

Text Classification: We conduct experiments using small bert Turc et al. (2019) (uncased, L=2, H=768, A=12) and Sun et al. (2020) embeddings as base extractor with weights initialized from pre-trained models on Wikipedia Vrandečić and Krötzsch (2014) and Books corpus datasets Zhu et al. (2015). All text sentences are first converted to lowercase.

Audio Classification: We convert the audio wav files into Mel spectrogram with a sampling rate of 44100, number of Mel bands as 347, length of the FFT window as 2560, lowest frequency as 20, and highest frequency as 44100//2. We then treat it as an image classification problem with the setup described previously.

All the models are trained for 25 epochs using Adam Kingma and Ba (2014) optimizer with an initial learning rate of 1e-4, weight decay of 4e-5, and a batch size of 32. We reduce the learning rate on plateau by a factor of 0.2 with patience of 5 epochs. We tune the hyperparameter β between 0 and 1 using validation performance. Data splits are provided in the supplementary material. We run all experiments three times and report the average performance.

5.3 Comparison to Baselines

Table 2 compares the performance of CHAMP with the standard BCE, M2S, and Asymmetric loss. CHAMP-Assymetric performs 18% ($p<0.05$) better on hierarchy (better mistakes) metric LCA compared to Asymmetric loss (Multi-label) averaged over 6 datasets. CHAMP has matching AUPRC performance compared to Asymmetric loss (mean difference $< 1\%$, $p<0.05$). To summarize, we see substantial improvements in hierarchical metrics (LCA) on all the datasets in addition to considerable improvements in AUPRC as well. In fact,

Table 2. This table compares standard binary cross-entropy (BCE) loss for multilabel classification and CHAMP. The average value over three runs is reported in the table. Most of the experiments have a standard deviation value of <0.007. For each metric ↑ indicates that higher is better. We note that CHAMP significantly outperforms BCE and Asymm on hierarchical metrics while maintaining AUPRC demonstrating the utility of hierarchy information. Due to the exponential increase in the size of the OpenImage and NYT datasets after repeating samples, it became unfeasible to compute the M2S experiments for them.

Dataset	Experiment	AUPRC ↑	LCA ↑
OpenImages v4	BCE	0.342 ± 0.009	1.556 ± 0.012
	M2S	–	–
	CHAMP	**0.564 ± 0.002**	**1.852 ± 0.006**
	Asymm	**0.435 ± 0.005**	0.777 ± 0.006
	CHAMP-Asymm	0.407 ± 0.003	**0.887 ± 0.006**
Food201	BCE	0.577 ± 0.001	1.415 ± 0.013
	M2S	0.573 ± 0.004	1.431 ± 0.008
	CHAMP	**0.585 ± 0.001**	**1.486 ± 0.008**
	Asymm	**0.590 ± 0.001**	0.824 ± 0.018
	CHAMP-Asymm	0.584 ± 0.001	**1.114 ± 0.009**
COCO	BCE	0.780 ± 0.001	1.994 ± 0.006
	M2S	0.744 ± 0.002	1.649 ± 0.032
	CHAMP	**0.779 ± 0.003**	**2.004 ± 0.004**
	Assym	0.751 ± 0.010	1.480 ± 0.001
	CHAMP-Asymm	**0.768 ± 0.002**	**1.924 ± 0.006**
NYT	BCE	0.515 ± 0.002	2.636 ± 0.001
	M2S	–	–
	CHAMP	**0.591 ± 0.009**	**2.662 ± 0.002**
	Asymm	**0.655 ± 0.001**	2.324 ± 0.013
	CHAMP-Asymm	0.643 ± 0.001	**2.599 ± 0.005**
RCV1	BCE	0.659 ± 0.006	1.621 ± 0.003
	M2S	0.674 ± 0.003	**1.714 ± 0.014**
	CHAMP	**0.675 ± 0.002**	1.633 ± 0.002
	Asymm	**0.676 ± 0.001**	1.426 ± 0.004
	CHAMP-Asymm	0.662 ± 0.001	**1.470 ± 0.005**
FSDK Audio	BCE	**0.467 ± 0.003**	1.649 ± 0.010
	M2S	0.449 ± 0.009	1.459 ± 0.076
	CHAMP	0.448 ± 0.002	**1.667 ± 0.014**
	Asymm	**0.461 ± 0.005**	1.322 ± 0.019
	CHAMP Asymm	0.458 ± 0.007	**1.511 ± 0.006**

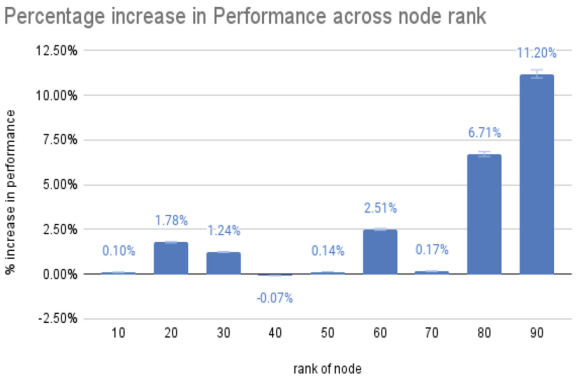


Fig. 3. Comparing the % increase in AUPRC vs. rank of the node with respect to the number of samples per label

on some datasets, we see substantial improvements in AUPRC, e.g. 64.91% relative improvement over BCE on the OpenImages dataset. This could potentially be attributed to OpenImages being a large dataset with an extremely branched hierarchy as seen in Table 1, offering semantic information from parents and sibling entities while learning. As we can see in Table 2, single-label hierarchical training (M2S) using the same distance-based loss formulation as CHAMP leads to lower AUPRC performance. We also demonstrate how CHAMP can augment state-of-the-art multi-label loss functions. In particular, we compare the Asymmetric loss against its augmented version CHAMP-Asymmetric. The results demonstrate that CHAMP-Asymmetric obtains significant improvements on hierarchical metrics such as LCA, without compromising on standard metrics such as AUPRC. This is because the Asymmetric loss is trained with the objective to improve AUPRC, and introducing hierarchy then helps to make better mistakes. We can further see that the highest percentage gains are achieved for the 3 datasets with the highest branching factors (OpenImages, COCO, and Food201). Besides, we hypothesize that hierarchy should add valuable learning information in the low data and long tail regimes. We investigate this further in subsequent experiments and try to understand the reasons behind AUPRC improvements of CHAMP. Overall, these results demonstrate that CHAMP provides a generic framework that can be applied to more sophisticated multilabel classification methods to improve their mistake quality. Experiments in subsequent sections are designed to show the robustness of the CHAMP algorithm in various data scenarios. We thus use BCE as a baseline for comparison for interpretation simplicity. This helps to see the improvements from both hierarchy and multilabel information.

5.4 Localising Improvements

The information from the hierarchy likely helps labels that have very few examples. To verify this intuition, we calculate the average performance of the top

Table 3. AUPRC computed on the top 20% and bottom 20% labels, sorted according to the number of examples per label, for both BCE and CHAMP on the six datasets. We see that CHAMP achieves much more improvement on the bottom 20% labels than on the top 20% labels.

Label distribution	Experiment\Dataset	OpenImages	COCO	Food201	RCV1	NYT	FSDK
Top 20%	BCE	0.537	0.678	0.574	0.897	0.886	0.435
Top 20%	CHAMP	0.584	0.682	0.593	0.901	0.612	0.442
Bottom 20%	BCE	0.146	0.639	0.121	0.337	0.203	0.520
Bottom 20%	CHAMP	0.612	0.675	0.138	0.357	0.459	0.573

20% and bottom 20% of labels ordered by frequency. To investigate CHAMP's multi-label learning capabilities further, we analyze the AUPRC scores of the models in detail, although similar trends are also observed for LCA. Table 3 presents the performance of BCE and CHAMP on these subsets of labels on all six datasets. The results show that the AUPRC improvements achieved by CHAMP for the bottom 20% labels are substantially larger than that for the top 20%. We further dissect where we find improvements these improvements. We demonstrate this using the COCO dataset for brevity, although similar trends are observed for other datasets. Figure 3 shows that CHAMP improves performance for low-ranked nodes more, that is, labels with less number of samples. This is intuitive as hierarchical and multilabel information together helps in improving the low data problem.

5.5 Contribution of Hierarchy and Co-occurrence Information in Better Performance

HMC benefits from both hierarchy and label co-occurrence information. In this section, we aim to study the attribution of the obtained improvements to hierarchy and co-occurrence information. We work with ordered pairs of labels in this experiment. We use the co-occurrence probabilities to sort and bin pairs of classes into two categories: high co-occurrence (top 20% by co-occurrence) and low co-occurrence (bottom 20%). We use a similar process to bin the pairs as high-semantic-similarity and low-semantic-similarity based on path distance in the hierarchy, i.e., labels closer in the tree are assumed to be more semantically similar. We report the average relative improvement in the AUPRC scores between the two labels in CHAMP and BCE for the COCO dataset for brevity in all four combinations of the categories in Table 4. We observe that the low-co-occurrence-high-semantic-similarity category shows maximum improvement, which demonstrates the benefit of using hierarchical information. The results suggest that HMC methods may be well suited to learn and adapt to rare co-occurring classes, improve generalization, and be more robust toward out-of-distribution samples.

Table 4. Relative increase in AUPRC of CHAMP over BCE on label pairs with low/high co-occurrence and similarity.

Subset	% Δ AUPRC ↑	#pairs
low co-occurrence low semantic similarity	1.46%	300
low co-occurrence high semantic similarity	3.32%	14
high co-occurrence low semantic similarity	1.61%	428
high co-occurrence high semantic similarity	1.87%	184

6 Conclusion and Discussion

Summary: In this work, we consider the problem of hierarchical multilabel classification. While there is a rich literature on encouraging better, hierarchy-aware mistakes in single-label hierarchical classification, there were no such works for hierarchical multilabel classification. In this paper, we develop one such framework – CHAMP– and, through experiments on six diverse datasets across vision, text, and audio, demonstrate that CHAMP provides improvements not only on hierarchical metrics but also on standard metrics like AUPRC. The framework can be used to define any kind of set associations between predicted and ground truth labels, even beyond hierarchical association. This provides a platform for further research to enable multilabel classification methods to make better mistakes and incorporate domain information outside of training data. We further demonstrate the benefits of hierarchy in increased robustness and better sample complexity. Finally, we also demonstrate that the proposed framework is general and can be incorporated with other multilabel classification algorithms. Besides, we conducted further analysis into alternative formulations of mistake assignment such as a softer assignment strategy involving a harmonic mean of the distances from each label and severity (linear scaling, exponential scaling, etc.). We observed that no major improvements were obtained with such added complexities. Further details and results of these experiments are presented in the appendix.

Future Work: Though we envisage a natural extension of our algorithm to DAG (directed acyclic graphs), it would need more work to make semantic-aware mistakes in datasets with generic graphical label relationships. Our methods also set the platform and extend to more complex learning algorithms like detection Shkodrani et al. (2021), segmentation, image retrieval, and ranking to make better mistakes. Going further, contrastive learning approaches like triplet loss and object detection Shkodrani et al. (2021) are also compatible with our approach and benefit from adjusting the loss functions with hierarchy and co-occurrence information to make better mistakes. Hierarchy relations are more relevant when the feature representation similarity does not align with semantic similarity, and the cost of making mistakes is high. For example, a black slate and a tv monitor look visually similar but are semantically different. Based on some preliminary

results, we see 2.74% and 0.9% percentage drop in AUPRC when trained with random tree structures on OpenImages and COCO datasets. We believe that this occurs due to a conflict in semantic and visual similarities.

Discussion: Supervised deep neural networks rely on large data to generalize well to unseen data. While typical way to increase the data is to collect more examples with supervised labels, there is also a growing appreciation for capturing labels with richer semantics and richer annotations. Hierarchical and multiple label annotations can cover many relations like part-of, is-a, and similar-to. We encourage the community to invest more in better labeling procedures by showcasing downstream gains such as training with less data, making better mistakes, and robustness to noise. Out-of-distribution examples and adversarial attacks can exploit the over-reliance on high co-occurring labels. CHAMP leads to more robust feature representations as it does not only heavily rely on co-occurrence information but also uses semantic relationships. We conclude our CHAMP work by setting the first platform in the journey to learn to make better mistakes in the context of rich label relationships.

References

Agarwal, A.: Selective sampling algorithms for cost-sensitive multiclass prediction. In: 30th International Conference on Machine Learning, ICML 2013, pp. 2257–2265 (2013)

Agrawal, R.; Gupta, A., Prabhu, Y., Varma, M., Varma, M.: Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In: Proceedings of the International World Wide Web Conference (2013)

Barros, R.C., Cerri, R., Freitas, A.A., de Carvalho, A.C.P.L.F.: Probabilistic clustering for hierarchical multi-label classification of protein functions. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8189, pp. 385–400. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40991-2_25 ISBN 978-3-642-40991-2

Ben-Baruch, E., et al.: Asymmetric loss for multi-label classification (2020)

Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., Lord, N.A.: Making better mistakes: leveraging class hierarchies with deep networks (2019)

Bi, W., Kwok, J.T.: Multi-label classification on tree- and DAG-structured hierarchies. In: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML 2011, pp. 17–24. Omnipress, Madison (2011). ISBN 9781450306195

Brust, C.-A., Denzler, J.: Integrating domain knowledge: using hierarchies to improve deep classifiers (2018)

Cerri, R., Barros, R.C., de Carvalho, A.C.P.L.F.: Hierarchical multi-label classification for protein function prediction: a local approach based on neural networks. In: 2011 11th International Conference on Intelligent Systems Design and Applications, pp. 337–343 (2011)

Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Hierarchical classification: combining Bayes with SVM. In: Proceedings of the 23rd International Conference on Machine Learning (2006)

Cesa-Bianchi, N., Valentini, G.: Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. In: Džeroski, S., Guerts, P., Rousu, J. (eds.) Proceedings of the Third International Workshop on Machine Learning in Systems Biology, vol. 8 of Proceedings of Machine Learning Research, pp. 14–29. PMLR, Ljubljana (2009)

Chatterjee, S., Maheshwari, A., Ramakrishnan, G., Jagaralpudi, S.N.: Joint learning of hyperbolic label embeddings for hierarchical multi-label classification (2021)

Chen, B., Huang, X., Xiao, L., Cai, Z., Jing, L.: Hyperbolic interaction model for hierarchical multi-label classification (2019)

Chen, Z.-M., Wei, X.-S., Jin, X., Guo, Y.: Multi-label image recognition with joint class-aware map disentangling and label correlation embedding. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 622–627 (2019)

Chen, Z.-M., Wei, X.-S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks (2019)

Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: learning augmentation policies from data (2018)

Daisey, K., Brown, S.D.: Effects of the hierarchy in hierarchical, multi-label classification. Chemom. Intell. Lab. Syst. **207**, 104177 (2020)

Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 71–84. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15555-0_6

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)

Deng, J., Satheesh, S., Berg, A., Li, F.: Fast and balanced: efficient label tree learning for large scale object recognition. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 24. Curran Associates Inc., Newry (2011)

DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout (2017)

Dhall, A., Makarova, A., Ganea, O., Pavllo, D., Greeff, M., Krause, A.: Hierarchical image classification using entailment cone embeddings. arXiv:2004.03459 (2020)

Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Hierarchical annotation of medical images. Pattern Recogn. **44**(10), 2436–2449 (2011)

Durand, T., Mehrasa, N., Mori, G.: Learning a deep convnet for multi-label classification with partial labels (2019)

Feng, S., Fu, P., Zheng, W.: A hierarchical multi-label classification method based on neural networks for gene function prediction. Biotechnol. Biotechnol. Equip. **32**(6), 1613–1621 (2018)

Fonseca, E., Plakal, M., Font, F., Ellis, D.P.W., Serra, X.: Audio tagging with noisy labels and minimal supervision (2019)

Gao, B.-B., Zhou, H.-Y.: Learning to discover multi-class attentional regions for multi-label image recognition. IEEE Trans. Image Process. **30**, 5920–5932 (2021)

Garnot, V.S.F., Landrieu, L.: Leveraging class hierarchies with metric-guided prototype learning (2021)

Giunchiglia, E., Lukasiewicz, T.: Coherent hierarchical multi-label classification networks. ArXiv arxiv:2010.10151 (2020)

Giunchiglia, E., Lukasiewicz, T.: Coherent hierarchical multi-label classification networks. arXiv:2010.10151 (2020)
Karthik, S., Prabhu, A., Dokania, P.K., Gandhi, V.: No cost likelihood manipulation at test time for making better mistakes in deep networks (2021)
Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
Kumar, V., Pujari, A.K., Padmanabhan, V., Kagita, V.R.: Group preserving label embedding for multi-label classification (2018)
Kuznetsova, A., et al.: The open images dataset V4. Int. J. Comput. Vision **128**(7), 1956–1981 (2020). https://doi.org/10.1007/s11263-020-01316-z
Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
Liu, J., Chang, W.-C., Wu, Y., Yang, Y.: Deep learning for extreme multi-label text classification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 115–124. Association for Computing Machinery, New York (2017). ISBN 9781450350228
Mao, R., He, J., Shao, Z., Yarlagadda, S.K., Zhu, F.: Visual aware hierarchy based food recognition (2020)
Mao, Y., Tian, J., Han, J., Ren, X.: Hierarchical text classification with reinforced label assignment. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019)
Mo, X., Zhu, J., Zhao, X., Liu, M., Wei, T., Luo, W.: Exploiting category-level semantic relationships for fine-grained image recognition. In: Lin, Z., et al. (eds.) PRCV 2019. LNCS, vol. 11857, pp. 50–62. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31654-9_5 ISBN 978-3-030-31654-9
Myers, A., et al.: Im2Calories: towards an automated mobile vision food diary. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1233–1241 (2015)
Nam, J., Loza Mencía, E., Kim, H.J., Fürnkranz, J.: Maximizing subset accuracy with recurrent neural networks in multi-label classification. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates Inc., Newry (2017)
Otero, F.E.B., Freitas, A.A., Johnson, C.G.: A hierarchical multi-label classification ant colony algorithm for protein function prediction. Memetic Comput. **2**, 165–181 (2010)
Pal, A., Selvakumar, M., Sankarasubbu, M.: MAGNET: multi-label text classification using attention-based graph neural network. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications (2020)
Park, J., Kim, H., Paik, J.: Cf-cnn: coarse-to-fine convolutional neural network. Appl. Sci. **11**(8), 3722 (2021)
Peng, H., et al.: Large-scale hierarchical text classification with recursively regularized deep graph-CNN. In: Proceedings of the 2018 World Wide Web Conference, WWW 2018, pp. 1063–1072. CHE: International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2018). ISBN 9781450356398
Qu, X., Che, H., Huang, J., Xu, L., Zheng, X.: Multi-layered semantic representation network for multi-label image classification (2021)

Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E., Noy, A.: ML-decoder: scalable and versatile classification head (2021)
Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification models. J. Mach. Learn. Res. **7**, 1601–1626 (2006)
Sandhaus, E.: The new york times annotated corpus. Linguist. Data Consortium, Philadelphia **6**(12), e26752 (2008)
Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, vol. 2018, pp. 4510–4520 (2018)
Shen, J., Qiu, W., Meng, Y., Shang, J., Ren, X., Han, J.: TaxoClass: hierarchical multi-label text classification using only class names. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4239–4249. Association for Computational Linguistics (2021)
Shkodrani, S., Wang, Y., Manfredi, M., Baka, N.: United we learn better: harvesting learning improvements from class hierarchies across tasks (2021)
Silla Jr., C.N., Freitas, A.A.: A global-model naive bayes approach to the hierarchical prediction of protein functions. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 992–997 (2009)
Smith, L.N., Topin, N.: Super-convergence: very fast training of neural networks using large learning rates (2017)
Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., Zhou, D.: MobileBERT: a compact task-agnostic BERT for resource-limited devices (2020)
Tan, M., Le, Q.V.: EfficientNetV2: smaller models and faster training. ArXiv arxiv:2104.00298 (2021)
Triguero, I., Vens, C.: Labelling strategies for hierarchical multi-label classification techniques. Pattern Recogn. **56**, 170–183 (2016)
Turc, I., Chang, M.-W., Lee, K., Toutanova, K.: Well-read students learn better: on the importance of pre-training compact models (2019)
Valentini, G.: True path rule hierarchical ensembles. In: MCS (2009)
Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: a unified framework for multi-label image classification (2016)
Wang, Y., et al.: Multi-label classification with label graph superimposing (2019)
Wang, Z., Chen, T., Li, G., Xu, R., Lin, L.: Multi-label image recognition by recurrently discovering attentional regions (2017)
Wehrmann, J., Cerri, R., Barros, R.: Hierarchical multi-label classification networks. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, vol. 80 of Proceedings of Machine Learning Research, pp. 5075–5084. PMLR (2018)
Wu, H., Merler, M., Uceda-Sosa, R., Smith, J.R.: Learning to make better mistakes: semantics-aware visual food recognition. In: Proceedings of the 24th ACM International Conference on Multimedia, MM 2016, pp. 172–176. Association for Computing Machinery, New York (2016). ISBN 9781450336031
Wu, J., Xiong, W., Wang, W.Y.: Learning to learn and predict: a meta-learning approach for multi-label classification (2019)
Ye, J., He, J., Peng, X., Wu, W., Qiao, Y.: Attention-driven dynamic graph convolutional network for multi-label image recognition (2020)
You, R., Guo, Z., Cui, L., Long, X., Bao, Y., Wen, S.: Cross-modality attention with semantic graph embedding for multi-label classification (2019)

Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with image-level supervisions for multi-label image classification (2017)
Zhu, Y., et al.: Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: The IEEE International Conference on Computer Vision (ICCV) (2015)

IDAL: Improved Domain Adaptive Learning for Natural Images Dataset

Ravi Kant Gupta(✉), Shounak Das, and Amit Sethi

Indian Institute of Technology Bombay, Powai, Mumbai, India
{ravigupta131,21D070068,asethi}@iitb.ac.in

Abstract. We present a novel approach for unsupervised domain adaptation (UDA) for natural images. A commonly-used objective for UDA schemes is to enhance domain alignment in representation space even if there is a domain shift in the input space. Existing adversarial domain adaptation methods may not effectively align different domains of multi-modal distributions associated with classification problems. Our approach has two main features. Firstly, its neural architecture uses the deep structure of ResNet and the effective separation of scales of feature pyramidal network (FPN) to work with both content and style features. Secondly, it uses a combination of a novel loss function and judiciously selected existing loss functions to train the network architecture. This tailored combination is designed to address challenges inherent to natural images, such as scale, noise, and style shifts, that occur on top of a multi-modal (multi-class) distribution. The combined loss function not only enhances model accuracy and robustness on the target domain but also speeds up training convergence. Our proposed UDA scheme generalizes better than state-of-the-art for CNN-based methods on Office-Home, Office-31, and VisDA-2017 datasets and comaparable for DomainNet dataset.

Keywords: Adversarial · Deep Learning · Domain Adaptation · Natural Images

1 Introduction

Unsupervised Domain Adaptation (UDA) addresses the performance degradation caused by domain shift in supervised learning, where there's a significant distribution difference between training (source) and testing (target) data domains. Adversarial-based UDA, utilizing frameworks like Generative Adversarial Networks (GANs) [25] and Domain Adversarial Neural Networks (DANN) [8], aims to mitigate this by learning domain-invariant features from unlabeled target data. By promoting feature harmonization while retaining class information, these models enhance target domain generalization. Despite promising results in image classification and object detection, adversarial UDA faces challenges such as hyper-parameter sensitivity, high-dimensional space navigation, and domain shift detection.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 283–297, 2025.
https://doi.org/10.1007/978-3-031-78107-0_18

To address the aforementioned challenge, we developed an unsupervised domain adaptation approach that surpasses the state-of-the-art UDA performance of convolution neural networks (CNNs) for benchmark natural image datasets – Office-Home [36], Office-31 [29], VisDA-2017 [26], and shows comparable results for DomainNet [27].

Inspired by the concept of a conditional domain adversarial network (CDAN) [22], our core approach – Improved Domain Adaptive Learning (IDAL) – involves concurrent training of a feature extractor (typically a deep neural network) and a domain classifier (discriminator) tasked with distinguishing between source and target domains. We have explored various CNN-based feature extractors such as ResNet-101, ResNet-50 [12], ViT [6], and ConvMixer [33] to extract meaningful features. The feature extractor's aim is to learn representations that are invariant to domain shifts, and thus deceive the domain classifier that endeavors to correctly classify the domain of the extracted features. The integration of ResNet-50 and FPN combines [20] deep feature representation and multi-scale extraction, essential for tasks like object detection and segmentation. Given that object scale and style vary by domain, this synergy makes ResNet + FPN a strong candidate for Unsupervised Domain Adaptation (UDA), focusing on higher-level domain-specific feature suppression. This application to UDA is novel.

In the adversarial training process, the feature extractor (ResNet-50 + FPN) and domain classifier compete: the extractor aims to produce domain-agnostic features, while the classifier attempts to distinguish between domains. This method fosters the development of domain-invariant features, enhancing transferability across source and target domains.

To improve the training process, we propose a novel loss function called pseudo label maximum mean discrepancy (PLMMD). We use this loss in addition to certain existing losses – maximum information loss (entropy loss) [15], maximum mean discrepancy (MMD) loss [21], minimum class confusion (MCC) loss [13]. Our model integrates several loss functions to enhance domain adaptation and classification accuracy: Maximum information loss clusters target features by class, preserving key information. MMD loss bridges domain gaps by comparing mean embeddings. MCC loss boosts accuracy by minimizing class confusion, vital for uneven datasets. Our innovative PLMMD loss selectively extracts domain-invariant features, speeding up training. This tailored mix of loss functions enables our method to outperform existing CNN models and achieve quicker convergence on natural image datasets– Office-Home [36], Office-31 [29], and VisDA [26].

2 Background and Related Works

In unsupervised domain adaptation (UDA), we have data from a source domain $D_s = \{(x_{s_i}, y_{s_i})\}_{i=1}^{n_s}$ as n_s labeled examples and that from a target domain $D_s = \{(x_{t_i}, y_{t_i})\}_{i=1}^{n_t}$ as n_t unlabeled examples where y_{t_i}'s are unknown. The source domain and target domain are sampled from the distributions $P(x_s, y_s)$ and $Q(x_t, y_t)$ respectively. Notably, the two distributions are initially not aligned; that is, $P \neq Q$.

Domain adversarial neural network (DANN) [8] is a framework of choice for UDA. This is a dual-player game involving two key components: the domain discriminator, denoted as D, and the feature representation, denoted as F. In this setup, D is trained to differentiate between the source domain and the target domain, while F is simultaneously trained to both confound the domain discriminator D and accurately classify samples from the source domain. The discrepancy between the feature distributions P_F and Q_F [7] has well corresponding with the error function of the domain discriminator. This is a key to bound the risk associated with the target in the domain adaptation theory [2].

An alternative approach in the field of Unsupervised Domain Adaptation (UDA) focuses on reducing the domain discrepancy as quantified by various metrics, e.g., maximum mean discrepancy(MMD). To establish class-level alignment across domains, the methodology outlined in the study conducted by Pei and colleagues [26] incorporates a multiplicative interaction between feature representations and class predictions. In their studies [3], efforts are made to ensure alignment between the centroids of labeled source data and the centroids derived from pseudo-labeled target data, particularly for shared classes within the feature space.

Another approach to UDA involves employing separate task classifiers for each of the two domains. These classifiers are used to identify non-discriminative features. In turn, they facilitate the learning of a feature extractor that focuses on generating discriminative features [17]. Several other studies emphasize the importance of directing attention towards transferable regions as a means to establish a domain-invariant classification model, as exemplified by [16]. In addition, for the purpose of extracting target-discriminative features, [14] employ techniques such as generating synthetic data from the raw input of the two domains, as described in [25].

Since our work modifies the network and losses of the CDAN framework [22], we explain it here for completeness. To reduce the shift in data distributions across the domains, CDAN trains a deep network $N : x \rightarrow y$, so that source risk $r_s = E_{(x_s,y_s)\sim P}[N(x_s) \neq y_s]$ can bound the target risk $r_t = E_{(x_t,y_t)\sim Q}[N(x_t) \neq y_t]$ plus the distribution discrepancy $disc(P, Q)$ quantified by a novel conditional domain discriminator. In the context of adversarial learning, Generative Adversarial Networks (GANs) [8] play a pivotal role in mitigating differences between domains. A deep network N generates features represented by $f = F(x)$ and classifier prediction denoted by $g = N(F(x))$.

We enhance existing methods for adversarial domain adaptation in two specific ways. Firstly, when dealing with non-identical joint distributions of features and classes across domains, as characterized by $P(x_s, y_s)$ and $Q(x_t, y_t)$, relying solely on the adaptation of the feature representation f may prove insufficient, as highlighted in [22]. A quantitative analysis indicates that deep representations tend to transition from a more general to a domain-specific nature as they traverse deeper layers within neural networks. This transition leads to a notable decrease in transferability, particularly observed in the layers responsible for domain-specific feature extraction (f) and classification (g), as detailed in [37].

Secondly, due to the nature of multi-class classification, the feature distribution is multi-modal, and hence adapting feature distribution may be challenging for adversarial networks.

Simultaneous modeling the domain variances in feature representation (f) and classifier prediction (g) facilitates effective domain gap reduction [22]. This joint conditioning helps capture and align data distributions between source and target domains. Thus, incorporating classifier prediction as a conditioning factor in domain adaptation shows promising potential for enhancing transferability and producing domain-invariant representations in challenging cross-domain scenarios. CDAN originally introduced a minimax optimization framework featuring two adversarial loss terms: (a) the source classifier loss, aimed at minimizing it to ensure a lower source risk, and (b) the discriminator loss applied to both the source classifier N and the domain discriminator D, spanning both the source and target domains. This loss is minimized with respect to D while simultaneously maximized with respect to $f = F(x)$ and $g = N(F(x))$:

$$L_{clc}(x_{s_i}, y_{s_i}) = \mathbb{E}_{(x_{s_i}, y_{s_i}) \sim D_s} L(N(x_{s_i}), y_{s_i}) \tag{1}$$

$$\begin{aligned} L_{dis}(x_s, x_t) = &- \mathbb{E}_{x_{s_i} \sim D_s} \log[D(f_{s_i}, g_{s_i})] \\ &- \mathbb{E}_{x_{t_j} \sim D_t} \log[1 - D(f_{t_j}, g_{t_j})], \end{aligned} \tag{2}$$

In this context, L corresponds to the cross-entropy loss, L_{clc} is the classifier loss, L_{dis} is the discriminator loss and $h = (f, g)$ signifies the combined variable encompassing the feature representation f and classifier prediction g. The minimax game of CDAN is

$$\begin{aligned} &\min_N L_{clc}(x_{s_i}, y_{s_i}) - \lambda L_{dis}(x_s, x_t) \\ &\min_D L_{dis}(x_s, x_t), \end{aligned} \tag{3}$$

Here, λ denotes a hyper-parameter that balances between the two objectives, allowing for a trade-off between source risk and domain adversary concerns.

As depicted in Fig. 1, the domain discriminator D is conditioned on the classifier prediction g via the joint variable $h = (f, g)$, aiming to address the two challenges inherent in adversarial domain adaptation, as discussed in [22]. To incorporate a basic form of conditioning for D, we employ $(D(f \oplus g))$—where we concatenate the feature representation and classifier prediction into a vector $(f \oplus g)$, which serves as the input for the conditional domain discriminator D. This conditioning approach aligns with the common practice observed in existing conditional GANs [8]. However, when utilizing the concatenation approach, f and g operate independently, missing the opportunity to fully capture the crucial multiplicative interactions between the feature representation and classifier prediction that play a pivotal role in domain adaptation. A multilinear map is formed by computing the outer product of multiple random vectors. This technique, which involves multilinear maps applied to infinite-dimensional nonlinear feature maps, has proven successful in embedding joint or conditional distributions into reproducing kernel Hilbert spaces [31]. In addition to the theoretical

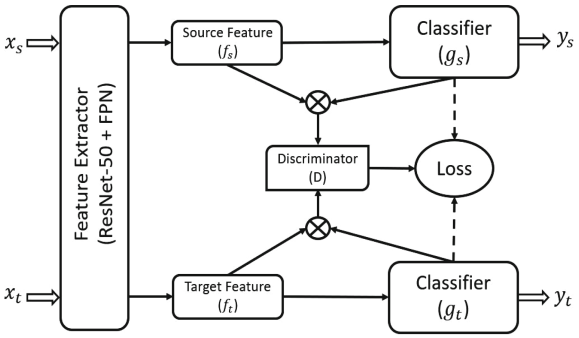


Fig. 1. The CDAN [22] architectural framework, illustrating the joint reduction of the cross-domain gap by the conditional domain discriminator D, with domain-specific feature representation f and classifier prediction g at its core. The symbol $\otimes$ signifies a multilinear mapping operation.

advantages offered by the multilinear map $(x \otimes y)$ in comparison to concatenation $(x \oplus y)$, as discussed in [31]. In this research, we harness the capabilities of the multilinear map to condition D on g. In contrast to concatenation, the multilinear map, denoted as $(x \otimes y)$, excels in capturing the intricate multi-modal structures that are inherent in complex data distributions. However, it's important to note that a drawback of the multilinear map is its potential for dimension explosion.

Our approach involves the joint minimization of the source classifier N and feature extractor F with respect to Eq. (1). Additionally, we minimize Eq. (2) to optimize the domain discriminator D and simultaneously maximize Eq. (2) to enhance the feature extractor F and source classifier N. This yields the mini-max problem of Domain Adversarial Networks:

$$\begin{aligned} \min_{G} \quad & \mathbb{E}_{(x_s^i, y_s^i)\sim D_s} L(G(x_s^i), y_s^i) \\ & + \lambda \left(\mathbb{E}_{x_s^i \sim D_s} \log[D(T(h_s^i))] \right. \\ & \left. + \mathbb{E}_{x_t^j \sim D_t} \log[1 - D(T(h_t^j))] \right) \\ \max_{D} \quad & \mathbb{E}_{x_s^i \sim D_s} \log[D(T(h_s^i))] + \mathbb{E}_{x_t^j \sim D_t} \log[1 - D(T(h_t^j))], \end{aligned} \tag{4}$$

In this context, λ serves as a hyper-parameter responsible for adjusting the weightage between the source classifier and the conditional domain discriminator and G acts as the generator. Meanwhile, $h = (f, g)$ represents the composite variable encompassing both the domain-specific representation f and the classifier prediction g which play pivotal roles in adversarial adaptation. [10] shows the improvement in performance with use of tailored loss function for medical datasets.

3 Proposed Method

In our Unsupervised Domain Adaptation (UDA) strategy for cross-domain classification, we leverage insights from a labeled source domain to enhance target domain performance, despite unlabeled conditions. Our method utilizes a multi-scale optimized neural architecture, ensuring well-separated, multi-modal class distributions. Data augmentation techniques such as flipping, resizing, and normalization are applied for domain consistency. A novel aspect of our approach is a unique loss function that, combined with selected existing ones, minimizes domain discrepancies and aligns feature distributions across multi-class datasets with varying image sizes. For augmented image feature extraction, we employ ResNet-50 coupled with the Feature Pyramid Network (FPN), blending deep feature capture with multi-scale extraction to effectively represent detailed and broad image features, a pioneering application in UDA and image classification.

3.1 Proposed Loss Function

The proposed loss function to train the new architecture in the CDAN [22] framework for improving UDA for image classification can be formulated as follows:

$$\begin{aligned} L = \min_{N} L_{clc}(x_{s_i}, y_{s_i}) - \lambda L_{dis}(x_s, x_t) \\ +\beta L_{IM} + \gamma L_{MCC} + \delta L_{MDD} + \eta L_{PLMMD}, \end{aligned} \tag{5}$$

where λ, β, γ, δ and η are hyper parameters, L_{IM} is the information maximization (entropy) loss, L_{MCC} is minimum class confusion loss, L_{MDD} is maximum mean discrepancy loss, and L_{PLMDD} is a novel pseudo-label maximum mean discrepancy loss. It is worth noting that the original CDAN [22] trained a ResNet (and not ResNet + FPN as proposed) using only L_{clc}, L_{dis}, and L_{IM}. On the other hand, all other individual loss terms have their own specialty and this novel combination of loss significantly surpasses the performance of CNN-based as well as transformer-based models. A detailed description of all the losses, including the proposed L_{PLMDD} are given below.

Information Maximization Loss: The Information Maximization loss is designed to encourage neural networks to learn more informative representations by maximizing the mutual information between the learned features and the input data [15]. By maximizing the mutual information between the empirical distribution of target inputs and the resulting distribution of target labels, which can be formally defined as:

$$
\begin{aligned}
I(p_t; x_t) &= H(\overline{p}_t) - \frac{1}{n_t}\sum_{j=1}^{n_t} H(p_{tj}) \\
&= -\sum_{k=1}^{K} \overline{p}_{tk} \log(\overline{p}_{tk}) + \frac{1}{n_t}\sum_{j=1}^{n_t}\sum_{k=1}^{K} p_{tkj} \log(p_{tkj})
\end{aligned}
\tag{6}
$$

where, $p_{tj} = \text{softmax}(G_c(G_f(x_{tj})))$, $\overline{p}_t = \mathbb{E}_{x_t}[p_t]$, and K is the number of classes. By taking into account $I(p_t; x_t)$, our model is incentivized to learn target features that exhibit tight clustering along with a uniform distribution. This approach is designed to retain discriminative information within the target domain.

Minimum Class Confusion: The minimum class confusion loss, referenced as $\mathcal{L}_{MCC}$ [13], aims to mitigate confusion between various classes represented by indices j and j', where these indices collectively encompass the entire set of classes. Notably, this loss term is focused on the target domain and is intended to minimize the confusion between pairs of classes, such as those denoted by j and j' is given by:

$$C_{jj'} = \hat{\mathbf{y}}_{\cdot j}^{\mathsf{T}} W \hat{\mathbf{y}}_{\cdot j'}^{\mathsf{T}}$$

After standardizing (normalising) the class confusion terms, the ultimate MCC Loss function is defined as:

$$\mathcal{L}_{MCC} = \frac{1}{c}\sum_{j=1}^{c}\sum_{j' \neq j}^{c} |C_{jj'}|, \tag{7}$$

This loss is computed as the summation of all non-diagonal elements within the class confusion matrix. The diagonal elements signify the classifier's level of "certainty," whereas the non-diagonal elements signify the "uncertainty" associated with classification. The MCC loss can be incorporated alongside other domain adaptation techniques.

Maximum Mean Discrepancy: Maximum Mean Discrepancy (MMD) is a kernel-based two-sample statistical test employed to assess the similarity between two distributions. The final loss for a given probability measure P and Q takes the following form:

$$
\begin{aligned}
L_{MMD} &= MMD^2(P, Q) \\
&= \mathbb{E}_P[k(X, X)] - 2\mathbb{E}_{P,Q}[k(X, Y)] + \mathbb{E}_Q[k(Y, Y)]
\end{aligned}
\tag{8}
$$

Pseudo-Label MMD: We propose a novel loss function called pseudo-label maximum mean discrepancy (PLMMD). This loss function takes into account pseudo-labels that can be generated on the target domain samples after the first few training iterations. Doing so strongly conditions the feature alignment on the classes. It is calculated using a procedure similar to that of calculating MMD.

The difference is that we multiply each of the expectations in Eq. 8 with weights that are calculated based on pseudo-labels:

$$L_{PLMMD} = w_{XX}\mathbb{E}_P[k(X,X)] - 2w_{XY}\mathbb{E}_{P,Q}[k(X,Y)] \\ +w_{YY}\mathbb{E}_Q[k(Y,Y)], \tag{9}$$

where w_{XX} represents weight to get similarity within the source domain, w_{YY} are weights for similarity within the target domain, and w_{XY} are weights to get similarity within the source and target domain. For calculating the weights, firstly source and target label data are normalized to account for class imbalances. For each class common to both datasets, dot products of normalized vectors are computed to quantify instance relationships. Calculated dot products are normalized by the count of common classes, ensuring fairness. This returns three weight arrays, representing relationships between instances in the source dataset, target dataset, and source-to-target pairs.

4 Experiments and Results

To validate the efficacy of our model, we undertake extensive investigations on well-established benchmarks and juxtapose our results with those achieved by state-of-the-art UDA methods. We also studied the impact of using a feature pyramid network (FPN) [20] for domain adaptation for classification. Additionally, we studied how the feature (representation) space of the target domain evolves during training. We also studied the contribution of various components of the loss function. We also studied the convergence speed of our method compared to FixBi [24].

4.1 Datasets

To evaluate the proposed method, we conducted experiments on benchmark UDA datasets – including Office-31 [29], Office-Home [36], VisDA-2017 [26], and

Table 1. Comparison with SoTA methods on Office-Home. IDAL(ours) is reported with and without(w/o) FPN. The best performance is marked as bold, and the second best is underlined

Model	A→C	A→textbfP	A→textbfR	C→textbfA	C→textbfP	C→textbfR	P→textbfA	P→textbfC	P→textbfR	R→textbfA	R→textbfC	R→textbfP	Avg.
ResNet-50 [12]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [8]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [22]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MDD [38]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
GVB-GD [5]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
SRDC [32]	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
SHOT [19]	56.9	**78.1**	81.0	67.9	78.4	78.1	67.0	54.6	81.8	73.4	58.1	84.5	71.6
SDAT [28]	58.2	77.1	**82.2**	66.3	77.6	76.8	63.3	57.0	82.2	74.9	**64.7**	86.0	72.2
FixBi [24]	58.1	77.3	80.4	67.7	**79.5**	**78.1**	65.8	**57.9**	81.7	76.4	62.9	**86.7**	72.7
IDAL w/o FPN	58.6	77.2	80.1	69.2	76.4	76.3	70.8	56.9	82.4	77.6	63.6	84.2	72.8
IDAL	**59.8**	77.8	80.8	**69.8**	76.9	77.0	**71.6**	57.4	**82.9**	**78.5**	64.1	85.6	**73.5**

Table 2. Accuracy(%) on DomainNet for UDA (ResNet-101). In each sub-table, the column-wise domains are selected as the source domain and the row-wise domains are selected as the target domain. Highest accuracy is marked as bold and second highest is underline.

ADDA [34]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	11.2	24.1	3.2	41.9	30.7	22.2
inf	19.1	–	16.4	3.2	26.9	14.6	16.0
pnt	31.2	9.5	–	**8.4**	39.1	25.4	22.7
qdr	15.7	2.6	5.4	–	9.9	11.9	9.1
rel	39.5	14.5	29.1	**12.1**	–	25.7	24.2
skt	35.3	8.9	25.2	14.9	37.6	–	25.4
Avg.	28.2	9.3	20.1	8.4	31.1	21.7	19.8

DANN [8]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	15.5	34.8	9.5	50.8	41.4	30.4
inf	31.8	–	30.2	3.8	44.8	25.7	27.3
pnt	39.6	15.1	–	5.5	54.6	35.1	30.0
qdr	11.8	2.0	4.4	–	9.8	8.4	7.3
rel	47.5	17.9	47.0	6.3	–	37.3	31.2
skt	47.9	13.9	34.5	10.4	46.8	–	30.7
Avg.	35.7	12.9	30.2	7.1	41.4	29.6	26.1

MIMTFL [9]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	15.1	35.6	10.7	51.5	43.1	31.2
inf	32.1	–	31.0	2.9	48.5	31.0	29.1
pnt	40.1	14.7	–	4.2	55.4	36.8	30.2
qdr	18.8	3.1	5.0	–	16.0	13.8	11.3
rel	48.5	19.0	47.6	5.8	–	39.4	32.1
skt	51.7	16.5	40.3	12.3	53.5	–	34.9
Avg.	38.2	13.7	31.9	7.2	45.0	32.8	28.1

ResNet-101 [12]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	19.3	37.5	11.1	52.2	41.0	32.2
inf	30.2	–	31.2	3.6	44.0	27.9	27.4
pnt	39.6	18.7	–	4.9	54.5	36.3	30.8
qdr	7.0	0.9	1.4	–	4.1	8.3	4.3
rel	48.4	22.2	49.4	6.4	–	38.8	33.0
skt	46.9	15.4	37.0	10.9	47.0	–	31.4
Avg.	34.4	15.3	31.3	7.4	40.4	30.5	26.6

CDAN† [22]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	20.4	36.6	9.0	50.7	42.3	31.8
inf	27.5	–	25.7	1.8	34.7	20.1	22.0
pnt	42.6	20.0	–	2.5	55.6	38.5	31.8
qdr	21.0	4.5	8.1	–	14.3	15.7	12.7
rel	51.9	23.3	50.4	5.4	–	41.4	34.5
skt	50.8	20.3	43.0	2.9	50.8	–	33.6
Avg.	38.8	17.7	32.8	4.3	41.2	31.6	27.7

MDD† [38]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	20.5	40.7	6.2	52.5	42.1	32.4
inf	33.0	–	33.8	2.6	46.2	24.5	28.0
pnt	43.7	20.4	–	2.8	51.2	41.7	32.0
qdr	18.4	3.0	8.1	–	12.9	11.8	10.8
rel	52.8	21.6	47.8	4.2	–	41.2	33.5
skt	54.3	17.5	43.1	5.7	54.2	–	35.0
Avg.	40.4	16.6	34.7	4.3	43.4	32.3	28.6

SCDA [18]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	18.6	39.3	5.1	55.0	44.1	32.4
inf	29.6	–	34.0	1.4	46.3	25.4	27.3
pnt	44.1	19.0	–	2.6	56.2	42.0	32.8
qdr	30.0	4.9	15.0	–	25.4	19.8	19.0
rel	54.0	22.5	51.9	2.3	–	42.5	34.6
skt	55.6	18.5	44.7	6.4	53.2	–	35.7
Avg.	42.6	16.7	37.0	3.6	47.2	34.8	30.3

CDAN + SCDA [18]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	19.5	40.4	10.3	56.7	46.0	34.6
inf	**35.6**	–	**36.7**	4.5	**50.3**	29.9	**31.4**
pnt	45.6	20.0	–	4.2	56.8	41.9	33.7
qdr	28.3	4.8	11.5	–	20.9	19.2	17.0
rel	**55.5**	22.8	53.7	3.2	–	42.1	35.5
skt	**58.4**	21.1	**47.8**	10.6	56.5	–	**38.9**
Avg.	44.7	17.6	38.0	6.6	48.2	35.8	31.8

MDD + SCDA [18]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	20.4	**43.3**	**15.2**	**59.3**	**46.5**	**36.9**
inf	32.7	–	34.5	**6.3**	47.6	29.2	30.1
pnt	46.4	19.9	–	8.1	**58.8**	**42.9**	35.2
qdr	31.1	**6.6**	**18.0**	–	**28.8**	**22.0**	**21.3**
rel	55.5	23.7	52.9	9.5	–	**45.2**	**37.4**
skt	55.8	20.1	46.5	**15.0**	**56.7**	–	38.8
Avg.	44.3	18.1	**39.0**	**10.8**	**50.2**	**37.2**	**33.3**

Table 3. Accuracy (%) on DomainNet for UDA with IDAL(ours). Highest accuracy is marked as bold and second highest is underline.

	clp	inf	pnt	qdr	rel	skt	Avg.
clp	–	**20.5**	36.8	9.2	54.8	44.5	33.2
inf	32.4	–	25.8	2.5	31.2	**41.2**	26.6
pnt	**72.5**	**25.0**	–	4.5	52.7	37.6	**38.5**
qdr	**47.4**	5.3	8.4	–	11.8	12.5	17.1
rel	51.5	**25.2**	**54.5**	2.9	–	34.8	33.8
skt	49.8	**24.8**	47.4	12.8	51.6	–	37.3
Avg.	**50.7**	**20.2**	34.6	6.4	40.4	34.1	31.1

DomainNet [27]. The details of the datasets and transfer tasks on these datasets are given below:

The Office-Home dataset is a key benchmark with 15,500 images across 65 classes and four domains: Artistic, Clip Art, Product, and Real-World, used to assess twelve transfer tasks. Office-31, another pivotal dataset, contains 4,110 images in 31 classes from Amazon, Webcam, and DSLR domains, evaluating six transfer tasks. VisDA-2017, aimed at cross-domain generalization, includes Synthetic and Real source domains with 12 categories, using the ImageNet validation set as the target. DomainNet, the largest dataset for domain adaptation, features about 0.6 million images across 345 categories from six domains (Clipart, Infograph, Painting, Quickdraw, Real, and Sketch), supporting 30 adaptation tasks, showcasing its scale and diversity in visual domain adaptation challenges.

Table 4. Comparison with SoTA methods on VisDA-2017. IDAL(ours) is reported with and without(w/o) FPN. The best performance is marked as bold, and the second best is underlined.

Model	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ResNet-50 [12]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
BNM [4]	89.6	61.5	76.9	55.0	89.3	69.1	81.3	65.5	90.0	47.3	89.1	30.1	70.4
MCD [30]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SWD [17]	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
FixBi [24]	**96.1**	87.8	**90.5**	**90.3**	**96.8**	95.3	**92.8**	**88.7**	**97.2**	**94.2**	**90.9**	25.7	87.2
IDAL w/o FPN	94.1	88.6	89.2	78.7	94.9	98.2	88.5	84.6	94.7	90.3	88.4	51.3	86.8
IDAL	94.7	**89.0**	89.6	79.0	95.6	**98.7**	89.4	85.2	95.6	90.5	88.9	**52.6**	**87.4**

Table 5. Comparison with SoTA methods on Office-31. IDAL(ours) is reported with and without(w/o) FPN. The best performance is marked as bold, and the second best is underlined.

Method	A → D	A → W	D → W	W → D	D → A	W → A	Avg
ResNet-50 [12]	68.9	68.4	96.7	99.3	62.5	60.7	76.1
DANN [8]	79.7	82.0	96.9	99.1	68.2	67.4	82.2
CDAN [22]	92.9	94.1	98.6	100.0	71.0	69.3	87.7
MDD [38]	93.5	94.5	98.4	100.0	74.6	72.2	88.9
GVB-GD [5]	95.0	94.8	98.7	100.0	73.4	73.7	89.3
SRDC [32]	**95.8**	95.7	99.2	100.0	76.7	77.1	90.8
SHOT [19]	93.1	90.9	98.8	99.9	74.5	74.8	88.7
f-DAL [1]	94.8	93.4	99.0	100.0	73.6	74.6	89.2
FixBi [24]	95.0	**96.1**	**99.3**	100.0	**78.7**	**79.4**	**91.4**
IDAL w/o FPN	94.4	95.0	99.0	100.0	75.6	76.6	90.1
IDAL	95.6	95.7	99.1	**100.0**	77.3	77.1	90.8

4.2 Implementation Details

All the experiments were conducted on an NVIDIA A100 in PyTorch, using the CNN-based neural network (ResNet-50) pre-trained on ImageNet [11] and feature pyramid network as the backbone for our proposed model. The base learning rate is 0.00001 with a batch size of 32, and we train models by 50 epochs. The hyper-parameters were $\beta = 0.05, \gamma = 0.1, \delta = 0.15$ and $\eta = 0.15$ for the experiment of Office-31 dataset. Similarly for Office-Home, the hyper-parameters were $\beta = 0.05, \gamma = 0.21, \delta = 0.25$ and $\eta = 0.25$, for the VisDA-2017 dataset the hyper parameters were $\beta = 0.05, \gamma = 0.3, \delta = 0.25$ and $\eta = 0.25$, and for the Domain-Net dataset the hyper parameters were $\beta = 0.05$, $\gamma = 0.01, \delta = 0.2$ and $\eta = 0.25$. We have used AdamW [23] with a momentum of 0.9, and a weight decay of 0.001 as the optimizer. We adhere to the standard procedure for unsupervised domain adaptation (UDA), wherein we make use of both labeled source samples

and unlabeled target samples during the training process. For a fair comparison with prior works, we also conduct experiments with the same backbones such as ResNet-50 [12], DANN [8], CDAN [22], MDD [38], GVB-GD [5], SRDC [32], FixBi [24], SHOT [19], SDAT [28], f-DAL [1], BNM [4], MCD [30] and SWD [17] for demonstration of results with different-different datasets.

4.3 UDA Benchmarks

We assess the performance of our proposed model by conducting comparisons with state-of-the-art methods that rely on ResNet-based architectures. In these experiments, we employ ResNet-50 as the underlying architecture for our evaluations across the Office-Home, Office-31 and VisDA-2017 datasets and ResNet-101 for the DomainNet dataset. Importantly, each ResNet-50 and ResNet-101 backbone is trained exclusively on source data and subsequently subjected to testing using target data.

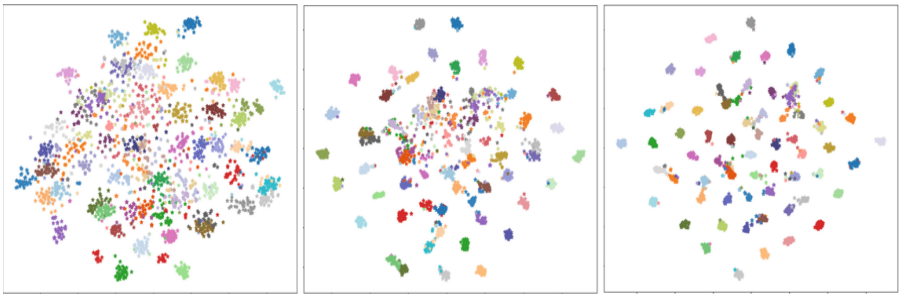


Fig. 2. Two-dimensional visual representation, generated using t-SNE [35], illustrates the evolution of the domain adaptation task from R to P on the Office-Home dataset. The images depict untrained *target* samples (leftmost), training progress after 4 epochs (middle), and the state after 9 epochs (rightmost). Initially, overlapping classes gradually reconfigure into distinct clusters during the training process.

Table 1 presents quantitative results with various backbones, demonstrating our proposed model's consistent superiority over the state-of-the-art, specifically achieving an impressive average accuracy improvement of over 0.8% when compared to the FixBi model for the Office-Home dataset. Table 4 further showcases our model's superiority, surpassing the current state-of-the-art and attaining a notable 0.2% average accuracy improvement over the FixBi model for the VisDA-2017 dataset. Table 5 illustrates results using diverse backbones, highlighting our model's performance, which is comparable with the current state-of-the-art for the Office-31 dataset. Table 2 and 3 depict the performance of our model with ResNet-101 + FPN as the feature extractor for the most challenging dataset of domain adaptation with comparable performance to state-of-the-art.

Table 6. Performance comparison of IDAL with different combinations of losses on Office-Home. The best performance is marked as bold.

M1	MMD	MCC	PLMMD	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
✓	×	×	×	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
✓	✓	×	×	59.4	76.8	80.3	69.1	75.7	76.2	69.7	56.8	82.3	78.4	63.4	84.6	72.7
✓	✓	✓	×	59.6	77.6	80.4	69.3	76.0	76.8	71.2	57.1	82.7	78.5	64.0	85.2	73.2
✓	✓	✓	✓	**59.8**	**77.8**	**80.8**	**69.8**	**76.9**	**77.0**	**71.6**	**57.4**	**82.9**	**78.5**	**64.1**	**85.6**	**73.5**

ResNet-50 combined with Feature Pyramid Network (FPN) for feature extraction offers multi-scale feature capture, superior object detection, and effective feature fusion. This versatile pairing, proven in various computer vision tasks, balances depth and scale, enhancing overall performance while reducing computational costs. Tables 1, 4, and 5 show the impact of having FPN with ResNet for feature extraction. We conducted ablation studies to understand the impact of the different feature extractors such as ConvMixer [33] and ResNet-101 [12]. However, the performance in these cases was worse than our reported results. We also compare the ResNet-based backbone and transformer-based backbone and notice a huge gap in parameter requirements. ResNet-based backbone needs relatively very less parameters compared to transformer-based backbones.

The tSNE [35] plot is shown in Fig. 2 for the office-Home dataset for the task of domain adaptation when R is the source and P is the target. We can see the rapid evolution of a multi-modal distribution of the target domain features where classes (denoted by separate colors) get separated.

4.4 Impact of Loss Components

To gauge the influence of individual loss functions and their collective impact, we conducted a thorough experimental analysis. Our findings revealed that Minimum Class Confusion (MCC) loss functions enhance classification models by reducing class confusion, especially in scenarios with imbalanced class distributions. Concurrently, we observed that information maximization losses assist the classifier in prioritizing the most confidently aligned samples for domain adaptation. Additionally, the Maximum Mean Discrepancy (MMD) loss effectively narrows the gap between the mean embeddings of the two distributions. Table 6 shows the effect of the individual loss function on the performance of our model IDAL for the Office-Home dataset and it indicates that our model IDAL performs best with a tailored combination of loss functions. By artfully combining these distinctive loss functions, we not only surpass the current state-of-the-art but also achieve a comprehensive solution that advances the field of classification models in diverse scenarios.

5 Conclusions and Future Directions

We proposed a novel method for unsupervised domain adaptation for image classification. We proposed a novel neural network architecture and a loss function. Architecturally, we have demonstrated that synergy between two deep learning architectures – ResNet [11] and feature pyramidal network (FPN) [20] – complement each other to extract multi-scale features and effectively separate style (domain) and content (class) information components. Our ablation studies confirm the importance of using FPN with ResNet. The proposed loss component PLMMD and judiciously chosen existing loss components leads to significant improvements in unsupervised domain adaptation (UDA) performance that can surpass the performance of CNNs using other UDA methods. Our ablation study confirmed the importance of each of the loss components. Additionally, using the proposed loss led to faster convergence and a rapid evolution of a class-wise multi-modal distribution of the target domain features.

In the future, computationally heavier architectures, such as, vision transformers [6] and its derivatives may be used for further improvements in domain adaptation. Additionally, the proposed loss function may be adapted for other tasks, such as semantic segmentation and object detection.

References

1. Acuna, D., Zhang, G., Law, M.T., Fidler, S.: f-domain adversarial learning: theory and algorithms. In: International Conference on Machine Learning, pp. 66–75. PMLR (2021)
2. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. **79**, 151–175 (2010)
3. Chen, C., et al.: Progressive feature alignment for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 627–636 (2019)
4. Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., Tian, Q.: Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3941–3950 (2020)
5. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12455–12464 (2020)
6. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. PMLR (2015)
8. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2030–2096 (2016)
9. Gao, J., Hua, Y., Hu, G., Wang, C., Robertson, N.M.: Reducing distributional uncertainty by mutual information maximisation and transferable feature learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 587–605. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_35

10. Gupta, R.K., Das, S., Sethi, A.: Domain-adaptive learning: unsupervised adaptation for histology images with improved loss function combination. arXiv preprint arXiv:2309.17172 (2023)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Jian, S., Kaiming, H., Shaoqing, R., Xiangyu, Z: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 770–778 (2016)
13. Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 464–480. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_28
14. Kang, G., Zheng, L., Yan, Y., Yang, Y.: Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 401–416 (2018)
15. Krause, A., Perona, P., Gomes, R.: Discriminative clustering by regularized information maximization. In: Advances in Neural Information Processing Systems, vol. 23 (2010)
16. Kurmi, V.K., Kumar, S., Namboodiri, V.P.: Attending to discriminative certainty for domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 491–500 (2019)
17. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10285–10295 (2019)
18. Li, S., et al.: Semantic concentration for domain adaptation. CoRR arxiv:2108.05720 (2021)
19. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International Conference on Machine Learning, pp. 6028–6039. PMLR (2020)
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
21. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105. PMLR (2015)
22. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
24. Na, J., Jung, H., Chang, H.J., Hwang, W.: Fixbi: bridging domain spaces for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1094–1103 (2021)
25. Pavanetto, S., Brambilla, M.: Generation of realistic navigation paths for web site testing using RNN and GAN. J. Web Eng. **20**(8), 2571–2604 (2021)
26. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
27. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1406–1415 (2019)

28. Rangwani, H., Aithal, S.K., Mishra, M., Jain, A., Radhakrishnan, V.B.: A closer look at smoothness in domain adversarial training. In: International Conference on Machine Learning, pp. 18378–18399. PMLR (2022)
29. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
30. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
31. Song, L., Huang, J., Smola, A., Fukumizu, K.: Hilbert space embeddings of conditional distributions with applications to dynamical systems. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 961–968 (2009)
32. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8725–8735 (2020)
33. Trockman, A., Kolter, J.Z.: Patches are all you need? arXiv preprint arXiv:2201.09792 (2022)
34. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. CoRR arxiv:1702.05464 (2017)
35. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(11) (2008)
36. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5018–5027 (2017)
37. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, vol. 27 (2014)
38. Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, vol. 97 of Proceedings of Machine Learning Research, pp. 7404–7413. PMLR (2019)

Large Multimodal Models Thrive with Little Data for Image Emotion Prediction

Peng He[3(✉)], Mohamed Hussein[1], and Wael Abd Almageed[2]

[1] Information Sciences Institute, University of Southern California, Los Angeles, CA 90292, USA
mehussein@isi.edu

[2] Electrical and Computer Engineering Department, Clemson University, Riggs Hall, Clemson, SC 29634, USA
wabdalm@clemson.edu

[3] Los Angeles, USA
penghe000@outlook.com

Abstract. Images can convey rich semantics and induce a wide range of emotions in viewers. However, predicting induced emotions from images can be challenging due to the subjective nature of emotions and the variability in how different viewers perceive them. Existing methods for image emotion prediction rely on neural networks to learn an image-to-emotion mapping. Such methods require large amounts of annotated training data to achieve good generalization. In this paper, we show that it is possible to train a model that generalizes better than state-of-the-art methods with significantly less data. Our method leverages the power of a pre-trained large multimodal model (LMM) with the addition of a shallow adapter module that transforms the LMM's output embedding to a classification output. On three out of four benchmark datasets, our method outperforms the previous state of the art (SOTA) results by a significant margin, with one even showing around a 9% accuracy improvement. Additionally, our method achieves a new SOTA with only 20% of the data on these three datasets, and improves further using more data. On the fourth dataset, which is the smallest one, our method is on par with the SOTA. Moreover, our method can naturally provide human-readable intermediate results, which could serve as textual explanations of the classification outputs. The code is available at https://github.com/vimal-isi-edu/LMM_Emotion_Prediction.

Keywords: Emotion Prediction · Vision · Large Multimodal Model

1 Introduction

The widespread use of social networks has led people to widely share their lives and express their opinions using images and text. The rapid growth of user-

P. He—This work was done during Peng He's employment at USC ISI.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 298–313, 2025.
https://doi.org/10.1007/978-3-031-78107-0_19

generated content has created a high demand for applications in which it is essential to understand the user's data at an emotional level, such as business intelligence [6] and opinion mining [28], as individuals' emotions can directly affect their decision making. This demand has significantly expedited the research on emotion prediction. In this paper, we focus on the problem of predicting emotions from images.

Text-based emotion prediction [5,7,35], is a relatively easier problem than image-based emotion prediction. Indeed, predicting emotions from images is a particularly challenging problem. The problem is subjective in nature. Image-induced emotions can be influenced by many factors, such as cultural background, personal experiences, and current mental state. This subjectivity also makes it difficult to collect large manually-labeled datasets, making the problem even more difficult. In fact, large datasets in this domain are typically either weakly annotated by web tags or are known to be biased [18]. To address the subjectivity issue, some datasets are annotated with multiple labels per image or with a probability distribution on the labels [10,21]. However, posing the problem as multi-label classification introduces more challenges in solving the problem and evaluating the solutions.

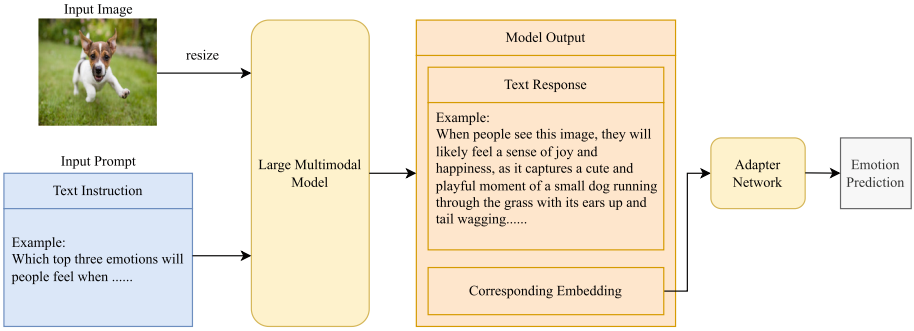


Fig. 1. Overview of our pipeline. The LMM takes both a resized image and a text prompt as input and outputs a text response and its corresponding embedding. The embedding is then used as input to the adapter network to perform emotion prediction.

Recent approaches to image emotion prediction either treat the problem as image classification and apply neural networks to learn a mapping from images to emotion labels in an end-to-end manner [13,34], or use an image encoder along with a pre-trained language model to merge or align features within a shared feature space, which is subsequently used for making predictions [9,37]. There are two main weaknesses in the existing approaches. First, they are data-intensive, which is especially problematic given the scarcity of large manually labeled data in the image emotion prediction community. Second, while the approaches that use the vision encoder and pre-trained language models benefit

from language models' rich semantic space that brings generalizability of unseen concepts, their vision encoders are either trained from scratch on limited amounts of data, which limits their potential for generalization, or trained in separation from the text encoder, which may not leverage the cross-modality interaction when combining features from both domains. In this paper, our objective is to address the aforementioned limitations by designing a pipeline that not only leverages the benefit of language models but also harnesses the strength of strong vision encoders. To demonstrate its generalization, this pipeline should perform well on testing data without extensive training data.

In recent years, large language models (LLMs) have unlocked immense potential in the field of artificial intelligence, as evidenced by many examples, such as GPT-3 [2] and GPT-3.5. These models have the ability of long-range context understanding, generalization, and reasoning, thanks to their training on large-scale textual corpora, paving the way for groundbreaking advancements. Among these models, the LLaMA series [26,27] stands out as models that outperform GPT-3 and strive to enhance the alignment capabilities of LLMs to better comprehend and follow human instructions. Follow-up works extend the success of LLMs from the text domain to the image-text domain and create multimodal vision-language models [12,16] or large multimodal models (LMMs). LMMs combine image and text data, enabling machines to understand and process both modalities simultaneously. These models have shown promising results in various applications, such as summarizing documents with images [16] and visual question answering [3].

In this paper, we make the first attempt at predicting induced emotions from images using a pre-trained LMM as shown in Fig. 1. Relying on the LMM, our approach benefits from the extensive training data used to train such models to address the generalizability issue. Another advantage of using the LMM is that we can obtain model's explanations regarding its decision through their textual output. The contributions of this paper can be summarized as follows:

- We introduce a novel method employing LMMs for image emotion prediction.
- We show that our approach can outperform previous SOTA methods while being trained on much less data on four datasets.
- We conduct ablation studies showing the importance of different components of our model and studying the effect of prompt choice across four general prompt categories and ten specific variations.

2 Related Work

In this section, we focus on the literature review of image emotion prediction by exploring the progression of building image-to-emotion mappings and the integration of language models.

2.1 Image-to-Emotion Mapping

The exploration of how images induce emotions has a long and rich history. Early works [14,30] use low-level image features such as color pattern, shape, and

texture, and train classical classification models, such as support vector machines (SVMs), for emotion prediction. Researchers then started to invite the idea of using high-level information, such as scene information or shape geometry [1,20] and modeling their relationships with emotions. More recently, with the upsurge of neural networks, there is an increasing inclination to employ them as tools for feature extraction, which has led to the adoption of various architectural designs, such as traditional CNN [21], multilevel feature fusion CNN [4] and graph neural network [29]. With neural networks, accuracy has improved significantly over the use of hand-crafted features.

2.2 Language Models Integration

Before integrating the language models, the problem that caught most of the attention was data quality. With the success of neural network-based feature extractors, the importance of data has been highlighted. However, a biased dataset can easily affect the generalizability and performance of the model [18]. To improve the dataset quality, one line of research on datasets provides additional information to help a trained model focus on the parts that induce the emotion, thereby minimizing potential distractions. For example, Peng et al. [22] created the EmotionROI dataset, which offered an emotion stimuli map with each image to facilitate finding the "affective regions" that align better with emotion concepts. Kosti et al. [10] created the multi-label EMOTIC dataset, including images featuring people with the corresponding bounding boxes and contextual scene information. Other researchers aimed to address the dataset bias problems with larger and more diverse datasets. You et al. collected a dataset [33] of over 3 million weakly labeled images by searching Flickr and Instagram images for eight emotion keywords and then curating 23,000 images with human annotations via Amazon Mechanical Turk (AMT). Panda et al. collected WEBEmo [18] with 268,000 images from a stock image database. The dataset includes three levels of labels.

Later, with the success of multimodal representation learning leveraging image-text pairs, such as CLIP [23], VinVL [36], and ALIGN [8], researchers realized that text input can play a vital role in computer vision tasks. The semantic meaning in the text input improves the traditional image classification pipeline by introducing the generalizability of unseen concepts, which has demonstrated effectiveness in downstream tasks such as zero-shot image classification [23]. Based on this idea of multimodal models, later works, such as [9], included language models in the emotion prediction pipeline to perform feature alignment between semantic textual embeddings and embeddings from a vision encoder. Currently, with the help of LMMs [12], we can take advantage of their training on very large numbers of image-text pairs. This gives us a well-trained vision encoder embedded in its model architecture, enabling a more comprehensive integration of image and text features during the early stages of the model.

3 Method

In this section, we first present the overview of our idea and introduce our emotion prediction architecture in Sect. 3.1, then explain the adapter network module we use to predict the emotion in Sect. 3.2. We will cover the prompt we use to trigger the model in Sect. 4.2 and discuss all the prompts we have used and their performances in Sect. 4.4.

3.1 Overview

As demonstrated in Fig. 1, our general prediction architecture has two parts: the first part is to use a text prompt to obtain the LMM's response and embedding regarding an input image, and the second part is to use an adapter network to process the LMM's embedding and output the emotion label. We can see from the example in Fig. 1 that the LMM can provide the emotion label and an explanation in its response. However, because of the subjectivity of emotion labeling and the training method of LMMs, the LMM forms a specific "opinion" about the image, which may differ from the labelers' views. In other words, the bias of the output of the LMM may not be consistent with the bias in the labeling process of a specific dataset. To bridge this gap, we employ a shallow adapter network that transforms the LMM's response to the dataset's labels. We choose to use the LMM's last hidden state as the input of the adapter network instead of using another sentence embedding model.

3.2 Adapter Network

As we explained above, the LMM's output needs to be transformed to the specific dataset's labeling to bridge the bias gap between the model's response and the dataset's labelers' responses. To accomplish this, we employ an adapter network. However, we need to have a suitable embedding for the model's response. Following the setting of many text embedding and classification tasks from natural language processing [24,25], we use the last hidden state of the LMM to construct the desired embedding. The dimensions of the last hidden state are (SEQUENCE_LENGTH, EMBEDDING_DIM). To mitigate the impact of the token sequence order, we compute the average across the first dimension resulting in an embedding of length EMBEDDING_DIM. To explore alternative possible ways, we attempted different methods to obtain the embedding utilizing the last hidden state, but observed no improvement in performance. We then use a shallow neural network as our adapter model.

4 Experiments

In this section, we first describe the datasets we use to evaluate our method in Sect. 4.1, then introduce the details of our implementation in Sect. 4.2. We then explain the experimental results in Sect. 4.3, followed by the ablation study in Sect. 4.4. Finally, we discuss our findings in Sect. 4.5.

4.1 Datasets

We evaluate the performance of our method on the following datasets. These datasets cover a wide range of scales and exhibit diversity across multiple domains. To show the generalizability of our approach, we will not only apply the experimentation settings followed in the previous SOTA methods, but also include varying proportions of the training data, spanning from 10% to 80% of the entire data, without changing the testing proportions from what have been used in prior work.

Emotion6. This dataset [21] is collected from Flickr by using Ekman's six basic emotions keywords and their synonyms. Each emotion category evenly contains 330 images (1,980 images in total). In our settings, we use the setup in [32,37] and treat this dataset as a single-label dataset with 80% of the data used for training and 20% used for testing.

UnBiasedEmotion. UnBiasedEmotion [18] is collected from Google using six basic-level emotions proposed in [19], making 3,045 images in total. We use the experimental setup in [31] with 80% of the images used for training and 20% used for testing.

Emotion-6. Emotion-6 dataset [18] contains 8,350 images labeled with six emotions proposed in [19] collected from 150K images from Flickr and Google, and labeled by five human subjects. We use the setup in [13,34] with 80% of the images used for training and 20% used for testing.

FI8. FI8 dataset [33] contains 23,308 images labeled with eight emotion labels derived from [15]. This dataset is queried online from Flickr and Instagram by the corresponding eight emotion keywords, and each image is labeled by five Amazon Mechanical Turk (AMT) workers, ensuring a minimum agreement of three workers on each label. As some images no longer exist on the Internet, only 23,185 images were crawled for our experiments. We use the split setting in [13,34] and randomly select 80% of the data for training, 5% for validation and the remaining 15% for testing.

4.2 Implementation Details

LMM Inference. We use the LLaVA model proposed in [12] as our LMM. Specifically, we use version 0 of this model. Its integration with Hugging Face allows us to use Hugging Face's general API to perform inference. To ensure the reproducibility of the result, we turn off the sampling and use beam search instead. Specifically, we use five beams for beam search. We also set the maximum new number of output tokens to 1024. Following the suggested pre-processing steps for this model, we first read raw image files in RGB channel format and resize their shapes to 224×224. Notably, the average inference time for one resized image is 7.101 s

Prompt to Trigger the Model. In this work, the prompt we use to trigger the model is: *Which top three emotions will people feel when they see this image? Choose answers from the list: [LABEL_LIST]. In the first sentence of answer, give the list of emotions. In the following sentences, provide your explanation.*

The *[LABEL_LIST]* means the list of labels in the target dataset. To make the order of this label list constant, we first sort them in alphabetical order. The reason for asking the model to output multiple labels and to explain is to trigger the model so that its answer will be more rational and more likely to be correct and reasonable.

We will cover all the prompts we have used in Sect. 4.4.

Adapter Network. We used a single linear layer with softmax activation function in our implementation of the adapter network. We also experimented with deeper networks, different activation functions, with and without residual connections, with no gain in performance.

Experiment Setup. We conducted our experiments using PyTorch on a single NVIDIA RTX 1080 GPU. To remove the influence of random seeds, we repeated each experiment ten times and reported the average and standard deviation. Our chosen optimizer is stochastic gradient descent (SGD) with a learning rate of 0.03 and momentum 0.9. The training process uses 50 epochs, with a batch size set at 1024. We also experimented with a different optimizer and different learning rates with no gain in performance.

Table 1. Emotion prediction test results on the previous SOTA method and our method. The "vg" and "td" scores of our method listed in the table are the average accuracy and its standard deviation of accuracy over ten experiments following the dataset split setting outlined in the corresponding SOTA paper to remove the influence of random seeds.

Datasets	Previous SOTA (%)	Our Method (%)	
		Avg	Std
Emotion6	72.73 [17]	**73.38**	1.74
UnBiasedEmotion	85.90 [4]	**92.59**	0.57
Emotion-6	65.01 [13]	**69.98**	0.79
FI8	79.21 [17]	**88.45**	0.73

4.3 Results

We compare the results of previous SOTA approaches and our results, as outlined in Table 1, following their dataset split setting. Additionally, we present the results achieved under different split ratios of the train set while keeping the

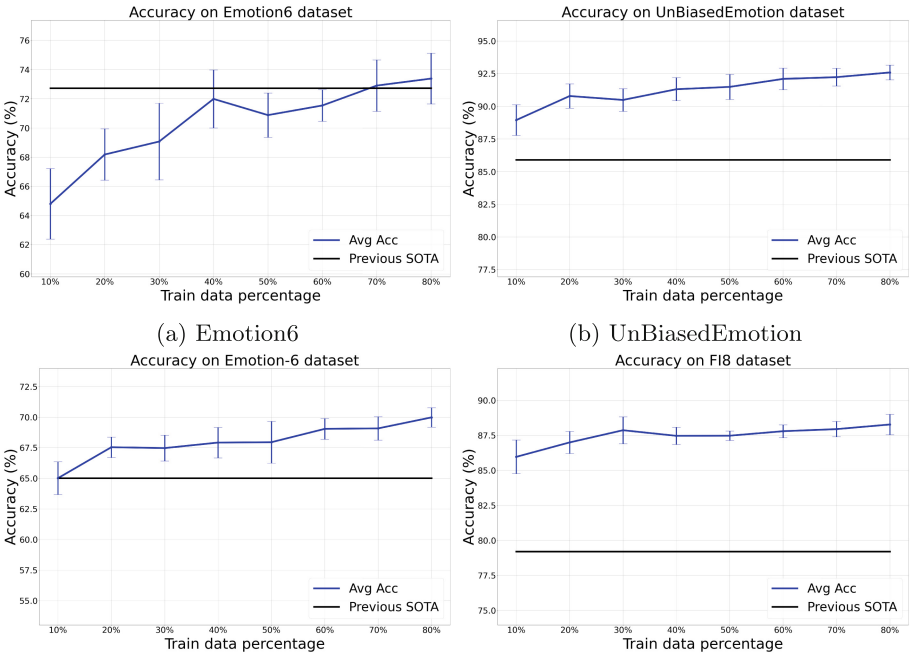


Fig. 2. The emotion prediction average test accuracy and standard deviation of our approach evaluated on four datasets, spanning training data ratios from 10% to 80% while keeping the testing data's size fixed with the previous SOTA setting. In each plot, the solid blue line is the average of ten experiments. We also include the result of the previous SOTA as a black line.

size of the test set fixed in Fig. 2 to study how the performance of our method is affected by the amount of training data used. Upon analyzing the results in the Table 1, it is evident that our method's average performance across ten experiments significantly outperforms the previous SOTA methods in UnBiasedEmotion, Emotion-6 and FI8 dataset by a substantial margin. On the largest FI8 dataset, we are around 9% better than the previous SOTA. The standard deviations for these datasets remain below 0.8, indicating the consistency of our method. On the smallest dataset, which is Emotion6, our method beats the previous SOTA by 0.65 on average, but this advantage is not consistently obtained because it is less than the standard deviation for our method. In Fig. 2, where we experimented with different dataset split settings, we observe the exceptional ability of our method to generalize. We can see that even with as little as 20% of the data, the average performance across ten experiments can surpass the previous SOTA results with a margin bigger than the standard deviation on three datasets.

4.4 Ablation Study

Table 2. Different versions of prompts we used to trigger the LMM. The placeholder `LABEL_LIST`, present in Type III and Type IV prompts, means the list of labels in the target dataset. To make the order of this label list constant, we first sort its elements in alphabetical order.

Type	Index	Prompt Content
I	1	Describe this image from an emotional perspective
II	2	Which emotion will people feel the most when they see this image?
	3	Which emotion will people feel the most when they see this image? First, provide the step by step explanation of your answer. Then, provide your answer
	4	Predict the most likely emotion people will feel when they see this image.
III	5	Which emotion will people feel the most when they see this image? Choose the answer from the list: [LABEL_LIST]. Only give the chosen emotion
	6	Predict the most likely emotion people will feel when they see this image. You need to choose the answer from the list: [LABEL_LIST]. Give the name of the chosen emotion in the first sentence of your answer. Then, provide your explanation in the following sentences
	7	Predict the most likely emotion people will feel when they see this image. You need to choose answers from the list: [LABEL_LIST]. Your response should take into account the visual cues, context, and any other relevant information. Remember to give the name of the chosen emotion in the first sentence of your answer. Then, provide your explanation in the following sentences
IV	8	Which top three emotions will people feel when they see this image? Choose answers from the list: [LABEL_LIST]. Only give the list of chosen emotions
	9	Which top three emotions will people feel when they see this image? Choose answers from the list delimited by square brackets. [LABEL_LIST]
	10	Which top three emotions will people feel when they see this image? Choose answers from the list: [LABEL_LIST]. In the first sentence of answer, give the list of emotions. In the following sentences, provide your explanation

Prompt Design. As mentioned in Sect. 3, here we will show the overall performance of different prompts we have attempted. We deploy multiple prompt versions to trigger a response from the LMM. In general, our prompts are split into four types, which do one of the following: (1) Ask the model to describe the image from an emotional perspective (2) Ask the model to predict the emotion people will feel when they see this image without giving the model any further information about the dataset. (3) Ask the model to do the same while providing

it with the dataset's label list and asking the model to choose an answer from the list. (4) Similar to (2), but ask the model to provide multiple answers from the list instead of just one. In total, we have ten different prompt versions across the four types. We evaluate the performance of our pipeline on all ten versions. Table 2 includes all prompt versions used in our experiments. From the results in Table 3, we can see that the accuracy is stable across different prompt types on three datasets. For the FI8 dataset, the choice of prompt has a huge influence over the accuracy, indicating the importance of the prompt choice for LMMs. However, once the prompt is fixed, the accuracy is very stable under multiple repetitions.

Table 3. Emotion prediction test accuracy for each prompt in Table 2. The accuracy is the average over ten experiments following the dataset split setting outlined in the corresponding SOTA paper to remove the influence of random seeds.

Datasets	Acc (%) by prompt index									
	1	2	3	4	5	6	7	8	9	10
Emotion6	73.01	73.01	72.95	74.22	72.95	72.10	73.84	73.43	73.38	73.38
UnBiasedEmotion	92.96	92.46	92.71	92.81	93.12	92.13	92.50	92.40	92.41	92.59
Emotion-6	69.69	69.55	70.02	69.97	69.61	69.65	70.60	68.98	69.21	69.98
FI8	77.76	77.77	77.38	77.67	78.79	77.47	77.56	77.13	77.45	88.27

Choice of LMM. Currently, there are lots of open source instruction fine-tuned LMMs available in the community, and we will compare the performance between two of such LMMs using the same experimental settings: LLaVA, which is the one we use in the main experiments, and BLIP2 [11], which is another widely used LMM, in Table 4. From the documentations for both LMMs, the LLaVA model has around 13 billion parameters and BLIP2 has around 2.7 billion parameters. Despite being almost 80% smaller, the BLIP2 model's results are comparable to the SOTA results in half of the evaluation datasets. However, there is a clear positive correlation between the size of the LMM and its accuracy performance, suggesting that the model's scale is important for optimal accuracy.

Importance of the Adapter Network. To show the importance of the adapter network, we compare the performance between our current pipeline with k-nearest neighbor (kNN) results without the adapter network in Table 5. From the results, we can see that the accuracy of our method surpasses the kNN results by a large margin. Nonetheless, it is worth noting that the kNN approach beats the previous SOTA results in three out of the four datasets, which again highlights the effectiveness of LMMs for image emotion prediction.

Table 4. Emotion prediction test accuracy on four datasets using LLaVA and BLIP2 as the LMM modules using the same prompt. The accuracy is the average over ten experiments following the dataset split setting outlined in the corresponding SOTA paper to remove the influence of random seeds.

Datasets	Acc (%)	
	LLaVA	BLIP2
Emotion6	73.38	71.99
UnBiasedEmotion	92.59	87.54
Emotion-6	69.98	55.95
FI8	88.45	72.78

Table 5. Emotion prediction test accuracy on four datasets following our method and using kNN instead of adapter network to prove the importance of the adapter network. The accuracy is the average over ten experiments following the dataset split setting outlined in the corresponding SOTA paper to remove the influence of random seeds. N means number of neighbor in the kNN

Datasets	Acc (%)				
	Adapter	kNN			
		N = 3	N = 5	N = 10	N = 25
Emotion6	73.38	65.10	64.92	67.55	66.19
UnBiasedEmotion	92.59	87.95	89.10	87.91	86.80
Emotion-6	69.98	61.68	64.17	65.08	65.14
FI8	88.45	85.62	85.51	86.76	85.88

4.5 Discussion

LMM's Ability to Follow Instructions. As shown in Table 2, for Type III and Type IV prompts, we not only ask the LMM to predict the induced emotion(s), but also require it to choose one or more answers from the provided label list. However, from its responses we observe that the LMM does not always follow such instructions. We present four examples using the UnBiasedEmotion dataset in Fig. 3. The first two examples employ the Type III prompt. In the first example, the LMM can choose the emotion "sadness" from the label list in the prompt, but it ignores the instruction "Only give the chosen emotion". In the second example, the LMM outputs the emotion "warmth", which is not in the label list, and again it disregards the requirement to provide only the emotion. The following two examples use the Type IV prompt. In the third example, the LMM can follow instructions as it chooses the three emotions from the label list in the prompt and provides explanations. In the fourth example, the LMM outputs the emotion "excitement", which is not in the label list.

Case Analysis. In Fig. 4, we perform a detailed case analysis of our method using the prompt we mentioned in the main experiment on the Emotion6 dataset. The first two examples demonstrate the alignment between the text response, the predicted label, and the true label, which serves as an indicator of the potential of the LMM for image emotion prediction. Conversely, in the third example, the predicted label and the LMM's text response are consistent but diverge from the ground-truth label. However, the model's response seems rational since joy is a reasonable emotion people would feel when they see this image; it is just different from the labelers' view. In the fourth example, all three components: the predicted label, the LMM text response, and the ground truth label are different. While the predicted label differs from the ground truth label, the predicted label is closer to the ground truth label than it is to the text response, indicating the importance of our adapter network. Also, note that in this example, the text output accurately describes the image, but gives a different perspective on the emotions associated with it.

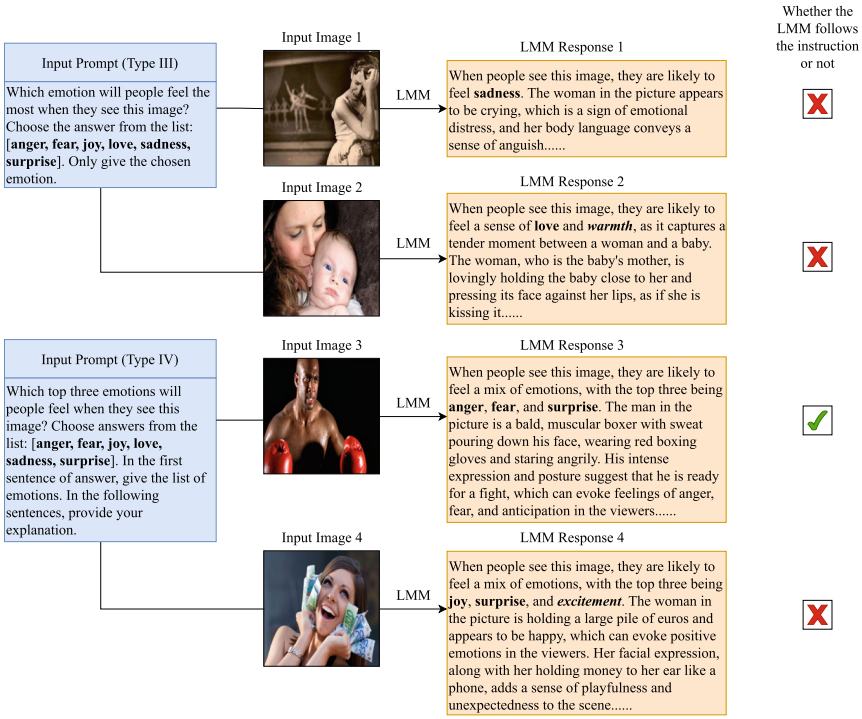


Fig. 3. Example analysis on Type III and Type IV prompts using UnBiasedEmotion dataset. From these examples, we can see that the LMM's ability to follow instructions can still be improved.

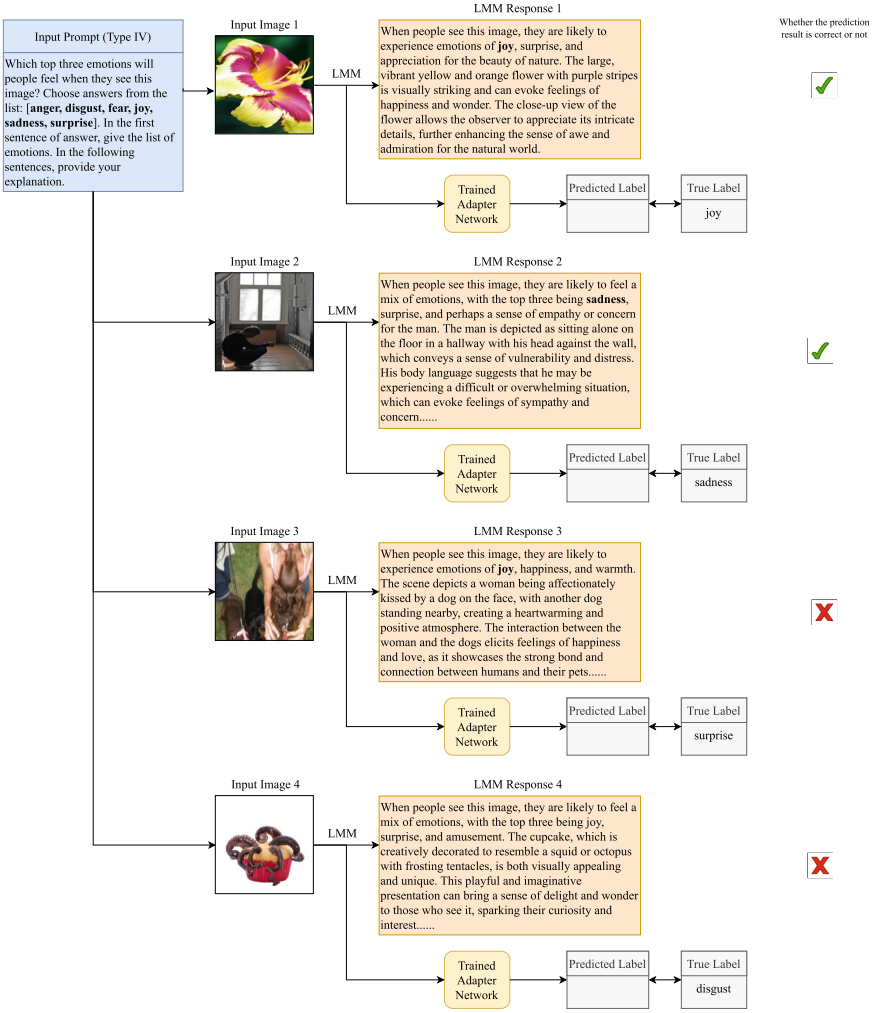


Fig. 4. Success and failure case analysis on Type IV prompts using Emotion6 dataset.

5 Conclusion

In summary, we present a novel approach for predicting emotions from images, leveraging the power of large multimodal models (LMMs). Our method combines a pre-trained LMM with a shallow adapter network to map the output of the LMM to the emotion label. Our experimental results serve as a compelling evidence for the robustness and efficacy of our method. In particular, our method is capable of surpassing the previous SOTA performance even when trained on only 20% of the data on three of the four studied datasets, which underscores

the potency of our method. Moreover, the text response of the LMM could serve as an explanation of the predicted label.

Acknowledgement. This research is based upon work supported by the Defense Advanced Research Projects Agency (DARPA)'s INCAS program, project HR001121S0008. The views and conclusions contained herein should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

1. Ali, A.R., Shahid, U., Ali, M., Ho, J.: High-level concepts for affective understanding of images. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 679–687. IEEE (2017)
2. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
3. Driess, D., et al.: Palm-e: an embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
4. Feng, T., Liu, J., Yang, J.: Probing sentiment-oriented pre-training inspired by human sentiment perception mechanism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2850–2860 (2023)
5. Hartmann, J.: Emotion English distilroberta-base (2022). https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/
6. Holbrook, M.B., O'Shaughnessy, J.: The role of emotion in advertising. Psychol. Mark. **1**(2), 45–64 (1984)
7. Hosseini, M., Caragea, C.: Feature normalization and cartography-based demonstrations for prompt-based fine-tuning on emotion-related tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 12881–12889 (2023)
8. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning, pp. 4904–4916. PMLR (2021)
9. Kang, H., Hazarika, D., Kim, D., Kim, J.: Zero-shot visual emotion recognition by exploiting bert. In: Proceedings of SAI Intelligent Systems Conference, pp. 485–494. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-16078-3_33
10. Kosti, R., Alvarez, J.M., Recasens, A., Lapedriza, A.: Context based emotion recognition using emotic dataset. IEEE Trans. Pattern Anal. Mach. Intell. **42**(11), 2755–2766 (2019)
11. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: Proceedings of the 40th International Conference on Machine Learning. ICML 2023. JMLR.org (2023)
12. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
13. Luo, Y., Zhong, X., Zeng, M., Xie, J., Wang, S., Liu, G.: Cglf-net: image emotion recognition network by combining global self-attention features and local multiscale features. IEEE Trans. Multimedia **26**, 1894–1908 (2023)
14. Machajdik, J., Hanbury, A.: Affective image classification using features inspired by psychology and art theory. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 83–92 (2010)

15. Mikels, J.A., Fredrickson, B.L., Larkin, G.R., Lindberg, C.M., Maglio, S.J., Reuter-Lorenz, P.A.: Emotional category data on images from the international affective picture system. Behav. Res. Methods **37**, 626–630 (2005)
16. OpenAI: Gpt-4 technical report. ArXiv arxiv:2303.08774 (2023)
17. Pan, J., Wang, S.: Progressive visual content understanding network for image emotion classification. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 6034–6044 (2023)
18. Panda, R., Zhang, J., Li, H., Lee, J.Y., Lu, X., Roy-Chowdhury, A.K.: Contemplating visual emotions: understanding and overcoming dataset bias. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 579–595 (2018)
19. Parrott, W.G.: Emotions in Social Psychology: Essential Readings. Psychology press, London (2001)
20. Patterson, G., Hays, J.: Sun attribute database: discovering, annotating, and recognizing scene attributes. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2751–2758. IEEE (2012)
21. Peng, K.C., Chen, T., Sadovnik, A., Gallagher, A.C.: A mixed bag of emotions: model, predict, and transfer emotion distributions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 860–868 (2015)
22. Peng, K.C., Sadovnik, A., Gallagher, A., Chen, T.: Where do emotions come from? predicting the emotion stimuli map. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 614–618. IEEE (2016)
23. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
24. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). https://arxiv.org/abs/1908.10084
25. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: masked and permuted pre-training for language understanding. Adv. Neural. Inf. Process. Syst. **33**, 16857–16867 (2020)
26. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
27. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
28. Truong, Q.T., Lauw, H.W.: Visual sentiment analysis for review images with item-oriented and user-oriented cnn. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 1274–1282 (2017)
29. Wang, M., Zhao, Y., Wang, Y., Xu, T., Sun, Y.: Image emotion multi-label classification based on multi-graph learning. Expert Syst. Appl., 120641 (2023)
30. Wang, X., Jia, J., Yin, J., Cai, L.: Interpretable aesthetic features for affective image classification. In: 2013 IEEE International Conference on Image Processing, pp. 3230–3234. IEEE (2013)
31. Wei, Z., et al.: Learning visual emotion representations from web data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13106–13115 (2020)
32. Yang, J., She, D., Lai, Y.K., Rosin, P.L., Yang, M.H.: Weakly supervised coupled networks for visual sentiment analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7584–7592 (2018)

33. You, Q., Luo, J., Jin, H., Yang, J.: Building a large scale dataset for image emotion recognition: The fine print and the benchmark. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
34. Zhang, H., Xu, M.: Weakly supervised emotion intensity prediction for recognition of emotions in images. IEEE Trans. Multimedia **23**, 2033–2044 (2020)
35. Zhang, J., Yang, D., Bao, S., Cao, L., Fan, S.: Emotion classification on code-mixed text messages via soft prompt tuning. In: Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pp. 596–600. Association for Computational Linguistics, Toronto (2023). https://aclanthology.org/2023.wassa-1.57
36. Zhang, P., et al.: Vinvl: revisiting visual representations in vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5579–5588 (2021)
37. Zhang, Y., Ding, W., Xu, R., Hu, X.: Visual emotion representation learning via emotion-aware pre-training. In: IJCAI, pp. 1679–1685 (2022)

Flatter Minima of Loss Landscapes Correspond with Strong Corruption Robustness

Liqun Zhong[1,2], Kaijie Zhu[1,2], and Ge Yang[1,2(✉)]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
{zhongliqun2019,ge.yang}@ia.ac.cn
[2] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Science, Beijing 100190, China

Abstract. To develop deep neural networks (DNNs) that maintain strong performance under different natural image corruptions such as noise and blur, namely with strong corruption robustness, is essential for real-world applications. Current methods to enhance corruption robustness focus primarily on data augmentation. But these methods are often empirically designed without a clear theoretical foundation. Here, we show that flatter minima of loss landscapes generally correspond with stronger corruption robustness. Based on this finding, we have developed a method named LASAM-DA that combines layer adaptive sharpness-aware minimization(LASAM) with data augmentation (DA) to boost corruption robustness. LASAM pushes DNNs towards flat minima of loss landscapes, while DA guides DNNs to memorize diverse data distributions. LASAM and DA reinforce each other without interference. Experiments on CIFRA-10-C, CIFAR-100-C and Tiny-ImageNet-C show that LASAM-DA outperforms competing methods and achieves state-of-the-art performance. LASAM-DA is versatile, not restricted to specific data types, network architectures or data augmentation methods. Code is openly accessible at https://github.com/cbmi-group/Flatter-Minima-Helps-for-Robustness.

Keywords: Corruption robustness · Flat minima · Layer adaptive sharpness-aware minimization · Data augmentation

1 Introduction

In real-world applications, images may be degraded by a wide variety of natural corruptions such as noise, blur, weather conditions, and digital distortions (see

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_20.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 314–328, 2025.
https://doi.org/10.1007/978-3-031-78107-0_20

Supplementary Fig. S1 for corrupted image examples). It is essential to ensure that deep neural networks (DNNs) maintain their performance against such corruptions. Corruption robustness characterizes the performance of DNNs under these corruptions [5]. Related to corruption robustness, adversarial robustness defines the worst-case performance of DNNs under targeted adversarial attacks [9]. While image corruptions occur naturally, adversarial attacks are designed in a tailored fashion. A considerable amount of work [18,27] has shown that adversarial robustness and corruption robustness are related but not equivalent. Several datasets such as ImageNet-C, CIFAR-10-C and CIFAR-100-C [11] have been developed to benchmark corruption robustness. However, because of the wide variety of image corruptions, it remains challenging to develop DNNs that are robust against all corruptions.

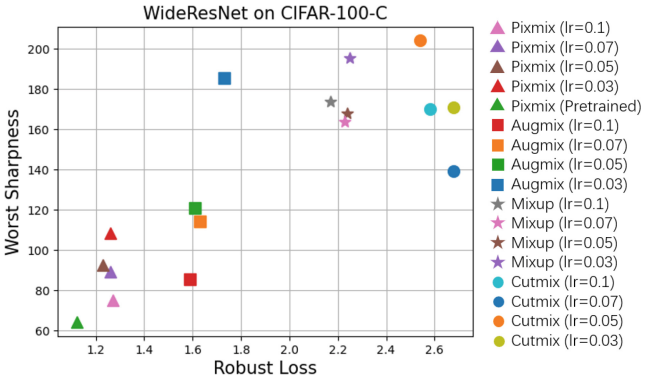


Fig. 1. Relation between corruption robustness (measured in cross-entropy) and loss landscape flatness (measured in worst sharpness). Training of WideResNet [33] with four data augmentation strategies and four initial learning rates. *RLoss* (lower corresponds to stronger robustness, x-axis) is plotted with respect to worst sharpness (lower corresponds to flatter minima, y-axis). There is a clear positive correlation between corruption robustness and loss landscape flatness (with a Pearson correlation of 0.82, p-value <0.001). Namely, flatter minima correspond with stronger robustness.

Specifically, for image classification, a good classifier should generalize well from training data to images with diverse corruptions. However, studies [8,11] have shown that vanilla DNNs can fail to handle such corruptions, resulting in performance collapse. Therefore, it is essential to enhance corruption robustness of DNNs. Currently, data augmentation is the most commonly used strategy for this purpose. It is hoped that when DNNs are trained with images of diverse conditions, they should become robust against shifts in data distribution caused by image corruptions. To this end, some studies choose to synthesize large numbers of corrupted images for training while others choose to increase the complexity or entropy of training data. However, these methods are empirically designed and lack a clear theoretical foundation. They are often limited in their performance or in the types of corruptions they can handle. In view of the wide variety

of corruptions, it is often difficult to determine *a prior* which types of augmentation operations are the most effective. Furthermore, increasing the diversity of training data comes at the expense of increased computational cost.

Here, we study corruption robustness from a different perspective. We quantify the corruption robustness of DNNs using corruption robustness loss (referred to as *RLoss*) and quantify the flatness of loss landscapes using worst sharpness [17,29]. Then we examine the relation between the corruption robustness and loss landscape flatness on several classification models trained with several representative data augmentation methods, we find that flatter minima generally correspond with smaller corruption *RLoss* hence stronger corruption robustness (Fig. 1). To further verify this finding, we have conducted additional experiments. We then propose a new strategy to enhance corruption robustness. Our main research contributions are as follows:

(1) Through experiments, we find that in image classification, flatness of local minima in loss landscape corresponds positively with corruption robustness. Models that converge to flatter minima generally show stronger corruption robustness.
(2) Based a simple linear binary classification, we demonstrate how flat minima helps for corruption robustness.
(3) We propose LASAM-DA to enhance corruption robustness, in which LASAM pushes DNNs towards flat minima of loss landscapes, while DA guides DNNs to memorize diverse data distributions. Experiments on CIFAR-10-C and CIFAR-100-C show that LASAM-DA outperforms competing state-of-art methods. Furthermore, LASAM-DA is broadly applicable, not restricted to specific data types, network architectures, and data augmentation approaches.

2 Related Work

2.1 Corruption Robustness

Multiple studies have shown that the performance of DNNs trained on clean or undistorted images can drop substantially under image corruptions [8,11]. Hendrycks et al. [11] simulates 18 types of corruptions with 5 levels of severity for each type. The simulated datasets, including CIFAR-10-C, CIFAR-100-C and ImageNet-C, can be used to benchmark corruption robustness of DNNs. Substantial research effort has been made to make DNNs more robust against corruptions simulated in these datasets. Currently, data augmentation is the most commonly used strategy. It is hoped that when DNNs are trained with more diverse data distributions, they will be more robust against diverse corruptions. Other than commonly used data augmentation such as cropping, random flipping, and rotation, several new methods have been proposed. For example, Cutout [3] randomly picks a patch in an image and fills it with zeros, keeping labels unchanged. Patch Gaussian [22] adds a patch of Gaussian noise to an image. Cutmix [32] randomly fills an image with a patch of other training

images. Mixup [36] randomly selects two training images and their labels, and uses linear interpolation to generate a new training image and a new training label.

Overall, these operations approaches are rather simplified and, unsurprisingly, less than satisfactory in performance. To address this problem, some more sophisticated methods have been proposed. Augmix [12] consists of three augmentation chains. In each augmentation chain, one to three augmentation operations are randomly selected. When feeding an image into the augmentation chains, outputs of these chains are weighted to obtain the final augmented image. Based on Augmix, Augmax [30] adjusts the weights of augmentation operations to get the hardest augmented samples. AutoAugment [2] employs a search method such as reinforcement learning in the space of augmentation operations to automatically find better data augmentation strategies. In addition, Pixmix [13] adds fractals and visualized features to increase the structural complexity of training data. Prime [24] provides a data augmentation method that relies on the simple yet rich families of max-entropy image transformations.

Different from these manually crafted image augmentation methods, learnable augmentation methods are proposed to synthesize diverse training data. For example, ANT [27] generates a series of adversarial noise samples through a generative network to fool a classifier. Augmented images are generated by adding these adversarial noise samples onto clean images. DeepAugoment [10] utilizes an image-to-image transformation network to generates a series of perturbations onto clean input images, thus increasing diversity of training data. Overall, these methods focus on augmentation of training data, pushing DNNs to see more diverse data distributions to achieve robustness. However, they are empirically designed without a clear theoretical foundation. Consequently, augmentation methods that combine augmentation operations cannot explain which property of augmentation is more effective. Studies in [23] have proposed Minimal Sample Distance between augmentations and corruptions to demonstrate a strong correlation between similarity and robustness, but with limited success.

2.2 Loss Landscape Flatness

The loss landscape is a concept widely used in machine learning for understanding the convergence behavior of optimization algorithms and analyzing the interplay between model architecture, data complexity, and optimization procedures. Each point on the landscape represents the loss that corresponds to a certain parametric configuration of a DNN.

Analysis of flat and sharp minima of loss landscape dates back at least to [14]. Recently, the notion of flat minima has received much attention in understanding the behavior of DNNs. Empirical evidence [16,17,20] suggests positive correlation between sharpness and generalization. Some studies on adversarial training and loss landscape [21,25,29] find adversarial training does not necessarily result in flatter loss landscape. Wu *et al.* [31] first explored the loss landscape with regard to adversarial examples and identified a strong connection between loss flatness and robust generalization gap. Stutz *et al.* [29] defined the flatness in

weight space and showed improved loss flatness contributes to adversarial robustness. But the relation between loss landscape flatness and corruption robustness is unclear.

Some studies focus on improving loss landscape flatness. Small batch size training [17] and usage of unlabeled data [34] are recommended, because they have been shown to help convergence towards flatter minima implicitly. Adversarial weight perturbation (AWP) [31] is proposed to explicitly regularize the flatness of loss landscape to obtain flatter local minima. Sharpness-aware minimization (SAM) [7] explicitly optimizes both loss function and loss sharpness. Adaptive SAM (ASAM) [19] is an improvement over SAM, in which a new concept of adaptive sharpness is introduced. ASAM is designed based on adaptive sharpness to alleviate the side-effect of the original SAM in training caused by scale-dependency.

3 Method

3.1 Flat Minima Helps for Corruption Robustness

For ease of understanding, we use a simple linear binary classification as an example to visually explain how a flat loss landscape contributes to stronger corruption robustness.

Given a dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N,\ y_N)\}$. Where $x_i \in R^n$ denotes input sample, $y_i \in [+1, -1]$ is category label of $x_i (i = 1, 3, \ldots, N)$. Suppose there are two separating hyperplanes, denoted as $(\omega_1, b_1), (\omega_2, b_2)$ respectively, where (ω_2, b_2) is the maximum margin separating hyperplane obtained by support vectors (has the maximum geometric margin from all sample points). While (ω_1, b_1) is close to some sample points. As shown in Fig. 2(a), the two separating hyperplanes can correctly classify all samples. For all positive samples, there are $\omega_1 \cdot x + b_1 = 1$, $\omega_2 \cdot x + b_2 = 1$, for all negative samples, there are $\omega_1 \cdot x + b_1 = -1$, $\omega_2 \cdot x + b_2 = -1$. Next, we will compare the flatness of (ω_1, b_1) and (ω_2, b_2). Before that, we see the definition of loss function of linear binary classification:

$$Loss\,(w, b) = -\sum_{i}^{N} y_i\,(\omega \cdot x_i + b) \tag{1}$$

When the classifier works correctly, the model outputs the same result as category label, that is, $\omega \cdot x + b = 1, y = +1$ or $\omega \cdot x + b = -1, y = -1$. In both cases, $-y\,(\omega \cdot x + b) = -1$, then $Loss\,(w, b) = -\sum_{i}^{N} y_i\,(\omega \cdot x_i + b) = -N$ reaches the minima. If there are M wrong classifications, for misclassified sample, $-y\,(\omega \cdot x + b) = 1$, then $Loss(\omega, b) = -N + M$.

Obviously, both $Loss(\omega_1, b_1)$ and $Loss(\omega_2, b_2)$ are $-N$. Because separating hyperplane (ω_2, b_2) stays farthest away from all samples and (ω_1, b_1) is close to some samples. So, there must exist a weight perturbation $(\Delta\omega, \Delta b)$. Under this perturbation, a new decision hyperplane $(\omega_2 + \Delta\omega, b_2 + \Delta b)$ is

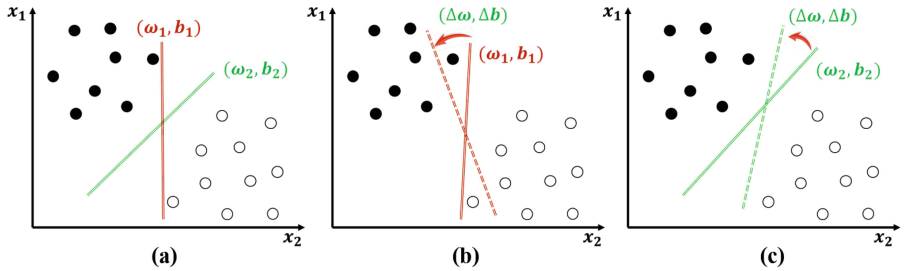


Fig. 2. An example of linear binary classification.

obtained. At this hyperplane, all samples can still be correctly classified, so $Loss(\omega_2 + \Delta\omega, b_2 + \Delta b) = -N$ still reaches the minima. However, for separating hyperplane $(\omega_1 + \Delta\omega, b_1 + \Delta b)$, some samples are misclassified due to angle offset, so $Loss(\omega_1 + \Delta\omega, b_1 + \Delta b) = -N + M$, where $M > 0$, indicating the number of misclassified samples. see Fig. 2(b, c). Under the same perturbation, we calculate numerical change of loss function:

$$Loss\left(\omega_1 + \Delta\omega, b_1 + \Delta b\right) - Loss\left(\omega_1, b_1\right) = M > 0 \tag{2}$$

$$Loss\left(\omega_2 + \Delta\omega, b_2 + \Delta b\right) - Loss\left(\omega_2, b_2\right) = 0 \tag{3}$$

As can be seen, loss change of (ω_2, b_2) is still 0, while loss change of (ω_1, b_1) is greater than 0. According to the intuitive understanding of loss flatness, within the same neighborhood, loss change of (ω_2, b_2) is smaller than (ω_1, b_1), thus (ω_2, b_2) is flatter than (ω_1, b_1).

We conclude that a flatter loss landscape corresponds to a larger geometric margin between the decision hyperplane and sample points. In this case, when input data distribution undergoes some deviations, the model has enough fault-tolerant capability to perform well. Conversely, if a model converges to a sharp minima, even small sample or weight perturbations can cause the loss increases sharply, resulting in decreased model performance. Similarly, for convolutional neural networks, flatter minima means less change in loss. That is, models will be more powerful to handle data offset while keeping its prediction error relatively stable. Therefore, models that converge to flatter minima have greater fault-tolerant capability and greater corruption robustness, while models that converge to sharper minima are less robust to data corruption.

3.2 Worst Sharpness

A variety of metrics have been proposed to quantify loss landscape flatness. In this study, we use worst sharpness [29], which is determined by the maximum and minima loss within a fixed weight perturbation v. The maximum loss is thought to represent the highest point within perturbation v. Thus lower worst sharpness indicates a flatter minima. Suppose the network eventually converges to weight w, the worst sharpness is computed as:

$$\max_{v \in B_\xi(w)} \left[loss\left(x; w+v\right) - loss\left(x; w\right) \right] \tag{4}$$

where $B_\xi\left(w\right)$ denotes the weight perturbation and x denotes input. ξ is weight perturbation coefficient. To ensure the scale-invariance [4] of weights, $B_\xi\left(w\right)$ is set to be proportional to weight norm.

$$B_\xi\left(w\right) = \left\{ w+v : ||v^{(l)}||_2 \leq \xi ||w^{(l)}||_2, \forall layer\ l \right\} \tag{5}$$

In our experiment, we search for local maximum loss $\max_{v \in B_\xi(w)} [loss(x; w+v)]$ alongside the direction of gradient ascent.

3.3 LASAM

Sharpness-aware minimization (SAM) [7] is a common training method to improve loss flatness, which is designed to optimize both loss function and loss flatness explicitly. Training with SAM is essentially solving the following sharpness aware-minimization (SAM) problem:

$$\begin{aligned} &\min_w \left(loss^{SAM}(w) + \lambda ||w||_2^2 \right) \\ &loss^{SAM}(w) \stackrel{\Delta}{=} \max_{||v|| \leq B_\xi(w)} \left[loss(x; w+v) \right] \end{aligned} \tag{6}$$

where w is convergence point. As is shown in Eq. (6), SAM contains two optimizations, the first one is to find a sharp minima, that is, to maximize the loss

Algorithm 1: LASAM algorithm

Input: Loss function $loss(w)$, training dataset $S : \cup_{i=1}^{n} \{(x_i,\ y_i)\}$, batch size b, total number of layers L of the model, model weight $w = [w^0, w^1 ... w^{L-1}]$, initial maximum weight perturbation coefficient $\xi_0 : ||v^{(0)}||_2 \leq \xi_0 ||w^{(0)}||_2$ for the first layer, weight decay coefficient λ, scheduled learning rate α, initial weight w_0.

Output: Trained weight w.

Initialize weight $w = w_0$.
while *not converged* **do**
  Sample a mini-batch B of size b from S, $B = \{(x_1, y_1), (x_2, y_2), \ldots (x_b, y_b)\}$;
  Compute $loss_B(w)$ and gradient $\nabla loss_B(w)$;
  T_w is element-wise normalization, $T_w = diag\left(\left|w^0\right|,\ \left|w^1\right|, \ldots \left|w^{L-1}\right|\right)$;
  for $l = 0, 1, .., L-1$ **do**
    $\xi_l = \xi_0 - l/L$ for layer l;
    $v^l = \xi_l T_w^2 \nabla loss_B(w) / \left\| T_w \nabla loss_B(w) \right\|_2$;
    $w^l = w^l - \alpha \left(\nabla loss_B\left(w^l + v^l\right) + \lambda w^l \right)$;
  end
end
return $w = [w^0, w^1 ... w^{L-1}]$

gaps within weight perturbations. When a model reach the sharp region, the second optimization helps to jump out of the sharp minima.

Adaptive SAM (ASAM) [19] is an improved SAM, in which a new concept called adaptive sharpness is introduced. ASAM is designed to alleviate the side-effect of the original SAM in training caused by scale-dependency.

In this study, we propose an improved ASAM called layer adaptive sharpness-aware minimization (referred to as LASAM). It is built on recent studies [1,35,37] which revealed that different layers within DNNs display distinct attributes in terms of generalization and robustness. In ASAM, the weight perturbation coefficient ξ (see Eq. (5)) for each layer are fixed. However, [37] reports that weights close to the input are more stable and can accept larger perturbations. As layer deepens, weights gradually become 'sensitive'. Sometimes, even a small perturbation may cause the model to become unstable and unsuitable for large-scale perturbations. So, in our experiments, ξ is set to be the highest in the first layer and to be the lowest in the last layer. Then in the intermediate layers, ξ values are determined through linear interpolation, gradually dropping from the highest value to the lowest value. See Algorithm 1 for detalied LASAM operations.

3.4 LASAM-DA

Further, training with LASAM only operates in the weight space, independent of hyperparameters, inputs and outputs. Since much work [12,13,22,24,32,36] has been done to enhance corruption robustness by diversifying data distribution in the input space, it is reasonable to believe that it may be beneficial to optimize both in the weight space and the input space.

We propose a new strategy (referred to as LASAM-DA) that combines LASAM with data augmentation in model training. Specifically, LASAM operates in the weight space to push the trained model towards a flat minima, while data augmentation will guide DNNs to memorize diverse data distributions from image corruptions. Training with LASAM-DA achieves the strongest corruption robustness to date.

4 Experiments

We use primarily two benchmark datasets CIFAR-10-C, CIFAR-100-C and Tiny-ImageNet-C [11] to characterize corruption robustness. CIFAR-10-C contains 10,000 test images in 10 categories while CIFAR-100-C contains 10,000 test images in 100 categories. Tiny-ImageNet-C is a subset of ImageNet and contains 200 categories, where each category contains 500 color images with size 64×64. In these three datasets, 19 types of corruption degradation are simulated on each image, with 5 levels of severity for each corruption type. We conduct experiments on WideResNet [33], tiny ViT (ViT-T) [6], DenseNet [15] and VGG19 [28]. Each of them represents an important yet different class of network architecture. For

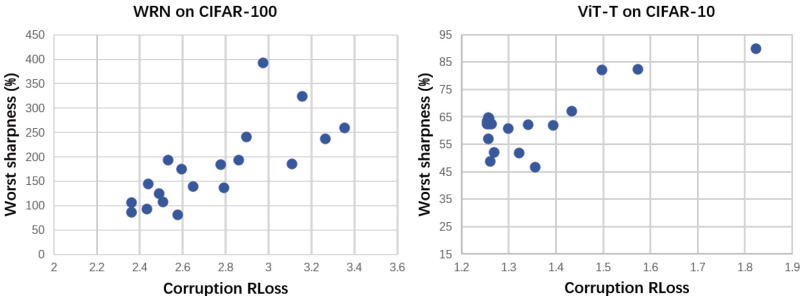


Fig. 3. Corruption robust loss (*RLoss*) with respect to worst sharpness throughout one training, each point corresponds to an epoch. WRN: WideResNet. Networks are trained on CIFAR-10, CIFAR-100 and evaluated on CIFAR-10-C, CIFAR-100-C.

data augmentation, we choose four representative methods: Mixup [36], Cutmix [32], Augmix [12] and Pixmix [13].

Suppose $x \in R^{N \times N}$ is an image to be tested, with label is y, $f_\theta(\cdot)$ is model. Then *RLoss* and *Test Error* can be defined as:

$$RLoss = CrossEntropy\,(f(x); y) \tag{7}$$

$$Test\ Error = N_{misclass}/N_{total} \tag{8}$$

In particular, when x is corrupted image, *Test Error* is equal to robust error (*RErr*). Where $N_{misclass}$ is the number of misclassification on test datasets, N_{total} is the number of total images being tested.

4.1 Relation Between Corruption Robustness and Worst Sharpness

When WideResNet [33] is trained on CIFAR-100 and ViT-T [6] on CIFAR-10, we recorded train loss, test loss and corruption loss curves over epochs, (Supplementary Fig. S2). Unlike robust overfitting commonly observed in adversarial robustness [26], corruption loss, characterized by cross-entropy, gradually decrease during training until stabilizing at a steady band. When we computed worst sharpness and corruption robust loss over epochs, we observed positive relationship between sharpness and corruption loss (see Fig. 3).

To further verify this observation, we went on to test four data augmentation methods (Mixup [36], Cutmix [32], Augmix [12] and Pixmix [13]). Considering that the training is sensitive to initial learning rate, we try four initial learning rates for each augmentation training. Again, we had the same observation, as shown in Fig. 1. Taken together, these results indicate that networks with good robustness mostly converge to flat minima.

4.2 Improving Loss Minima Sharpness

Table 1. Results of two batch sizes for training of WideResNet on CIFAR-100, tested on CIFAR-100-C. *RLoss*: Robust loss, cross entropy on corrupted images. *RErr*: Robust error, misclassification rate on corrupted images. For *RErr* and *RLoss*, lower is better. Lower sharpness means flatter minima.

	Sharpness(%)↓		*RErr*(%)↓		*RLoss*↓	
Batch Size	128	1024	128	1024	128	1024
Cutmix	352.3	358.8	50.25	52.92	2.43	2.57
Mixup	212.7	349.7	45.59	48.77	2.03	2.18
Augmix	85.3	330.2	35.18	47.22	1.59	2.18
Pixmix	75.2	183.0	32.01	33.18	1.27	1.33

Table 2. Corruption test errors (lower is better) of some networks trained on CIFAR-10, CIFRA-100 and Tiny-ImageNet with LASAM. Test errors are evaluated on CIFAR-10-C, CIFAR-100-C and Tiny-ImageNet-C. Vanilla: Standard train. TIN: Tiny-ImageNet.

Data	Train	VGG19	WideResNet	DenseNet	ViT-T
CIFAR-10	Vanilla	26.38	22.84	23.38	27.33
	LASAM	22.97	21.02	21.41	24.79
CIFAR-100	Vanilla	54.50	49.35	46.74	56.58
	LASAM	51.49	45.83	43.82	55.53
TIN	Vanilla	79.51	74.75	68.53	78.28
	LASAM	78.55	72.32	66.83	76.36

In this section, we will try small batch size and LASAM to improve loss flatness and observe changes of corruption robustness. Consistent with findings in [17] and [19], small batch size and LASAM are helpful to cause ejection of a DNN from an sharp basin during training, enabling subsequent convergence to a flatter minima. To test our hypothesis that flatter minima in loss landscapes are associated with stronger corruption robustness, we experiment with two batch sizes: a small batch size of 128 and a large batch size of 1024.

We train WideResNet on CIFAR-100 with a drop rate of 0.3 for 200 epochs. Each training is performed with different type of data augmentation from Pixmix, Cutmix, Mixup and Augmix. Initial learning rate is 0.1, then decreasing the learning rate by 10 at 50% training and 75% training. Optimizer is SGDM. Corruption loss and errors are both evaluated on CIFAR-100-C and recorded

Table 3. Test error (lower is better) of WideResNet on CIFAR-100 (corresponds to *'Clean'*) and CIFAR-100-C (corresponds to *'Corrup'*). WideResNet is trained on CIFRA-100, with vanilla (without LASAM or data augmentation), LASAM (only LASAM), DA (only data augmentation) and Ours (LASAM-DA, both LASAM and data augmentation) respectively. Bold: best in each row.

	Test data	Vanilla	LASAM	DA	**Ours**
Cutmix	Clean	21.21	18.97	18.68	**16.71**
	Corrup	49.35	44.52	48.85	**43.73**
Mixup	Clean	21.21	18.97	19.61	**18.90**
	Corrup	49.35	44.52	45.58	**42.95**
Augmix	Clean	21.21	18.97	20.17	**18.91**
	Corrup	49.35	44.52	35.14	**31.96**
Pixmix	Clean	21.21	18.97	20.23	**18.33**
	Corrup	49.35	44.52	32.25	**29.74**

together with worst sharpness in Table 1 (Results of ViT-T trained on CIFAR-10 is presented in Supplementary Tab. S1).

It is clear in Table 1, small batch size of 128 does contribute to better flatness [17]. When using small batch size, we see smaller robust loss and smaller robust errors. Besides batch size, we also achieve general improvements in corruption robustness using LASAM (in Table 2). Taken together, experiments indicate that networks converging to flatter minima will achieve better corruption robustness.

4.3 Ablation Studies

Several sets of ablation experiments have been conducted. The results are summarized in Table 3. Four data augmentation are examined. We train WideResNet on CIFAR-100 without any corruption. Then the model is tested on CIFAR-100 valid data to characterize in-distribution generalization and CIFAR-100-C valid data to characterize corruption robustness. Results of more DNNs and training data (CIFAR-100-C and Tiny-ImageNet-C) are presented in the Supplementary Tab. S2-5.

From Table 3, we can observe: (i). LASAM-DA always achieves the lowest test error no matter which data augmentation method is used. (ii). Comparing column 4 to 6, LASAM-DA is better than either pure data augmentation or LASAM, indicating that simultaneous optimization of input and weight reinforces each other without interference. (iii). Better in-distribution generalization error does not always mean lower corruption error. For example, for Cutmix, the generalization error decreases from 21.21% to 18.68%, but corruption error remains almost unchanged. Therefore, better in-distribution generalization does not necessarily mean better corruption robustness.

For further verification, we plot scatter plots of generalization errors and corruption errors. As can be seen from the Fig. 4(a), there is no obvious corre-

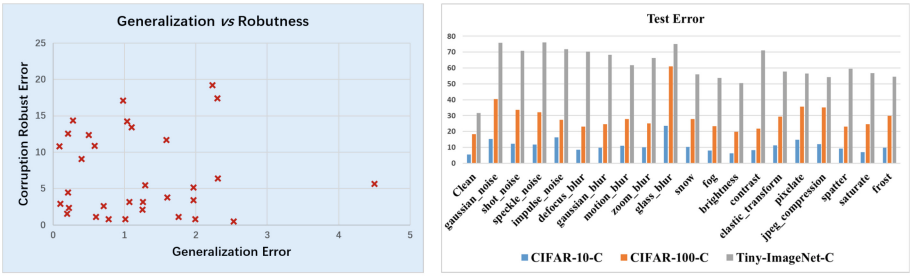


Fig. 4. (a): Comparison of in-distribution generalization and corruption robust errors. (b) Test errors on each type of corruption of LASAM-DA.

lation between the two errors. Additionally, in-distribution generalization errors range from 0–5%, while corruption errors range from 0–20%. It means that the gain of corruption robustness is much greater than the gain of generalization performance. Taken together, although LASAM aims to improve generalization, it essentially improves loss flatness. Based on our observation, LASAM contributes to corruption robustness mainly because of improved flatness rather than improvements of in-distribution generalization.

4.4 LASAM-DA

Based on the results of above ablation experiments, our method, LASAM-DA, performs best when Pixmix is selected as data augmentation. We will evaluate the performance of existing advanced methods for enhancing corruption robustness on CIFAR-10-C and CIFAR-100-C benchmark datasets. These methods include Patch Gaussian [22], Augmax [30], AutoAugment [2], Prime [24], Pixmix [13], and ANT [27]. Most of these methods have been reproduced using shared code. It should be noted that our reproduced results may be worse than those reported in the original papers due to different model selected and without any parameter adjustment. To ensure fairness, all methods share the same hyperparameters. Table 4 shows the final result, which is averaged across all corruption types (about 9000,000 images). Detailed test errors on each type of corruption on CIFAR-10-C, CIFAR-100-C, Tiny-ImageNet-C of LASAM-DA are presented Fig. 4(b), more resluts are presented in Supplementary Fig. S3.

While LASAM-DA is not the best-performing method on clean test images of CIFAR-10 (with only 0.5% difference from the best-performing method), it achieves the lowest test error in all other cases. In particular, LASAM-DA outperforms the best method by 1–2% on corrupted images. Overall, our proposed method, LASAM-DA, performs the best corruption robustness, indicating its superiority over other methods.

Table 4. Test errors on CIFAR-10-C, CIFAR-100-C and Tiny-ImageNet-C. Model is trained on clean images, then tested on CIFAR-10, CIFAR-100, Tiny-ImageNet (corresponds to *'Clean'*) and CIFAR-10-C, CIFAR-100-C, Tiny-ImageNet-C (corresponds to *'Corrup'*). Vanilla (without any data augmentation). AutoAug: AutoAugment. Ours: LASAM-DA. Bold: best, Underline: second best.

	CIFAR-10-C		CIFAR-100-C		Tiny-ImageNet-C	
Test data	Clean	Corrup	Clean	Corrup	Clean	Corrup
Vanilla	6.61	24.72	21.21	49.35	35.04	74.75
PG	7.23	19.95	21.67	40.14	35.89	71.53
Augmix	6.03	13.85	20.17	35.14	33.85	65.64
AutoAug	<u>5.31</u>	19.65	<u>19.51</u>	39.74	**33.56**	73.80
Prime	**5.03**	12.65	19.97	<u>31.84</u>	34.99	65.34
Augmax	6.62	12.19	20.40	33.35	34.07	63.94
ANT	7.03	<u>12.15</u>	22.05	33.54	34.73	64.20
Pixmix	6.61	12.38	20.23	32.25	34.49	<u>63.34</u>
Ours	5.51	**11.29**	**18.33**	**29.74**	<u>33.61</u>	**61.62**

5 Conclusion

In this study, we find that flatness of loss landscape correlates positively with corruption robustness of DNNs. Then we explain the phenomenon through a linear binary classification model. Specifically, models that converge to flatter minima will obtain greater fault-tolerant capability. As a result, these models are more powerful in handling data offset while keeping their prediction error relatively stable. Based on this finding, we propose LASAM to boost loss flatness. Further, we have developed a method that combines LASAM and data augmentation (DA) to enhance corruption robustness. Importantly, LASAM and DA reinforce each other without interference. LASAM-DA outperforms competing methods and achieves state-of-the-art corruption robustness. It is versatile and can be combined with different data types, model architectures and data augmentation methods. Overall, our study provides a new perspective and a clear theoretical foundation for enhancing corruption robustness of DNNs.

Acknowledgements. The authors thank colleagues in the Laboratory of Computational Biology and Machine Intelligence for their technical assistance. This work was supported in part by the National Natural Science Foundation of China (grants 92354307, 91954201, 31971289, 32101216), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant XDB37040402) and the Fundamental Research Funds for the Central Universities (grant E3E45201X2).

References

1. Chatterji, N.S., Neyshabur, B., Sedghi, H.: The intriguing role of module criticality in the generalization of deep networks. In: ICLR (2020)
2. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)
3. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
4. Dinh, L., Pascanu, R., Bengio, S., Bengio, Y.: Sharp minima can generalize for deep nets. In: ICML, pp. 1019–1028. PMLR (2017)
5. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2016)
6. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
7. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. In: ICLR (2020)
8. Geirhos, R., Temme, C.R., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. NeurIPS **31** (2018)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
10. Hendrycks, D., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: ICCV, pp. 8340–8349 (2021)
11. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)
12. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: ICLR (2019)
13. Hendrycks, D., et al.: Pixmix: Dreamlike pictures comprehensively improve safety measures. In: CVPR, pp. 16783–16792 (2022)
14. Hochreiter, S., Schmidhuber, J.: Flat minima. Neural Comput. **9**(1), 1–42 (1997)
15. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR, pp. 4700–4708 (2017)
16. Kaddour, J., Liu, L., Silva, R., Kusner, M.J.: When do flat minima optimizers work? NeurIPS **35**, 16577–16595 (2022)
17. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. In: ICLR (2016)
18. Kireev, K., Andriushchenko, M., Flammarion, N.: On the effectiveness of adversarial training against common corruptions. In: Uncertainty in Artificial Intelligence, pp. 1012–1021. PMLR (2022)
19. Kwon, J., Kim, J., Park, H., Choi, I.K.: Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In: ICML, pp. 5905–5914. PMLR (2021)
20. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. NeurIPS **31** (2018)
21. Liu, C., Salzmann, M., Lin, T., Tomioka, R., Süsstrunk, S.: On the loss landscape of adversarial training: identifying challenges and how to overcome them. NeurIPS **33**, 21476–21487 (2020)

22. Lopes, R.G., Yin, D., Poole, B., Gilmer, J., Cubuk, E.D.: Improving robustness without sacrificing accuracy with patch gaussian augmentation. arXiv preprint arXiv:1906.02611 (2019)
23. Mintun, E., Kirillov, A., Xie, S.: On interaction between augmentations and corruptions in natural corruption robustness. NeurIPS **34**, 3571–3583 (2021)
24. Modas, A., Rade, R., Ortiz-Jiménez, G., Moosavi-Dezfooli, S.M., Frossard, P.: Prime: a few primitives can boost robustness to common corruptions. In: ECCV, pp. 623–640. Springer (2022)
25. Prabhu, V.U., Yap, D.A., Xu, J., Whaley, J.: Understanding adversarial robustness through loss landscape geometries. arXiv preprint arXiv:1907.09061 (2019)
26. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: ICML, pp. 8093–8104. PMLR (2020)
27. Rusak, E., Schott, L., Zimmermann, R.S., Bitterwolf, J., Bringmann, O., Bethge, M., Brendel, W.: A simple way to make neural networks robust against diverse image corruptions. In: European Conference on Computer Vision (ECCV), pp. 53–69. Springer (2020)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Stutz, D., Hein, M., Schiele, B.: Relating adversarially robust generalization to flat minima. In: ICCV, pp. 7807–7817 (2021),
30. Wang, H., Xiao, C., Kossaifi, J., Yu, Z., Anandkumar, A., Wang, Z.: Augmax: adversarial composition of random augmentations for robust training. NeurIPS **34**, 237–250 (2021)
31. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems (NeruIPS), pp. 2958–2969 (2020)
32. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV, pp. 6023–6032 (2019)
33. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: BMVC (2016)
34. Zhai, R., et al.: Adversarially robust generalization just requires more unlabeled data. arXiv preprint arXiv:1906.00555 (2019)
35. Zhang, C., Bengio, S., Singer, Y.: Are all layers created equal? J. Machi. Learn. Res. **23**(1), 2930–2957 (2022)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. In: ICLR (2018)
37. Zhu, K., Hu, X., Wang, J., Xie, X., Yang, G.: Improving generalization of adversarial training via robust critical fine-tuning. In: ICCV, pp. 4424–4434 (October 2023)

Restoring Noisy Images Using Dual-Tail Encoder-Decoder Signal Separation Network

Akshay Agarwal[1(✉)], Mayank Vatsa[2], Richa Singh[2], and Nalini Ratha[3]

[1] IISER Bhopal, Bhopal, India
akagarwal@iiserb.ac.in
[2] IIT Jodhpur, Jodhpur, India
{mvatsa,richa}@iitj.ac.in
[3] University at Buffalo, Getzville, USA
nratha@buffalo.edu

Abstract. Obtaining paired noisy-clean images for various types of corruption is challenging; however, a noisy image can be viewed as the superposition of two distinct signals. Drawing inspiration from this concept, we address the problem of image purification by focusing on separating these signals to recover accurate classifier decisions. We introduce a dual-tail convolutional autoencoder designed to isolate the noise signal from the clean image. This architecture is engineered to simultaneously generate the additive noise pattern and the original clean signal. We conducted extensive experiments across various types of natural image noise with differing severity levels under both seen and unseen conditions. The results demonstrate that the proposed unique architecture effectively manages multiple noise types and significantly improves object recognition performance, which is severely impacted by image corruption. For example, Salt & Pepper noise reduces ResNet's accuracy on CIFAR10 from 91.81% to 20.48%, however, the dual-tail signal separator restores it to 91.61%. Additionally, the proposed method outperforms state-of-the-art approaches, uncovers connections between different corruptions, and, being cost-effective, has the potential to enable safe and secure AI deployment on low-cost devices.

Keywords: Natural Noises · Noise Remover · Dual Tail Architecture · Signal Separation · Robustness

1 Introduction

The susceptibility of image classifiers to common corruptions, such as noise and blur, has become a significant concern in the field [4,7–9,22]. Image enhancement, therefore, plays a crucial role in executing vision tasks effectively [30]. The quality of enhanced (or purified) images is critical and can significantly influence the performance of various vision algorithms, whether in super-resolution [25], object recognition [2], or object detection [37]. Moreover, images captured in unconstrained environments or through imperfect camera systems often contain noise [10,19], making image purification an essential preprocessing step. Several approaches have been proposed in the literature,

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 329–345, 2025.
https://doi.org/10.1007/978-3-031-78107-0_21

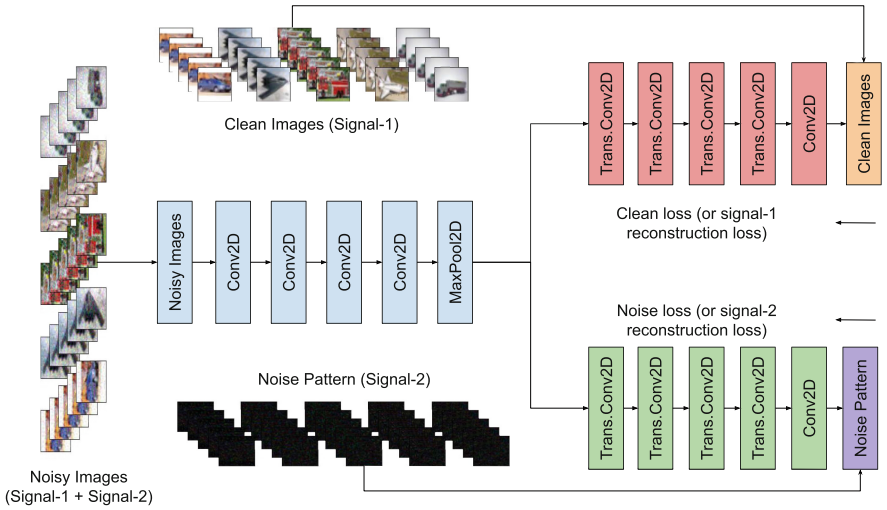


Fig. 1. Proposed dual tail encoder-decoder architecture *(DuTaNet)* for noise signal separation to generate the clean images. Trans.Conv2D represents the transpose convolution that aims to increase the resolution of the feature maps.

leveraging deep neural networks (DNNs) to map noisy images to their clean counterparts [16,43,45]. However, supervised learning methods [11,16,35,46], which rely heavily on large pairs of clean and noisy images, often struggle to generalize to unseen noise distributions. Traditional denoising methods [13,18] also fall short when dealing with unknown noise types, primarily because they tend to suppress noise rather than effectively remove it. In this research, we introduce a novel perspective, asserting that an image and its noise can be viewed as two distinct signals. By separating these signals, it is possible to restore the clean image by effectively removing the noise. We propose a dual-tail convolutional encoder-decoder network designed to separate the noise from an image. One tail focuses on extracting the clean image from noisy data, while the other isolates the added or inherited noise. We conduct extensive experiments across multiple datasets using deep convolutional networks to validate the effectiveness of the proposed Dual Tail (DuTa) signal separator network (*DuTaNet*). The key contributions of this research are:

- The development of a dual-tail image purification algorithm capable of removing noise with varying severity levels.
- Benchmarking different natural noises on multiple convolutional neural networks using CIFAR and F-MNIST datasets to advance research in this area.
- Demonstrating through extensive experiments that the proposed algorithm is robust and unbiased toward specific datasets or corruption types.

1.1 Related Work

In the literature, several image restoration networks [20,38] have been proposed to decompose the images into multiple image parts including raindrops and shadows. Han et al. [20] proposed the blind image decomposition network for image decomposition. Wang et al. [38] proposed a context-aware pre-training of image decomposition by using two parallel transformer encoders and an information fusion module based on a multi-head prediction module. Apart from that utilizing the strength of supervised learning, several image restoration algorithms have been proposed [39,46,47]. Kim et al. [26] have used the invertible encoder-decoder architecture, where the first stage learns the object recognition network and the second stage produces low-quality and high-quality (HQ) images. Due to the high dependency on high-quality images, the algorithm is found less resilient across multiple forms of corruption. Yang et al. [42] proposed the feature distillation method to produce HQ features. Byun et al. [14] propose the network to model the Gaussian-Poisson noise distribution. Along with the requirement of a Gaussian-Poisson noise pair, the algorithm is computationally heavy compared to the proposed algorithm. The proposed algorithm fails to generalize against unseen noises. IDR [47] works on iterative generation of noisier-noisy images. The algorithm is effective in handling unseen and real-world noise and shows marginal improvement as compared to the existing works. These algorithms are trained on large amounts of data and aim to map the noisy images to the clean images. Due to the significant distribution shift of the noise signals, it is hard to map the unseen noise images to the clean images. To overcome these limitations, self-supervised approaches trained on different tasks have evolved. However, the primary limitation as observed in the literature is their repetitive training [38]. In contrast to the existing algorithms, the proposed algorithm utilizes a cost-effective encoder-decoder architecture that aims to segregate the noise signals from the clean images in place of suppressing them into the clean images.

2 Proposed Signal Separator for Robustness

The proposed Dual-Tail (DuTa) encoder-decoder architecture used to separate the clean image noise and the noise pattern from the noisy input images is shown in Fig. 1. In contrast to the traditional trend of either mitigating the impact of noise using a denoising auto-encoder or detecting whether the images are noisy or clean, in this research, we propose not to only generate the clean images from their noisy counterpart but also to extract the noise pattern as well. The intentional modification of an input image through noise leaves its fingerprints in the images [32]. These fingerprints can be considered unwanted signals and need to be removed so that the original decision of a classifier can be restored.

The proposed architecture consists of seven encoding layers to effectively learn the latent representation of the input images. These latent representations of images are bifurcated into two branches to segregate the representation related to the clean image signals and noise signals. The encoding layers significantly reduce the size of the input images; therefore, to decode both clean and noise patterns, transpose convolutional layers are adopted. The configuration of the encoder architecture can be described as follows: the first two convolutional layers contain 64 filters each dimension 3×3. In the subsequent pair of convolution, the number of filters is increased to better encode the

Table 1. Severity scale of different corruptions used to perturb the clean images of different datasets.

Noise	Scale	Noise	Scale	Noise	Scale
GN	0.08	IN	0.03	SN	60
SPKN	0.15	UN	0.3	SPN	0.1

latent representation helpful for signal separation. In each pair of convolutional layers, the number of filters is double the filters used in the previous pair of layers. In the later layers of encoder architecture, the max-pooling operation has been applied to retain the useful information only and limit the computational cost by reducing the resolution of the feature maps.

Both the tails of the proposed architecture contain the 6 transpose convolutional layers to map each of the convolution outputs produced by the convolutional layers used in the encoder architecture. The number of filters in the decoding layers follows the reverse pattern of those in the encoding layers. The reason is to generate images that are of the same resolution as the input images to learn an accurate mapping function. The final convolutional layer in the decoding architectures aims to generate the images whether clean or noise patterns added to the images. Similar to the generative adversarial networks (GANs), where the discriminator has access to the real data to map the noise vector into the real-world images, both the decoding branches have the associate data to map noisy input to clean input and noise patterns added in the images. Henceforth, the proposed network is trained end-to-end through the combination of two losses as defined below, where each aims to minimize the loss of the network by mapping the output images close to the provided target images (i.e., signal-1 and signal-2).

$$\begin{aligned} \mathcal{L}_{clean} &= ||y - \hat{y}||^2 \\ \mathcal{L}_{noise} &= ||n - \hat{n}||^2 \\ \mathcal{L}_{total} &= w_1 \cdot \mathcal{L}_{clean} + w_2 \cdot \mathcal{L}_{noise} \end{aligned} \tag{1}$$

where, n and $\hat{n}$ represents the target and predicted noise pattern, respectively. y and $\hat{y}$ represent the original and generated clean images, respectively. w_1 and w_2 are the weights assigned to both the loss. We used equal weight.

Implementation Details: The proposed architecture is trained using an 'Adam' optimizer. The batch size is set to 64 and trained for 50 epochs with the initial learning rate set to 0.001. The ReLU non-linearity is applied in the intermediate layers of both encoding and decoding architecture. The final convolutional layer in each decoding architecture uses the 'sigmoid' activation function. The code is written in the Keras library by utilizing the TensorFlow backend. In this research, the images are modified using a single corruption type, and hence the networks are trained on a single corruption.

Table 2. Accuracy on the clean images reported using two different CNNs on the CIFAR-10 and CIFAR-100 datasets. Along with that, the accuracies of corrupted images are reported to showcase the vulnerability of CNNs. The proposed unique architecture shows strength in mitigating the impact of each corruption and restoring the performance significantly. A represents the accuracy of corrupted images and P represents the accuracy of the purified images.

CNN	Clean	GN		IN		SN		SPKN		UN		SPN		Average	
		A	P	A	P	A	P	A	P	A	P	A	P	A	P
CIFAR-10															
ResNet	**91.81**	32.97	85.01	51.73	91.89	32.73	84.24	44.37	86.48	32.12	85.33	20.48	91.61	35.73	87.43
WRN16-8	**93.14**	45.21	87.80	63.52	93.09	45.33	87.61	58.17	89.30	41.70	88.45	20.20	93.03	45.69	89.88
CIFAR-100															
WRN28-10	**76.21**	24.71	64.28	26.81	75.67	22.95	65.08	31.60	67.56	22.52	65.89	6.64	75.73	22.54	69.03
WRN16-8	**74.57**	20.09	63.81	21.59	74.07	17.44	65.35	25.68	67.21	18.43	65.36	4.12	73.95	17.89	68.29

3 Experimental Results and Analysis

Datasets: In this research, we have used multiple popular object recognition datasets, namely CIFAR-10, CIFAR-100 [28], and a subset of ImageNet namely ImageNette[1] which belongs to the color object category, and F-MNIST [40], a grayscale dataset.

CNNs: On CIFAR-10, ResNet50 [21] and Wide-ResNet16-8 (WRN16-8) [44] have been used. Similarly, on the CIFAR-100 dataset, Wide-ResNet16-8 and Wide-ResNet28-10 (WRN28-10) have been used for evaluating vulnerability against image degradations. On the F-MNIST dataset, two custom models (4 and 8 layers deep) are built. On ImageNette, Xception [17] and MobileNet [24] architectures are used. We assert that the variant networks will ensure the effectiveness of DuTaNet and make the proposed study trustworthy. It is to be noted here that the networks are trained on the clean images of a training set of individual datasets and tested on the clean, noisy, and purified images of the test set of the corresponding dataset.

Image Degradations: In this research, we have used six image degradations namely (i) Gaussian Noise (GN), (ii) Salt & Pepper Noise (SPN), (iii) Shot Noise (SN), (iv) Speckle Noise (SPKN), (v) Uniform Noise (UN), and (vi) Impulse Noise (IN). The severity level of each corruption is described in Table 1.

3.1 Robustness Analysis

Since multiple datasets of varying characteristics are utilized, the section is further divided to present a comprehensive analysis covering factors considered in the evaluation.

On Low-Resolution Color Images. The results on the CIFAR-10 and CIFAR-100 datasets are reported in Table 2. The results can be described in terms of the type of

[1] https://github.com/fastai/imagenette.

noise corruption and the classifier. The ResNet and WRN16-8 classifier on the CIFAR-10 dataset yields an accuracy of 91.81% and 93.14%, respectively. Not only, the wider network has shown better recognition performance on clean images, but it shows higher resiliency on corrupted images as well. For instance, the Gaussian noise reduces the performance of the ResNet classifier up to 32.97%; whereas, the WRN classifier can retain the accuracy of 45.21%. However, both classifiers are found highly sensitive to natural noise corruption. As promised, need not worry, the proposed unique architecture namely DuTaNet can boost the recognition performance of both the classifier. For instance, on the shot noise (SN) corrupted images, the proposed network increases the recognition accuracy of ResNet and WRN16-8 by $\mathbf{51.51}\%$ and $\mathbf{42.28}\%$, respectively. The improvement in recognition performance due to the proposed noise mitigation network is significant, nearly eliminating the impact of corruption. DuTaNet is designed to be classifier-agnostic, allowing seamless integration with any classifier to safeguard against various noise corruptions. This adaptability is especially crucial given the susceptibility of most machine learning classifiers to such corruptions [31].

In terms of the different noise corruptions, we have observed that each noise corruption significantly reduces the performance of each CNN. Out of all the corruptions applied, the SPN proves highly sinister and reduces the performance of ResNet and WRN16-8 by more than 71% and 73%, respectively; However, surprisingly, the performance boost on the SPN stealthy images is also the highest. The IN noise shows the lowest reduction in the recognition performance on both networks; however, it does not impact the mitigation strength of the DuTaNet. It further shows that DuTaNet is *agnostic to the success rate* of the corruption. The reduction can directly be related to imperceptibility. While each corruption is applied of severity 1 [22], different corruption has different perceptibility scores (Fig. 4).

As observed from the CIFAR-10 analysis, the wider network shows better robustness as compared to the deep counterpart. The WRN28-10 classifier yields an accuracy of 76.41% on the clean set of CIFAR-100 which reduces significantly under the influence of each image corruption. However, the proposed unique architecture without any configuration change can increase the robustness of each network significantly. For instance, on the impulse noise images which yield an accuracy of 26.81%, the proposed network increases the performance to $\mathbf{75.67}\%$. In addition to the previous agnostic properties, *DuTaNet is found dataset generalized* as well. The final performance of the shallow and deeper WRN after the proposed noise signal separation network lies in the range of 0.3 to 1.78% only. Similar to the CIFAR-10 dataset, the SPN noise with a scale of 0.1 is found most stealthy among all the used corruptions. On the WRN16-8, the SPN corruption can reduce the performance up to $\mathbf{4.12}\%$ from 74.57%. The proposed '*DuTa*' network boosts the performance by $\mathbf{73.95}\%$. In brief, (i) the multifold effectiveness of the proposed algorithm shows that it is agnostic to classifier and corruption types, (ii) the shallow classifiers are found less robust than deep and wide architectures, (iii) different corruption have varying success rates; however, DuTaNet is agnostic to such nature and can defend low and high success corruptions.

On Low-Resolution Grayscale Images. The applicability of the proposed DuTaNet is also evaluated on the grayscale dataset namely F-MNIST as well to strengthen its

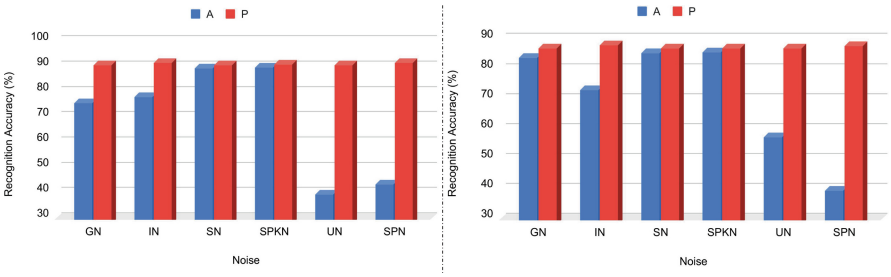


Fig. 2. Accuracy on the corrupted ('A') and purified images ('P') on the F-MNIST dataset using custom-1 (left) and custom-2 (right) models.

Table 3. Severity levels 2 and 3 of each corruption used to corrupt the clean images.

Severity	GN	UN	SPN	SN	IN	SPKN
2	0.12	0.3	0.3	25	0.06	0.2
3	0.18	0.1	0.5	12	0.09	0.35

Table 4. Corruption robustness of DuTaNet on ImageNet subset.

Attack		GN		IN		SN		SPKN		UN		SPN	
Level		2	3	2	3	2	3	2	3	2	3	2	3
Xception	A	55.30	25.60	53.90	25.00	47.20	25.00	68.10	50.10	30.00	59.80	15.80	0.00
	P	70.20	42.00	93.60	88.60	57.00	40.90	80.40	62.70	66.60	85.90	95.40	70.20
MobileNet	A	30.40	07.70	25.00	08.00	28.30	13.40	51.80	35.90	08.40	33.40	6.70	04.50
	P	49.50	25.00	95.90	87.60	49.90	36.30	72.90	58.10	46.80	69.80	93.10	53.70

impact. The results are reported in Fig. 2. Two custom CNNs with clean accuracy 91.45% (custom-1) and 87.96% (custom-2) are built. Interestingly, compared to the impact on the color object dataset, the impact of most of the corruption on grayscale and shallow CNNs is low. However, few corruptions can decrease the performance significantly, which the proposed DuTaNet can restore tremendously. For instance, the UN corruption drops the accuracy of the custom-1 model to $\mathbf{39.14\%}$ which DuTaNet improves to $\mathbf{90.09\%}$. Similarly, on the SPN corruption, DuTaNet improves the performance of the custom-2 model from $\mathbf{39.17\%}$ to $\mathbf{87.36\%}$. DuTaNet is agnostic to the color channel features of images, performing effectively on both color and grayscale images. This characteristic is crucial, as existing methods often exhibit bias towards these features, leading to reduced effectiveness [29].

On High-Resolution Images. While the above datasets are benchmark datasets in the AI community, they are of low resolution; therefore, to properly reflect the robustness of the proposed algorithm, we have used a high-resolution ImageNet subset. Being high resolution, it is observed that the corruption of low severity (Table 1) shows a negligible

Table 5. The comparison in the performance between (i) 'DuTa' and baseline and (ii) 'DuTa' and DCAE. The proposed 'DuTa' outperforms the baseline and DCAE by a significant margin across each corruption. Raw represents the accuracy of the corrupted testing set. The results are reported on CIFAR-10. The comparisons have been made with state-of-the-art works: [A] by Salman et al. [36] and [B] by Xie et al. [41].

Corrup.	CNN	Raw	Baseline	DCAE	[A]	[B]	DuTa
GN	ResNet	32.97	75.65	74.21	55.61	57.21	85.01
	WRN	45.27	80.04	79.30	60.04	45.69	87.80
IN	ResNet	51.73	78.17	79.75	58.17	65.20	91.89
	WRN	63.52	82.34	83.87	62.34	58.36	93.09
SN	ResNet	32.73	74.53	76.47	64.53	48.31	84.24
	WRN	45.33	78.76	80.02	58.76	46.89	87.61
SPKN	ResNet	44.37	75.94	77.90	55.94	71.05	86.48
	WRN	58.17	79.93	81.82	69.93	57.01	89.30
UN	ResNet	32.12	74.24	73.99	64.24	66.53	85.33
	WRN	41.70	77.93	78.08	67.93	61.00	88.45
SPN	ResNet	20.48	76.55	80.07	56.55	42.18	91.61
	WRN	20.20	80.98	84.02	60.98	35.76	93.03

Table 6. Comparison results against SOTA works. "Clean" indicates Top-1 clean accuracy (%) (higher is better). "mCE" shows the performance (%) over 15 corruption types (less is better).

Algorithm	CIFAR10-C		CIFAR100-C	
	Clean↑	mCE↓	Clean↑	mCE↓
Kim et al. [26]	75.3	45.6	57.9	48.1
Yang et al. [42]	77.2	41.9	58.0	46.4
Zou et al. [48]	62.3	45.1	55.2	45.8
Hendrycks et al. [23]	79.5	43.4	60.6	44.9
DuTaNet	**81.7**	**39.6**	**62.4**	**42.2**

impact on the CNNs: Xception and MobileNet. Hence, we have increased the severity level of corruption as shown in Table 3 to increase the attack success rate. The robustness results in Table 4 show that the proposed algorithm is effective in mitigating corruption on high-resolution images as well. On clean images, Xception achieves 97.00% which drops down to 0.00%; whereas MobileNet yields clean accuracy of 95.70%, which drops to 4.50% under the influence of SPN corruption. However, the proposed algorithm boosts the performance of both networks not only on the SPN images but also on any other corrupted images whether coming with average (level 2) or high severity (level 3). The above analysis showcases that DuTaNet is *agnostic to resolution and severity levels* of corruption.

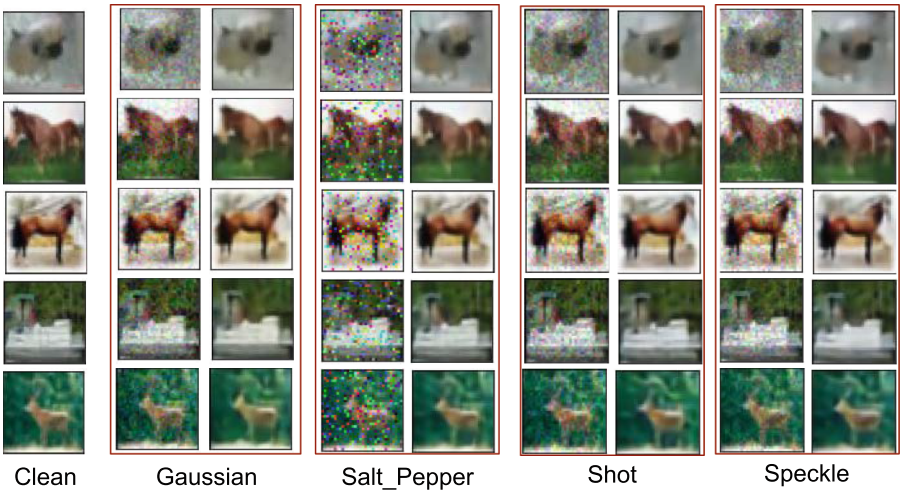


Fig. 3. Visualization of the impact of proposed DuTaNet in purifying the images. The first column is clean images. Later, each box contains corrupted (left) and purified images (right).

Table 7. PSNR/SSIM (↑) comparison with SOTA.

Algorithm	CIFAR10-C		CIFAR100-C	
	PSNR	SSIM	PSNR	SSIM
Wang et al. [39]	35.4	0.84	32.7	0.81
Zhang et al. [47]	37.5	0.86	34.3	0.82
Pang et al. [34]	34.8	0.82	32.7	0.80
Byun et al. [14]	38.7	0.91	36.1	0.84
Han et al. [20]	36.5	0.88	31.9	0.77
Wang et al. [38]	40.1	0.91	35.6	0.83
DuTaNet	**40.7**	**0.93**	**37.8**	**0.86**

3.2 Comparison with State-of-the-Art Works

To further strengthen the impact of the proposed research effort, a comprehensive comparison with existing state-of-the-art (SOTA) algorithms [A] by Salman et al. [36] and [B] by Xie et al. [41] has been performed. The comparison has been performed using multiple corruptions algorithms to not only reflect the capacity (in increasing the robustness against individual corruptions) but also the generalizability (handling unseen corruptions). As seen each corruption can degrade the recognition performance of the image classification networks. The proposed and existing mitigation algorithms aim to restore recognition accuracy. The results reported in Table 5 show that the existing algorithms perform significantly lower than the proposed algorithm. For example, when the algorithm [A] by Salman et al. [36] is used to mitigate the impact of corruption, it shows an improvement from 32.97% to 55.61% on the GN images, and the

performance is significantly lower (29.4%) than the proposed algorithm. Similar limitations of the algorithm [B] by Xie et al. [41] can be noticed whose performance is lower than the proposed algorithm by even more than 57% when SPN corruptions need to be handled. Further, the existing algorithms are found not generalized in handling different corruptions and yield poor recognition accuracy in comparison to one another. For example, [B] yields the recognition accuracy of 71.05% on SPKN images, however, the same algorithm can yield an accuracy of 35.76% only on SPN images. Whereas, the proposed algorithm shows consistent performance across the different types of corruption. On top of that, the existing algorithms have heavy computational demand to develop an image enhancement algorithm. For instance, Xie et al. [41] have utilized the concept of adversarial training and the algorithm took 52 h on 128 V100 GPUs to build a mitigation algorithm against PGD attack only. On the other hand, the proposed algorithm requires 20–30 minutes of training on an NVIDIA GeForce RTX 2080 GPU machine with the CUDA v11+ to achieve significantly better robustness. Apart from these algorithms, the performance of the proposed algorithm is at least 15% better than another recent network fine-tuned-based algorithm [12]. Another state-of-the-art defense algorithm namely RLAT [27] requires heavy data augmentation to improve the performance on corrupted images. The proposed DuTaNet does not require heavy data augmentation and yields at least 5% better performance than RLAT. The high performance of the DuTaNet against computationally heavy architectures makes it an ideal candidate for noise mitigation and deployment on mobile devices. The closest defense to the DuTaNet is MagNet [33] which uses the distribution shift as a reconstruction error of adversarial examples to detect the adversarial examples. However, noises can have different distributions, therefore, MagNet is not generalized against unseen and fails against complex attacks [1,6,15]. The rectifier part of MagNet is the baseline network used in this paper for comparison. We have also performed an extensive comparison with recent SOTA image-denoising algorithms and the results are reported in Tables 6 and 7. The existing algorithms utilize different evaluation metrics such as performance on clean images, mean corruption error (mCE) [26], Peak signal-to-noise ratio (PSNR), and Structural Similarity (SSIM) [20,38].

We have also conducted ablation studies and made the comparison with standard denoising convolutional auto-encoder (DCAE). The results of these ablation studies are also reported in Table 5. When the ResNet architecture is attacked by the impulse noise (IN) but protected using the baseline and DCAE architecture, the performance of both these architectures improves from 51.73% to 78.17% and 79.75%, respectively. However, the performance improvement is at least **12.14**% lower than the proposed *DuTaNet*. Similar higher robustness is observed when the WRN16-8 is used for object classification and protected against corruption. The higher robustness against each corruption and classification network establishes the importance of dual tail and a higher number of latent representation filter maps as we have asserted while building the proposed 'DuTaNet' signal separator. Figure 3 shows that the proposed DuTaNet can reduce the corruption produced by the different noise vectors. Henceforth, helps in improving the performance of the classifiers significantly.

4 On Generalization Capacity

The above comprehensive experiments and comparison with the state of the algorithms (SOTA) algorithms demonstrated the effectiveness of the proposed algorithm. The proposed algorithm not only has a low computational cost but can mitigate the impact of several corruptions and can increase of performance of almost 'any' CNN.

Table 8. Robustness of the proposed DuTaNet network under '***unseen severity strength***' of the corruptions. The DuTaNet network is trained on severity level 1 and tested on the same and higher severity to demonstrate its generalizability capacity. 1, 2 and 3 represent the severity level. A and P represent the accuracy of corrupted and purified images, respectively.

Severity →	1A	1P	2A	2P	3A	3P	1A	1P	2A	2P	3A	3P
Dataset: **CIFAR-10**												
Corruption ↓	ResNet						WRN28-10					
IN	51.73	91.89	30.82	91.59	21.52	91.20	63.52	93.09	35.79	92.90	21.74	92.76
GN	32.97	85.01	19.46	70.94	15.22	31.50	45.27	87.80	25.06	74.54	15.16	30.27
SPN	20.48	91.61	13.14	89.16	10.96	68.25	20.20	93.03	11.09	91.33	10.32	74.53
SN	32.73	84.24	18.13	67.06	14.97	31.57	45.33	87.61	25.72	71.54	17.77	32.31
UN	32.12	85.33	18.19	47.88	15.40	25.88	41.70	88.45	21.72	51.03	16.55	26.75
SPKN	44.37	86.48	32.51	83.32	18.60	40.93	58.17	89.30	46.02	86.67	26.00	45.31
Dataset: **CIFAR-100**												
Corruption ↓	WRN28-10						WRN16-8					
IN	26.81	75.67	10.79	75.36	6.99	74.55	21.59	74.07	7.97	74.11	4.66	73.35
GN	24.71	64.28	12.5	48.52	6.57	15.08	20.09	63.81	9.17	48.29	4.03	13.88
SPN	6.64	75.73	10.88	75.74	1.00	37.82	4.12	73.95	8.02	74.14	1.06	36.16
SN	22.95	65.08	10.87	39.26	6.67	13.08	17.44	65.35	7.78	37.81	3.88	11.99
UN	22.52	65.89	9.44	24.03	4.35	9.62	18.43	65.36	6.70	22.23	3.37	8.67
SPKN	31.60	67.56	20.82	61.64	8.18	20.24	24.68	67.21	15.40	61.36	6.27	17.75

To further showcase the strength of the proposed algorithm, we have tested its *generalizability* in several challenging scenarios: ❶ unseen severity level of the corruption, ❷ unseen corruption, and ❸ dual agnosticism: where both corruption and severity levels are unseen.

4.1 Unseen Severity Level

First, we describe the analysis concerning the unseen severity level where at the time of training one level of corruption has been used (shown in Table 1), while at the time of testing, images might be perturbed with different even higher (Table 3) corruption severity. The results of this scenario using extensive experiments are reported in Table 8. As

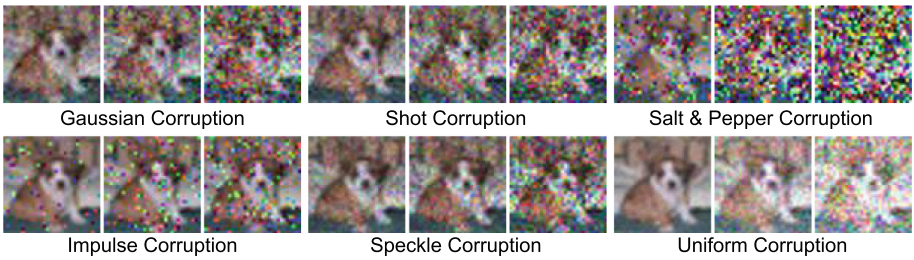


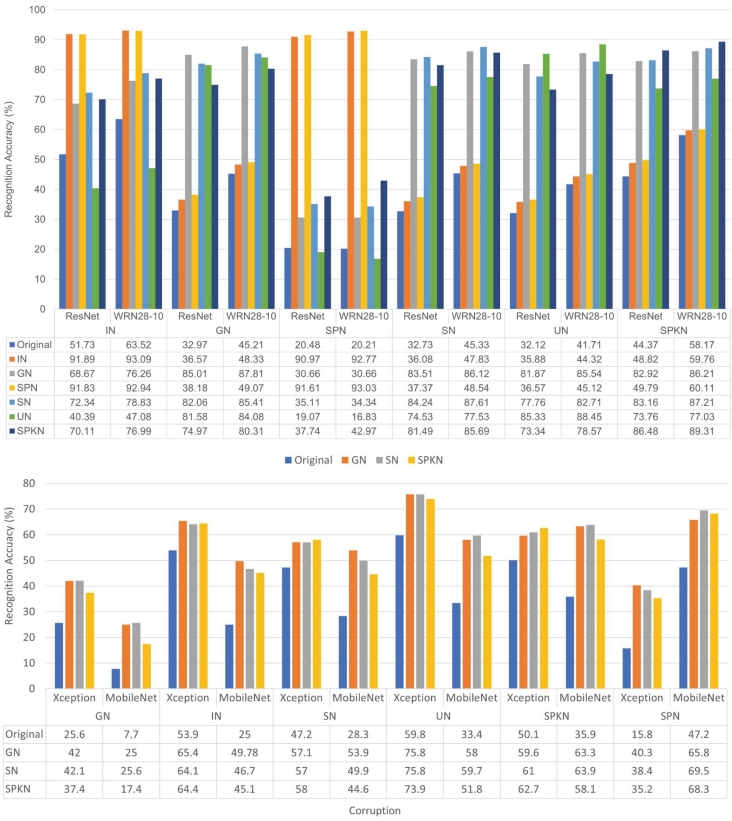
Fig. 4. Showcasing the impact of different corruptions and their associated severity levels. In each corruption, the leftmost image corresponds to severity 1 and the rightmost to severity 3.

expected as the severity level increases, the performance of each network working on different datasets decreases. For example, when the IN corruption has been used with severity 1, the performance of the ResNet network decreases from 91.81% to 51.73% on the CIFAR-10 dataset. However, as soon the strength of the corruption increases, the performance shows a steep decrement and drops to 21.52% with severity level 3. Interestingly, the proposed signal separator network which trained on severity 1 not only increased performance on severity level 1 corrupted images but also on severity level 3 images as well. Further, the increment is so significant that the proposed algorithm shows only the difference of 0.69% between the accuracy on severity 3 and severity 1. The boost in the performance is not specific to any network, a similar jump in the recognition accuracy can be seen on WRN28-10. We want to highlight that higher severity approximately destroys the visual appearance of the images (as shown in Fig. 4); even in such a scenario, the proposed algorithm shows robustness.

The proposed algorithm is agnostic to a dataset, CNN, corruption, and severity. It can be seen from the results that the proposed *DuTaNet* can increase the performance on CIFAR-100 images as well under unseen severity. The SPN corruption is found the most stealthy corruption due to destroying the visual image features (Fig. 4). For example, when the SPN noise with severity 1 is used, the performance of WRN28-10 drops from 76.21% to 6.64% which drops down further as the severity of corruption reaches 3, we observed a recognition accuracy of 1.00%. The WRN16-8 is also not robust to such corruption and shows a drop in the accuracy to the value of 1.06% when severity 3 of SPN is applied. The proposed DuTaNet can increase the performance of both networks from 1.0% to 37.82% and 1.06% to 36.16%.

4.2 Unseen Corruption

Next, we present the generalizability power of the proposed algorithm against unseen corruption. It can be easily assumed that if the purifier is trained on the corruption it is going to be tested, it will yield the best performance. We have observed a similar observation where the robustness accuracy is highest if the DuTaNet is trained and tested on the same corruption. However, that does not limit the capacity of the DuTaNet which shows the generalizability in handling unseen corruption. We assert that this robustness not only demonstrates the effectiveness of the DuTaNet but also reflects '*whether is*



	IN		GN		SPN		SN		UN		SPKN	
	ResNet	WRN28-10	ResNet	WRN28-10	ResNet	WRN28-10	ResNet	WRN28-10	ResNet	WRN28-10	ResNet	WRN28-10
Original	51.73	63.52	32.97	45.21	20.48	20.21	32.73	45.33	32.12	41.71	44.37	58.17
IN	91.89	93.09	36.57	48.33	90.97	92.77	36.08	47.83	35.88	44.32	48.82	59.76
GN	68.67	76.26	85.01	87.81	30.66	30.66	83.51	86.12	81.87	85.54	82.92	86.21
SPN	91.83	92.94	38.18	49.07	91.61	93.03	37.37	48.54	36.57	45.12	49.79	60.11
SN	72.34	78.83	82.06	85.41	35.11	34.34	84.24	87.61	77.76	82.71	83.16	87.21
UN	40.39	47.08	81.58	84.08	19.07	16.83	74.53	77.53	85.33	88.45	73.76	77.03
SPKN	70.11	76.99	74.97	80.31	37.74	42.97	81.49	85.69	73.34	78.57	86.48	89.31

	GN		IN		SN		UN		SPKN		SPN	
	Xception	MobileNet	Xception	MobileNet	Xception	MobileNet	Xception	MobileNet	Xception	MobileNet	Xception	MobileNet
Original	25.6	7.7	53.9	25	47.2	28.3	59.8	33.4	50.1	35.9	15.8	47.2
GN	42	25	65.4	49.78	57.1	53.9	75.8	58	59.6	63.3	40.3	65.8
SN	42.1	25.6	64.1	46.7	57	49.9	75.8	59.7	61	63.9	38.4	69.5
SPKN	37.4	17.4	64.4	45.1	58	44.6	73.9	51.8	62.7	58.1	35.2	68.3

Fig. 5. Generalizability of the proposed algorithm against 'unseen corruption'. Rows show the corruption name on which the proposed DuTaNet is trained and columns represent the corruption that we aim to defend. Original refers to the scenario where no purification has been applied and individual corrupted images are used for classification. Best viewed in color.

there any connection among corruptions'. For example, when DuTaNet aims to mitigate IN corruption, SN corruption shows the highest robustness on both networks. GN and SPKN corruption also shows that this corruption can effectively mitigate impulse noise (IN). Whereas, it is observed that uniform noise (UN) shows the lowest robustness reflecting its lowest similarity with IN. However, UN shows significant success in separating the Gaussian noise (GN) signal from the clean images and boosting the performance of both ResNet and WRN28-10. For example, UN-trained DuTaNet improves the recognition accuracy of ResNet from 32.97% to 81.58% and the performance of WRN28-10 boosted from 45.21% to 84.08%. We have observed that the SN shows the best robustness compared to other corruptions as it effectively mitigated IN, GN, UN, and SPKN unseen corruptions. "*We assert that such missing knowledge from the literature can boost the development of a robustness algorithm*". We want to highlight that recent works also put a step forward in tackling natural and adversarial perturbations;

however, they deal with a simple problem, i.e., binary classification [3,5]. Figure 5 (top) shows the results related to the unseen corruption-agnostic nature of the proposed algorithm. We have compared the generability of the proposed algorithm with two recent blind image decomposition networks namely Han et al. [1] and Wang et al. [2]. The average performance of the proposed algorithm under unseen corruption is 6.4% and 5.8% better than Han et al. [20] and Wang et al. [38], respectively. On top of that, the cost-effectiveness of the proposed algorithm can hugely help in building secure and trustworthy AI systems even on mobile devices.

4.3 Unseen Noise Type on ImageNet Dataset

An exciting and strong feature of the proposed signal separator network compared to the existing defense work is the handling of inherent perturbations [10]. We observe that our framework can boost the performance on noisy images by at least $\mathbf{12}\%$. The prime reason can be that the proposed algorithm tries to separate noise in place of generating two image signals (both containing raw images such as [48]) which can not remove inherent or minute noises present in any signal. *Consequently, we observed that the natural noise-trained DuTaNet mitigates the PGD adversary and increases the performance by* $\mathbf{45}\%$.

4.4 Dual Generalization

Further, we have evaluated the dual generalizability of the proposed DuTaNet on the high-resolution images of ImageNette, and the results are reported in Fig. 5 (down). To showcase this observation, we have selected the three best-performing corruptions observed above and used them for training the proposed noise separator namely DuTaNet. Foremost, in terms of network, MobileNet architecture is found highly sensitive against each corruption as compared to Xception except for SPN corruption. For example, the performance of MobileNet is $3\times$, $2\times$, and $1.7\times$ lower than Xception on GN, IN, and UN, respectively. Contrary to the general belief that the corruption domain seen both in training and testing will yield the best robustness, on high-resolution images, we have observed that it is not always true. For example, when the DuTaNet trained on SN corruption is used to mitigate the GN corruption, on MobileNet it yields 0.6% better performance than the DuTaNet trained on GN corruption itself. A similar phenomenon can be seen when GN corruption-trained DuTaNet yields 4% better robustness in handling SN corruption than SN itself. These exciting analyses reveal that there might be a strong correlation between corruption. The identification of correct correlation can help in reducing the load where we will not need to develop a specific defense for individual attacks and can help in building a robust and generalized defense. In brief, the proposed DuTaNet demonstrates the high effectiveness towards both the dimension of attack, i.e., corruption and severity. These high generalization capacities make the proposed DuTaNet ready for its deployment in the real world where the attack features will be unconstrained.

Even being the shallow architecture, the proposed DuTaNet is found cost-effective and generalized. To further demonstrate that, in place of utilizing the shallow encoder-decoder architecture in the DuTaNet, we have used the ResNet as the base

architecture for the encoder and decoder. With the increased complexity of the architecture, as expected, the computational cost of the defense system also increased. Second, we also found that the architecture is significantly less generalized than DuTaNet. For example, on average, the dual generalizability and unseen corruption generalizability of *ResNet based purifier* is found $\mathbf{45}\%$ and $\mathbf{32}\%$ lower than *DuTaNet*, respectively. The analysis is consistent where in DuTaNet we tried to increase the number of layers/filters, we observed the issue of overfitting.

5 Conclusion

Natural noise corruption is a powerful form of image degradation that can significantly impair the performance of neural networks, even without any prior knowledge of the network's structure. In recent literature, these stealthy corruptions have received relatively less attention compared to adversarial perturbations. To address this gap, we have proposed a dual-tail (DuTa) encoder-decoder architecture, DuTaNet, designed to purify noisy images and enhance recognition performance. DuTaNet is versatile and robust across various datasets, classifiers, types of corruption, and severity levels, making it well-suited for real-world applications. Additionally, this study, for the first time, identifies potential connections between different types of corruptions. It also shows that deeper and wider networks tend to be more robust than shallower ones. We believe this research will contribute to better benchmarking of classifier robustness against natural image corruptions.

References

1. Agarwal, A., Goswami, G., Vatsa, M., Singh, R., Ratha, N.K.: Damad: database, attack, and model agnostic adversarial perturbation detector. IEEE TNNLS **33**(8), 3277–3289 (2022)
2. Agarwal, A., Ratha, N., Vatsa, M., Singh, R.: Impact of super-resolution and human identification in drone surveillance. In: IEEE WIFS, pp. 1–6 (2021)
3. Agarwal, A., Ratha, N., Vatsa, M., Singh, R.: Benchmarking robustness beyond l_p norm adversaries. In: IEEE ECCVW (2022)
4. Agarwal, A., Ratha, N., Vatsa, M., Singh, R.: Crafting adversarial perturbations via transformed image component swapping. IEEE TIP **31**, 7338–7349 (2022)
5. Agarwal, A., Ratha, N., Vatsa, M., Singh, R.: Exploring robustness connection between artificial and natural adversarial examples. In: IEEE/CVF CVPRW, pp. 179–186 (2022)
6. Agarwal, A., Singh, R., Vatsa, M., Ratha, N.: Image transformation-based defense against adversarial perturbation on deep learning models. IEEE TDSC **18**(5), 2106–2121 (2021). https://doi.org/10.1109/TDSC.2020.3027183
7. Agarwal, A., Singh, R., Vatsa, M., Ratha, N.: Ibattack: being cautious about data labels. IEEE TAI **4**(6), 1484–1493 (2022)
8. Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Parameter agnostic stacked wavelet transformer for detecting singularities. Inf. Fusion **95**, 415–425 (2023)
9. Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Corruption depth: analysis of DNN depth for misclassification. Neural Netw. **172**, 106013 (2024)
10. Agarwal, A., Vatsa, M., Singh, R., Ratha, N.K.: Noise is inside me! generating adversarial perturbations with noise derived from natural filters. In: IEEE/CVF CVPRW, pp. 3354–3363 (2020)

11. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: IEEE/CVF ICCV, pp. 3155–3164 (2019)
12. Benz, P., Zhang, C., Karjauv, A., Kweon, I.S.: Revisiting batch normalization for improving corruption robustness. In: IEEE/CVF WACV, pp. 494–503 (2021)
13. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: IEEE CVPR, vol. 2, pp. 60–65 (2005)
14. Byun, J., Cha, S., Moon, T.: FBI-denoiser: fast blind image denoiser for Poisson-Gaussian noise. In: IEEE/CVF CVPR, pp. 5768–5777 (2021)
15. Carlini, N., Wagner, D.: Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. arXiv preprint arXiv:1711.08478 (2017)
16. Chang, M., Li, Q., Feng, H., Xu, Z.: Spatial-adaptive network for single image denoising. In: ECCV, pp. 171–187 (2020)
17. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: IEEE CVPR, pp. 1251–1258 (2017)
18. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE TIP **16**(8), 2080–2095 (2007)
19. Guo, Q., et al.: Watch out! motion is blurring the vision of your deep neural networks. NeurIPS **33**, 975–985 (2020)
20. Han, J., et al.: Blind image decomposition. In: ECCV, pp. 218–237. Springer (2022)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR, pp. 770–778 (2016)
22. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)
23. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: a simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)
24. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
25. Huang, Y., et al.: Simultaneous denoising and super-resolution of optical coherence tomography images based on generative adversarial network. Opt. Express **27**(9), 12289–12307 (2019)
26. Kim, I., Han, S., Baek, J.W., Park, S.J., Han, J.J., Shin, J.: Quality-agnostic image recognition via invertible decoder. In: IEEE/CVF CVPR, pp. 12257–12266 (2021)
27. Kireev, K., Andriushchenko, M., Flammarion, N.: On the effectiveness of adversarial training against common corruptions. In: UAI, pp. 1012–1021. PMLR (2022)
28. Krizhevsky, A., Hinton, G., et al.: Learning Multiple Layers of Features from Tiny Images (2009)
29. Kurmi, V.K., Sharma, R., Sharma, Y.V., Namboodiri, V.P.: Gradient based activations for accurate bias-free learning. In: AAAI (2022)
30. Liu, D., Wen, B., Jiao, J., Liu, X., Wang, Z., Huang, T.S.: Connecting image denoising and high-level vision tasks via deep learning. IEEE TIP **29**, 3695–3706 (2020)
31. Mahmood, K., Mahmood, R., Van Dijk, M.: On the robustness of vision transformers to adversarial examples. In: IEEE/CVF ICCV, pp. 7838–7847 (2021)
32. Marra, F., Gragnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: IEEE MIPR, pp. 506–511 (2019)
33. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: ACM SIGSAC CCS, pp. 135–147 (2017)
34. Pang, T., Zheng, H., Quan, Y., Ji, H.: Recorrupted-to-recorrupted: unsupervised deep learning for image denoising. In: IEEE/CVF CVPR, pp. 2043–2052 (2021)
35. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: MICCAI, pp. 234–241 (2015)

36. Salman, H., Sun, M., Yang, G., Kapoor, A., Kolter, J.Z.: Denoised smoothing: a provable defense for pretrained classifiers. NeurIPS **33**, 21945–21957 (2020)
37. Shijila, B., Tom, A.J., George, S.N.: Simultaneous denoising and moving object detection using low rank approximation. Futur. Gener. Comput. Syst. **90**, 198–210 (2019)
38. Wang, C., Zheng, Z., Quan, R., Sun, Y., Yang, Y.: Context-aware pretraining for efficient blind image decomposition. In: IEEE/CVF CVPR, pp. 18186–18195 (2023)
39. Wang, Z., Liu, J., Li, G., Han, H.: Blind2unblind: self-supervised image denoising with visible blind spots. In: IEEE/CVF CVPR, pp. 2027–2036 (2022)
40. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
41. Xie, C., Wu, Y., Maaten, L.V.D., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: IEEE CVPR, pp. 501–509 (2019)
42. Yang, Z., Dong, W., Li, X., Wu, J., Li, L., Shi, G.: Self-feature distillation with uncertainty modeling for degraded image recognition. In: ECCV, pp. 552–569. Springer (2022)
43. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational denoising network: toward blind noise modeling and removal. NeurIPS **32** (2019)
44. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
45. Zamir, S.W., et al.: Learning enriched features for fast image restoration and enhancement. IEEE TPAMI **45**(2), 1934–1948 (2022)
46. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE TIP **26**(7), 3142–3155 (2017)
47. Zhang, Y., Li, D., Law, K.L., Wang, X., Qin, H., Li, H.: IDR: self-supervised image denoising via iterative data refinement. In: IEEE/CVF CVPR, pp. 2098–2107 (2022)
48. Zou, Z., Lei, S., Shi, T., Shi, Z., Ye, J.: Deep adversarial decomposition: a unified framework for separating superimposed images. In: IEEE/CVF CVPR, pp. 12806–12816 (2020)

Utilizing Deep Incomplete Classifiers to Implement Semantic Clustering for Killer Whale Photo Identification Data

Alexander Barnhill[1(✉)], Jared R. Towers[2], Elmar Nöth[1], Andreas Maier[1], and Christian Bergler[3]

[1] Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
alexander.barnhill@fau.de

[2] Bay Cetology, 257 Fir Street, Alert Bay, BC V0N 1A0, Canada

[3] Department of Electrical Engineering, Media and Computer Science, Technical University of Applied Sciences Amberg-Weiden, Amberg, Germany

Abstract. Machine-assisted photo identification processes require significant amounts of data for each member of a population of interest but offer the possibility to alleviate a significant amount of manual effort. Gathering such data is time consuming and opportunistic, leading to imbalanced datasets ill-suited for traditional machine (deep) learning efforts. Incomplete classifiers, trained on a subset of classes in a population, can be initially useful to identify the most commonly seen individuals. This study investigates the use of incomplete classifiers trained on a subset of often-observed individual killer whales to generate latent space representations of the larger population containing unseen individuals. These semantically relevant representations are subsequently clustered to investigate the efficacy of this method as a secondary identification mechanism. This method proves to be robust to a significant amount of noise while being able to isolate individuals unknown to the classifier when applying limited expert knowledge to the approximate size of the population.

Keywords: Killer whale · classification · photo identification

1 Introduction

The photo identification process of killer whales *(Orcinus orca)* has historically been manually completed [3,6]. The identification of individuals is reliant on a number of primary factors including dorsal fin shape and size, the shape and color of the saddle patch – a dorso-posterior and ventro-anterior flare – and any markings or scars to the body [19]. The process of building a complete population overview is difficult, time consuming, expensive, and opportunistic, as it is often unclear where the individuals of the population are at any given time, and their

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 346–361, 2025.
https://doi.org/10.1007/978-3-031-78107-0_22

overlap with locations of the photographers leads to a dataset which displays the long-tailed effect [23], where a few individuals are photographed many times.

These data collection constraints and effects hold true for the Bigg's killer whale population, located in the northeastern Pacific Ocean. Field research began on this population in the 1970's and by 2018 766 individuals had been identified[19]. Due to the opportunistic nature of the data collection of these individuals, initial attempts at automating the identification process and addressing this imbalanced dataset focused on the 100 most commonly seen individuals out of the 367 living individuals between 2011 and 2017 [2] and achieved an accuracy of 92.5% with 97.2% top-3 unweighted accuracy (TUA). Automation of the identification process for additional cetaceans using machine (deep) learning is becoming more prevalent as computing costs are reduced and the availability of data is increasing [7,14,18].

Two additional problems arise with automated detection over extended periods of time: (1) the population is not fixed – members die and new ones are born –, and (2) the identifying marks of the animals can change. Significant changes in body structure in killer whale adults are relatively rare. The evolution of the appearance of the animal plays a large role in its identification. Photos of each age class of an individual are important as the they progress into adulthood, and identification photos of the individuals in the earlier phases of life are particularly important as the body changes quickly [19]. The first issue largely describes the open set problem [1,8,17], while the second one can be mitigated with data analysis and management. The data used here can be described as a *semi-open set*, due to several factors including (1) the fact that new individuals are born and added to the set over long periods of time, (2) that some individuals are much more elusive than others, leading to their data being generally unsuitable for use in supervised classification due to lack of samples and therefore often excluded from the problem [2], although they are still confined within the larger overall set of individuals, (3) these groups examined here do not intermingle, meaning that it is highly unlikely for individuals of another population or species to be photographed together, and (4) it is conceivable that, due to the opportunistic nature of data collection for these animals, that one or more individuals within a group has simply not been captured in a quality photo which can therefore be used in identification purposes.

All machine-assisted identification requires acquiring an efficient representation, either by using standard computer vision feature extraction techniques [13,15], or by learning one [2,7,14]. Learned representations can either make use of identifying prior information, such as in the case of supervised learning, or simply rely on the structure of the input as in unsupervised learning. Supervised learning requires at least some labeled data which can be challenging and expensive to acquire, and in open systems, having labels for all classes may be infeasible or impossible. Unsupervised learning, while requiring no labels, may not reliably maintain a space which ensures reasonable separability between classes.

Supervised learning for animal photo identification is more directly applicable due to the large amount of labeled data necessary for population monitoring

[2,7,14]. Direct application of unsupervised learning to, for example, learn some low-level embedding, has limited uses and this process is often seen as an initial step for down-stream supervised classification tasks [9,14,24].

This study investigates the use of *incomplete supervised classifiers*, which are herein defined as classifiers trained on a subset of known individuals of a given population, to generate semantically separable representations for downstream clustering and possible identification, as well as the robustness of this method with respect to image quality and the impact of known metadata and background content. It is investigated if these classifiers can generate representations of unknown individuals which are similarly grouped in a high-dimensional space which can be interpreted as the same or similar individuals as well as the effect image corruption and the presence of background has on these clusters. It is also investigated to what degree the method for encoding images is affected by the group to which the individual belongs, in this case either a different population of killer whales, or a different dorsal-finned species altogether. Thereby it is shown if utilizing incomplete classifiers can be utilized in down-stream identification tasks separately and parallel to traditional supervised classification in a closed-set problem and/or if they are useful for capturing meta-data such as photographer or location. The clustering algorithm here is kept simple (k-Means) to solely focus on differences in representation, rather than the efficacy of more advanced clustering mechanisms.

2 Related Work

Deep Learning-assisted identification of cetaceans has been performed successfully in a number of scenarios. Patton et al. [14] have successfully utilized deep metric learning to not only differentiate between 25 unique cetacean species, but to also identify individuals within those species, some with only a single training example. Konovalov et al. [12] have utilized off-the-shelf deep learning tools to identify, with 93% prediction accuracy, one the 76 most common dwarf minke whales in the waters around the northern Great Barrier Reef. Bogucki et al. [4] achieved an accuracy of 87% when applying deep learning to the identification of the endangered North Atlantic right whales, again using a pipeline of networks to identify a region of interest, and subsequently to scale and rotate an the image to classify to remove as much variance as possible.

Open set recognition [1,17], recognizing instances from classes not seen during training, is vital in real-world scenarios in which samples from unknown classes may not be readily available or easy to collect. Scheirer et al. [17] presented a 1-vs-set support vector machine (SVM) setup to identify out-of-set members. Bendale et al. [1] presented OpenMax as an alternative to the SoftMax loss function to be able to explicitly identify out-of-set instances. Wang et al. [20] have employed OneShot and unsupervised learning, implementing a novel Regularized Kernel Subspace Learning model, to explore the problem of re-identification (Re-ID) in an open-set problem, in a similar situation to what is presented here, in which representations are learned from unlabelled data and then applied to

the re-ID problem in an open set. This presents a more realistic scenario than a closed-set re-identification problem. Patton et al. [14] utilized a dummy class with a fixed similarity score in a discriminative learning context to identify when a new individual was present in a population of cetaceans.

Meta clustering [5], which deals with the idea that the optimal clustering is not always obvious, has been applied in various forms using mechanisms such as random feature weighting [5], or applying pseudo-labeling to a small subset of unknown data and using the learned centroids for proxy annotation [10].

Wheeldon and Serb [21] have recently investigated interpretability of latent space representations from autoencoders and classifiers, and have shown that classifiers generally provide representations leading to more homogeneous classifiers and that autoencoders do not seem to make full use of the latent space, with the majority of neurons saturating at 0 or 1 when encoding samples from the MNIST dataset.

To the best of the knowledge of the authors, this is the first study to investigate the clustering ability using representations from incomplete classifiers in a large open set problem and to show the robustness of this method as well as inspecting the level to which known metadata is a factor in clustering.

3 Data Material

Table 1. An overview of the datasets used in this research. The Bigg's killer whale Photo Identification Corpus (**BKWPIC**) acts as a superset for the Top 100-Extended (**T100-E**), Top 50 (**T50**), Top 10 (**T10**), Non-Top 100 (**N100**), and Non-Top 100 High Quality (**N100-HQ**) datasets. The Norwegian killer whale (**NKW-10**) and Bottlenose dolphin (**BD-23**) datasets are completely distinct. Along with the data source, the number of images in total ($\sum_{\text{Img}}$), the number of individuals represented within the dataset ($\sum_{\text{Ind}}$) as well as the average number of images per individual (μ_{Ind}^{Img}) are given.

Dataset	Source	$\sum_{\text{Img}}$	$\sum_{\text{Ind}}$	μ_{Ind}^{Img}
T100-E	Bigg's KW	62,740	100	$\approx$ 613
T50	Bigg's KW	39.352	50	$\approx$ 787
T10	Bigg's KW	11,447	10	$\approx$ 1,144
N100	Bigg's KW	19,171	213	$\approx$ 90
N100-HQ	Bigg's KW	5,148	158	$\approx$ 31
NKW-10	Norwegian KW	98	10	$\approx$ 9
BD-23	Bottlenose dolphin	4,515	23	$\approx$ 196

The data consists of two members of the family Delphinidae, namely the killer whale (*Orcinus orca*), and the bottlenose dolphin (*Tursiops truncatus gephyreus*). There are two populations of killer whale under investigation; both

the Bigg's killer whale and the Norwegian killer whale. The use of two populations of killer whale is to demonstrate the effectiveness of the methods presented here on a very similar, but genetically distinct population. An overview of these datasets including their source population/species can be seen in Table 1. The use of the bottlenose dolphin intends to show the effectiveness of the methods with data from a different species which nonetheless shares at least some of the same characteristics used for identification, namely the dorsal fin.

3.1 Bigg's Killer Whale Photo Identification Corpus (BKWPIC)

As seen in the data from [2], the data utilized here largely focuses on a single population of killer whales, namely the Bigg's population found in the north-eastern Pacific between California and Alaska. The corpus consists of 86,789 images, each of which contains only one of 367 individuals, which have been hand-labelled by experts, and spans a time period from 2011 to 2018. For each image, the body and dorsal fin of the animal was extracted using the FIN–DETECT element of the FIN–PRINT pipeline [2]. The resulting bounding box was up- or down-sampled to a 512 × 512 RGB image. An example of one such image can be seen in Fig. 1.

The dataset has been split into two main and non-overlapping datasets: (1) the Top-100-E (T100-E) dataset contains 62,740 images across the most commonly seen 100 individuals [2] from 2011–2017 which directly corresponds to the KWIDE11-17 dataset presented by Bergler et al. [2], and (2) the Non-Top-100 dataset (N100) which is comprised of 19,171 cropped images of every individual not present in the T100-E dataset which has at least 10 images, again between 2011–2017. This reduces the number of individuals present within the N100 from 267 to 213.

An additional subset of the N100 dataset was also created by selecting 3,117 left and 2,687 right sided images, representing 218 individuals (N100-HQ). The images were selected for their high quality and prominently visible saddle patch, rather than individual-image distribution and as such the images per individual range from 1 to 94, with the median being 18.5. After subsampling the dataset to remove instances of individuals having fewer than 10 images, 5,148 images across 158 individuals remained (2,624 left, 2524 right).

3.2 Norwegian Killer Whale Data Corpus (NKW-10)

Acting as an additional source of killer whale data, the Norwegian killer whale data corpus **(NKW-10)** consists of a selection of 10 individuals from the Norwegian killer whale population. These individuals, while sharing the same general identifying characteristics of the Bigg's killer whales, the saddle patch and dorsal fin display subtle differences in shape and size.

3.3 Bottlenose Dolphin Data Corpus (BD-23)

A collection of bottlenose dolphin images **(BD-23)** is also used to evaluate the performance of the encoding mechanisms presented within this research. The

corpus is comprised of 4,515 images across 23 female individuals. The images are all collected from the Patos Lagoon estuary, Brazil. In comparison to the killer whale, this group of female bottlenose dolphins displays significantly less variety in terms of dorsal fin shape and lacks the saddle patch which serves as a secondary identifier for the killer whale.

4 Methodology

4.1 FIN-DETECT

A YOLOv3-based network, which was trained on 2,286 manually labelled images as well as 7,511 pseudo-labelled images, was used to identify and extract cropped images, yielding a 512×512 RGB image of a single individual. This trained model is taken directly from [2] to maximize comparability.

4.2 FIN-IDENTIFY

The first method of encoding images utilized a ResNet18 classifier trained on one of three different datasets: (1) the most commonly seen 100 individuals **(T100-E)**, which achieved an overall accuracy of 92.5% with 97.2% top-3 unweighted accuracy (TUA) on the network test set, (2) the most commonly seen 50 individuals **(T50)** (93.1% accuracy, 97.1% TUA), and (3) the most commonly seen 10 individuals **(T10)** (92.9% accuracy, 98.1% TUA). In order to provide a consistent comparison to results presented in [2], only data from 2011 to 2017 was used for training. Feature vectors from the classifier were generated by taking the output from the last residual block and applying global average pooling to reduce the vector from 512×32×32 to 512×1×1

4.3 FIN-CODER

To fairly compare results from [2], a ResNet18-based undercomplete autoencoder was trained on the same 2011–2017 data used to train the FIN-IDENTIFY model (**T100-E**). The autoencoder used a latent size of 512 to keep the dimensionality consistent with the classifier. Both the FIN-IDENTIFY and FIN-CODER networks were trained on an NVIDIA Quadro RTX 5000 with the Adam optimizer ($\mathrm{lr} = 10^{-}5$, $\beta_1 = 0.5$, and $\beta_2 = 0.9$) and a batch size of 2 for the autoencoder and 8 for the classifier.

4.4 Image Corruption

In order to compare the semantic clustering capability of both encoding methods, varying levels of artificial noise in the form of a uniformly distributed $\mathcal{U}(0, 1)$ noise mask was applied to each channel of each image of the **N100** dataset before encoding. A percentage γ was chosen for the application of noise, indicating how many pixels of the original image were obscured by the noise mask, with $\gamma \in [0, 10, 25, 50, 75]\%$. See Fig. 1 for an example of the noise application process.

4.5 Killer Whale/Background Separation

Measuring the effect of background features on clustering performance was done by examination of the N100-HQ dataset. Due to the manual effort required to segment the images using the *Segment Anything* tool [11], namely that the tool performs initially quite poorly when trying to automatically segment the images, points on the dorsal fin and body of the individual had to be manually selected. These points were then given as assistance to the *Segment Anything* tool. The resulting masks were then applied to the images, substituting black pixels of the mask for white in the image, resulting in images containing only the body of the animal on a white background. The results of this process are visualized in Figs. 1 and 3.

4.6 Clustering

K-Means clustering was used to compare the viability of both the autoencoder and classifier as encoding mechanisms, with performance being evaluated primarily with V-Measure [16], calculated as the harmonic mean between homogeneity and completeness and defined as

$$v = \frac{(1+\beta) * \text{homogeneity} * \text{completeness}}{\beta * \text{homogeneity} + \text{completeness}}, \tag{1}$$

with $\beta = 1$ and the optimal values for all measures being 1. A perfect homogeneity indicates that the clusters contain only samples from one label, whereas a perfect completeness indicates that samples with one label are only present in one cluster. K-Means clustering was selected over other methods for its well-known performance as well as the generally low computing resources required.

5 Experiments

The experimental setup consists of the following five parts: **(1)** Utilizing both the autoencoder and classifier trained on the T100-E dataset to encode images from the N100 dataset, which have been corrupted by some level of noise γ, as detailed above, to generate latent representations which are subsequently clustered using the mechanism detailed above. The number of clusters used is equal to the number of individuals within the N100 dataset ($k = 213$). **(2)** Repeating this process using the classifiers trained on the T50 and T10 datasets. **(3)** Utilizing the representations generated by the classifier trained on the T100-E dataset of the N100-HQ dataset both with and without segmentation, again applying a level of noise γ, to examine the effect of the background on clustering. **(4)** Evaluation of the T100-E classifier as an encoding mechanism for images from two completely different groups of dorsal finned animals, namely the Norwegian killer whales and bottlenose dolphins, as described in the section above. In both of these cases the T100-E classifier is compared against an autoencoder trained

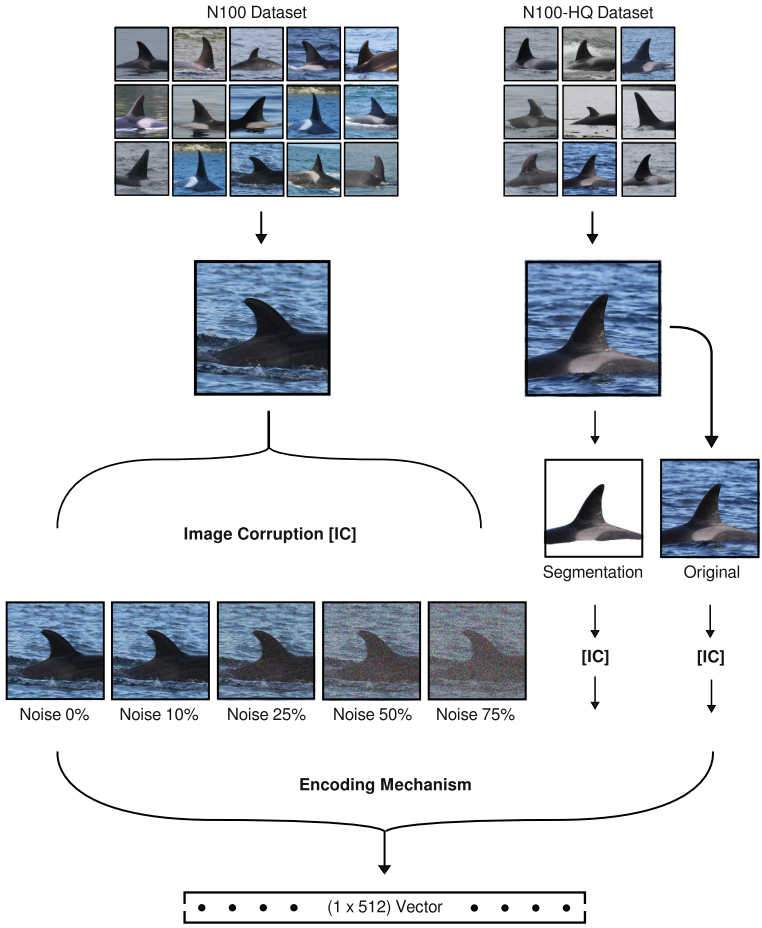


Fig. 1. A visualization of the pipeline which leads to the final 512-D vector used for clustering. This process involves applying some percentage γ of corrupting noise which may be preceded by the segmentation of the image to remove the background. The final (corrupted and/or segmented) image is then passed to the encoding mechanism of choice, either the unsupervised undercomplete autoencoder or the supervised classifier.

specifically on these individuals to investigate how the classifier handles the generation of embeddings for similar animals, as in the case of the Norwegian killer whales, as well as a completely different species, which lacks some of the identifying markers available for the killer whale, such as the saddle patch. Again, the number of clusters chosen for each of these experiments reflects the number of individuals known to be within the datasets ($k = 10$ and $k = 23$, respectively). **(5)** Finally, a more in-depth investigation of the clustering results of the N100

dataset was performed, with focus on meta features such as side of the individual, photographer, date, and location.

6 Results and Discussion

(1) Classifier Vs. Autoencoder. Clustering with the representations generated by the classifier trained on the T100-E dataset performed best when applied to the N100 data ($k = 213$), achieving a V-Measure of 0.71 in the complete absence of obscuring noise in comparison to the autoencoder representations which achieved a V-Measure of 0.21, indicating the inability of the autoencoder to adequately separate individuals. Furthermore, the T100-E encoder produced 14 perfectly homogeneous clusters, with images in those clusters ranging from 23 to 148. There were no clusters produced by the autoencoder which were perfectly homogeneous. Additionally, the T100-E encoding mechanism remains relatively robust to added noise, achieving a V-Measure of 0.71 and 0.68 when 10% and 25% of the pixels are corrupted. More significant performance losses occur when more than 50% of the images' pixels are corrupted, and the V-Measures decrease to 0.61 and 0.53 with 50% and 75% corruption, respectively. This performance indicates that the classifier is significantly more robust than the autoencoder when it comes to image corruption and requires significantly fewer features to effectively separate individuals it has never seen. Several examples of the input and output from the autoencoder trained on the T100-E dataset, as well as those trained on the NKW-10 and BD-23 datasets, can be seen in Fig. 2.

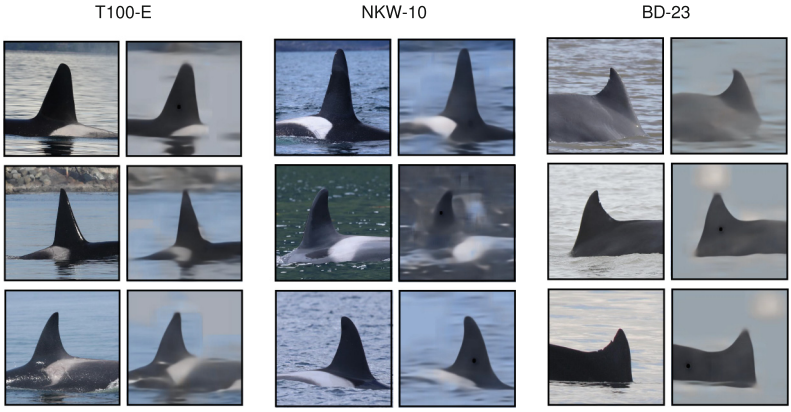


Fig. 2. Autoencoder input and output samples from the T100-E dataset, the Norwegian killer whale dataset (Copyright Ellyne Hamran, Ocean Sounds e.V.), and the bottlenose dolphin dataset (below, Copyright Lorenzo von Fersen, Nuremburg Zoo). Each input/output sample is generated from an autoencoder trained on data from the respective dataset

(2) Decreasing Target Count for Classifier. This performance is shown again when only using the T50 dataset for classifier training, which achieves a V-Measure of 0.69 and six perfectly homogeneous clusters with sizes ranging from 37 to 150 images. Performance degrades significantly when the T10 dataset is used for training, with the best result achieving a V-Measure of 0.46. These results are summarized in Table 2. Again, this demonstrates that an incomplete classifier, which has seen only a minor subset of the individuals within a population, is capable of separating individuals never before encountered.

(3) Effects of Segmentation on Clustering. When encoding for the segmented, body-only images ($k = 158$) the T100-E classifier again outperforms the AE, achieving V-Measures of 0.82 and 0.35 respectively. These scores are similar to those achieved using the same but unsegmented images, which resulted in V-Measures of 0.82 and 0.36. This similarity is likely due to the fact that the images selected are high quality, with the animal large and central within the image. However it is indicative that both approaches generally learn to ignore the background, but its removal may be beneficial to the classifier in providing better separation. As noise is increasingly added, the performance of the classifier on the original data declines more rapidly than on the segmented data, ending with V-Measures of 0.64 and 0.77, respectively. This may indicate that it is not necessarily the background that is having an effect on the clustering of the results, but rather the contrast of the fin to the background, which remains relatively consistent with the segmented images as more noise gets added. This is further supported by the fact that the images contained within this dataset are those in which the dorsal fin, the main identifying characteristic of the killer whale, is decidedly prominent within the frame. In contrast, the addition of noise to the images processed by the autoencoder showed no significant effect to the performance, with both the original and segmented producing V-measures between 0.37 and 0.35, regardless of absence or presence/amount of corrupting noise. A visualization of the samples used for this experiment, the segmented images with increasing levels of noise as well as the original, and the output of the autoencoder, is presented in Fig. 3. This may help to better understand the consistent results shown using the segmented images, as the variation in the background as seen in the autoencoder output is minimal in contrast to the original images, which may have led to the slightly worse performance. The results for this experiment are summarized in Table 3.

(4) Different Population and Different Species. The T100-E classifier also outperformed the autoencoder specifically trained on the 10 available Norwegian killer whales, achieving V-Measures of 0.52 and 0.31, respectively. This relative performance was similar for the bottlenose dolphins as well, reaching a V-Measure of 0.18 and 0.03 respectively. The poor performance of both methods for the dolphins may be attributed to either the limited quantity of data, the large variety of body positions, the relative similarity of the animals to each other, or the relatively low number of readily identifiable features, such as the

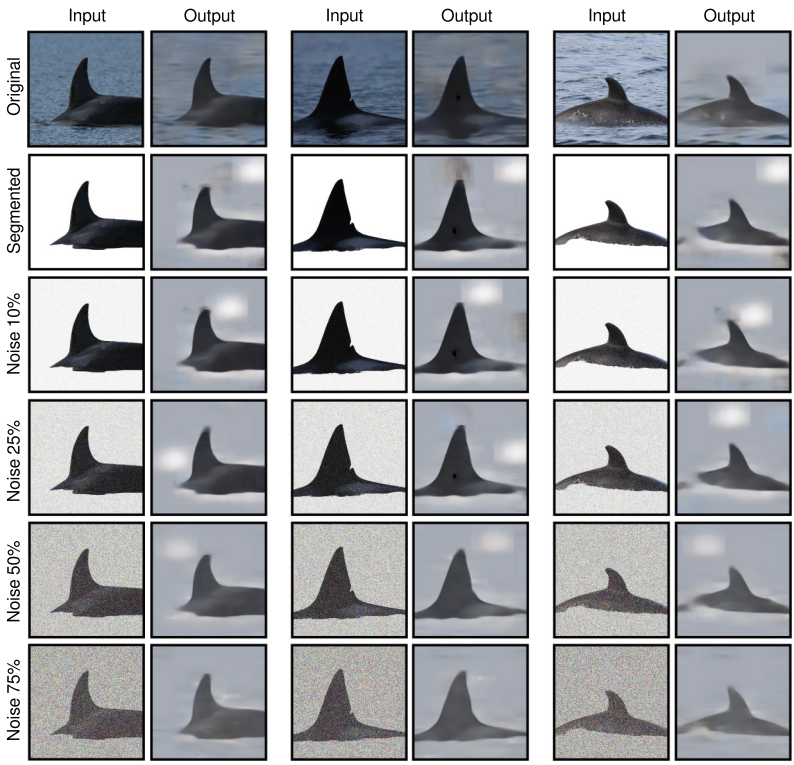


Fig. 3. Visualization of input and output samples of the autoencoder which was trained on the T100-E dataset when processing both the original, non-segmented images, and the images for which binary segmentation masks have been produced. In addition, the input and output of the segmented samples when applying an increasing amount of noise are also shown.

absence of a saddle patch as seen with the killer whales. As stated previously, visualizations of several samples processed by the autoencoders trained on the NKW-10 and BD-23 data corpora can be seen in Fig. 2.

(5) Meta Clustering Performance. Finally, a deeper investigation into the contents of the clusters was performed and additional metadata such as date, location, photographer, and side (left/right) were evaluated. It was revealed that, in the case of no added noise, the 14 perfectly homogeneous clusters contained images of 10 distinct individuals. Furthermore, the clusters show no tendency to favor a particular metadata characteristic such as date, location, photographer, or side, with the mix of left and right sides per cluster ranging from 51% to 81%

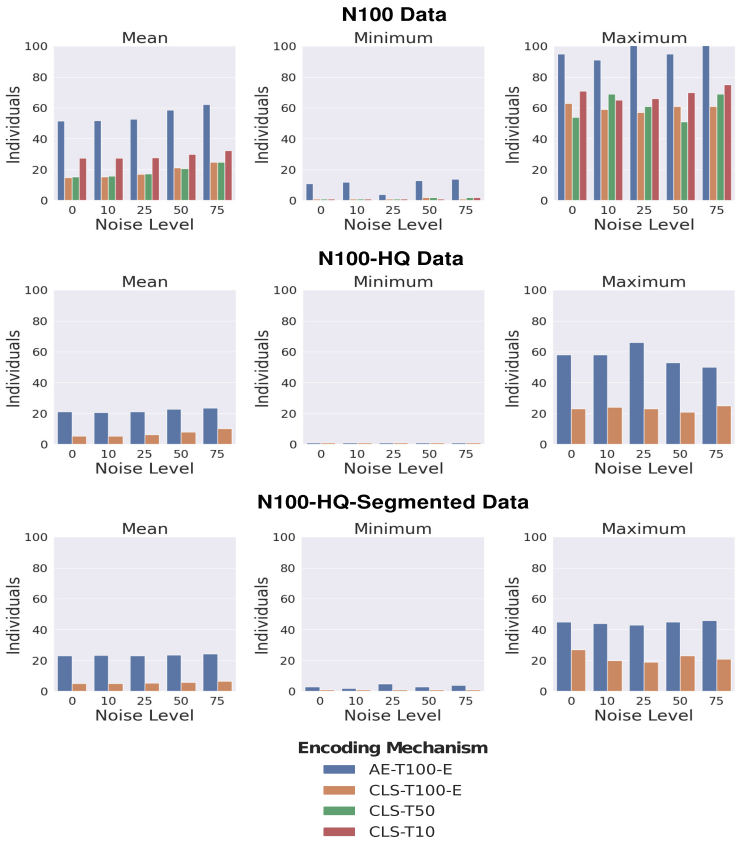


Fig. 4. The mean, minimum, and maximum number of individuals per cluster when varying the level of corrupting noise when encoding the N100 images (top), the N100-HQ images (middle), and the segmented N100-HQ images (bottom), with the various encoding mechanisms used for each. The larger the number of individuals, the more impure the cluster, indicating worse performance of the encoder with respect to individual identification

and the variability in combination of date, location, and photographer ranging from 0.18 to 0.49. This trend is also present in the high-noise scenario, where only four unique individuals in six perfectly homogeneous clusters are observed, but side purity remains between 55% and 83% and meta variability ranges between 0.16 and 0.63. A visualization of how the number of individuals within a cluster changes, including the mean, minimum, and maximum within each cluster can be seen in Fig. 4. This reflects varying the amount of noise and utilizing the various encoding mechanisms for the N100, N100-HQ, and segmented N100-HQ datasets.

Table 2. A summary of the clustering results, showing **V-Measure** for vectors generated from the $N100_{[noise]}$ dataset achieved with the Autoencoder (**AE**) or Classifier (**CLS**) trained on either 100, 50, or 10 most commonly observed animals (T100-E, T50, T10 datasets).

	$N100_0$	$N100_{10}$	$N100_{25}$	$N100_{50}$	$N100_{75}$
AE - Top 100-E	0.25	0.25	0.24	0.22	0.21
CLS - Top 100-E	0.71	0.71	0.68	0.61	0.53
CLS - Top 50	0.69	0.69	0.66	0.60	0.54
CLS - Top 10	0.46	0.46	0.45	0.42	0.40

Table 3. A summary of the clustering results of the **N100-HQ** data in both the original and segmented (**N100-HQ-S**) form with the vectors generated by either the Classifier (**CLS**) or Autoencoder (**AE**) trained on the most commonly seen 100 animals (**T100-E**). The **V-Measure** for each experiment is presented when applying varying percentage of image corrupting noise ($N_{[noise]}$), while the number of clusters remains constant, reflecting the number of individuals present within the dataset ($k = 158$).

	N_0	N_{10}	N_{25}	N_{50}	N_{75}
N100-HQ-CLS	0.82	0.81	0.78	0.71	0.64
N100-HQ-AE	0.36	0.36	0.35	0.35	0.35
N100-HQ-S-CLS	0.82	0.81	0.81	0.79	0.77
N100-HQ-S-AE	0.37	0.37	0.37	0.36	0.35

7 Conclusion and Future Work

This study investigates the use of incomplete classifiers and undercomplete autoencoders for the generation of latent vectors for clustering examples from unseen classes. It was additionally investigated how robust these mechanisms are to image corruption and the removal of background information. Through a comprehensive experimental setup including using classifiers trained on a varying number of classes and individuals, increasing the level of corrupting noise, and examining the difference in clustering performance when removing background content, the superiority of the incomplete classifier was demonstrated.

The classification separation mechanism appears to saturate as a certain number of classes is reached, as seen by the difference between the T100-E, T50, and the T10 classifiers, indicating the ability to use less data and fewer classes for the same general performance. Despite the relatively poor performance of the T10 classifier, all supervised classification mechanisms produce latent embeddings which are more easily separable and semantically relevant than those generated by the autoencoder. This demonstrates that, given a minimal subset of classes, robust embeddings useful for downstream clustering and potential identification is a possibility, which could be useful in a wide range of scenarios where inter-class variation is significant, but where data collection and labelling

is a tedious and expensive process. The usage of these incomplete classifiers may therefore be able to provide suggestions of images which may belong to a single class within a large dataset, thereby expediting downstream labelling and identification.

Finally, the homogeneous clusters produced by the classifier do not seem to discriminate between certain metadata such as only left or only right sided images or images from a certain day, location, or photographer, indicating the robustness of the classifier to generate embeddings which are relevant to a specific individual and lending more credence and reliability to their use as a preprocessing step for identification.

Future work will include comparing these methods with techniques such as discriminative learning for identifying unknown individuals and to investigate methods to identify further commonalities in clusters not related to known metadata and image quality (brightness, saturation, camera settings, etc.). Additionally, further image corruption and occlusion [22] techniques should be investigated in order to determine the relevance of certain regions in the image which lead to better separation in the latent space.

Acknowledgements. The authors would like to thank Ellyne Hamran with Ocean Sounds e.V. for the Norwegian killer whale photos used as comparison in this work, Lorenzo von Fersen with the Nuremburg Zoo for allowing the use of the bottlenose dolphin photos, and Gary Sutton and Tasli Shaw for their efforts in annotating the Bigg's killer whale data.

References

1. Bendale, A., Boult, T.E.: Towards open set deep networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1563–1572 (2016). https://doi.org/10.1109/CVPR.2016.173
2. Bergler, C., et al.: FIN-PRINT a fully-automated multi-stage deep-learning-based framework for the individual recognition of killer whales. Sci. Rep. **11**, 1–16 (2021). https://doi.org/10.1038/s41598-021-02506-6
3. Bigg, M., Olesiuk, P., Ellis, G.M., Ford, J., Balcomb, K.C.: Social organization and genealogy of resident killer whales (orcinus orca) in the coastal waters of British Columbia and washington state. Report Int. Whaling Commission **12**, 383–405 (1990)
4. Bogucki, R., Cygan, M., Khan, C.B., Klimek, M., Milczek, J.K., Mucha, M.: Applying deep learning to right whale photo identification. Conserv. Biol. **33**(3), 676–684 (Nov 2018). https://doi.org/10.1111/cobi.13226
5. Caruana, R., Elhawary, M., Nguyen, N., Smith, C.: Meta clustering. In: Sixth International Conference on Data Mining (ICDM'06). pp. 107–118 (2006). https://doi.org/10.1109/ICDM.2006.103
6. Ford, J.K., Ellis, G.M., Balcomb, K.C.: Killer whales: the natural history and genealogy of Orcinus orca in British Columbia and Washington. UBC press (1994)
7. Gómez Blas, N., de Mingo López, L.F., Arteta Albert, A., Martínez Llamas, J.: Image classification with convolutional neural networks using gulf of maine humpback whale catalog. Electronics **9**(5) (2020). https://doi.org/10.3390/electronics9050731, https://www.mdpi.com/2079-9292/9/5/731

8. Hassen, M., Chan, P.K.: Learning a neural-network-based representation for open set recognition, pp. 154–162. https://doi.org/10.1137/1.9781611976236.18, https://epubs.siam.org/doi/abs/10.1137/1.9781611976236.18
9. Hu, M., You, F.: Research on animal image classification based on transfer learning. In: Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering, pDp. 756–761. EITCE '20, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3443467.3443849
10. Jin, X., et al.: Meta clustering learning for large-scale unsupervised person re-identification. In: Proceedings of the 30th ACM International Conference on Multimedia. ACM (Oct 2022). https://doi.org/10.1145/3503161.3547900, https://doi.org/10.1145/3503161.3547900
11. Kirillov, A., et al.: Segment anything (2023). https://doi.org/10.48550/ARXIV.2304.02643, https://arxiv.org/abs/2304.02643
12. Konovalov, D.A., Hillcoat, S., Williams, G., Birtles, R.A., Gardiner, N., Curnock, M.I.: Individual minke whale recognition using deep learning convolutional neural networks. J. Geosci. Environ. Protect. **6**, 25–36 (2018)
13. Maglietta, R., et al.: DolFin: an innovative digital platform for studying risso's dolphins in the northern ionian sea (north-eastern central mediterranean). Sci. Reports **8**(1) (Nov 2018). https://doi.org/10.1038/s41598-018-35492-3
14. Patton, P.T., et al.: A deep learning approach to photo–identification demonstrates high performance on two dozen cetacean species. Methods in Ecology and Evolution (Jul 2023). https://doi.org/10.1111/2041-210x.14167
15. Renò, V., et al.: A sift-based software system for the photo-identification of the risso's dolphin. Ecol. Inform. **50**, 95–101 (2019). https://doi.org/10.1016/j.ecoinf.2019.01.006, https://www.sciencedirect.com/science/article/pii/S1574954118301377
16. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Eisner, J. (ed.) EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic, pp. 410–420. ACL (2007), https://aclanthology.org/D07-1043/
17. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(7), 1757–1772 (2013). https://doi.org/10.1109/TPAMI.2012.256
18. Thompson, J.W., et al.: finfindr: Automated recognition and identification of marine mammal dorsal fins using residual convolutional neural networks. Marine Mammal Sci. **38**(1), 139–150 (2022). https://doi.org/10.1111/mms.12849, https://onlinelibrary.wiley.com/doi/abs/10.1111/mms.12849
19. Towers, J.R. et al.: Photo-identification catalogue, population status, and distribution of bigg's killer whales known from coastal waters of British Columbia, Canada. Canadian Tech. Report Fisheries Aquatic Sci. **3311** (07 2019)
20. Wang, H., Zhu, X., Xiang, T., Gong, S.: Towards unsupervised open-set person re-identification. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 769–773 (2016). https://doi.org/10.1109/ICIP.2016.7532461
21. Wheeldon, A., Serb, A.: A study on the clusterability of latent representations in image pipelines. Front. Neuroinform. **17** (Feb 2023). https://doi.org/10.3389/fninf.2023.1074653, https://doi.org/10.3389/fninf.2023.1074653
22. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014, pp. 818–833. Springer International Publishing, Cham (2014)

23. Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
24. Zheng, X., Kellenberger, B., Gong, R., Hajnsek, I., Tuia, D.: Self-supervised pretraining and controlled augmentation improve rare wildlife recognition in uav images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 732–741 (October 2021)

FPMT: Enhanced Semi-supervised Model for Traffic Incident Detection

Xinying Lu and Jianli Xiao(✉)

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
222240474@st.usst.edu.cn, audyxiao@sjtu.edu.cn

Abstract. For traffic incident detection, the acquisition of data and labels is notably resource-intensive, rendering semi-supervised traffic incident detection both a formidable and consequential challenge. Thus, this paper focuses on traffic incident detection with a semi-supervised learning way. It proposes a semi-supervised learning model named FPMT within the framework of MixText. The data augmentation module introduces Generative Adversarial Networks to balance and expand the dataset. During the mix-up process in the hidden space, it employs a probabilistic pseudo-mixing mechanism to enhance regularization and elevate model precision. In terms of training strategy, it initiates with unsupervised training on all data, followed by supervised fine-tuning on a subset of labeled data, and ultimately completing the goal of semi-supervised training. Through empirical validation on four authentic datasets, our FPMT model exhibits outstanding performance across various metrics. Particularly noteworthy is its robust performance even in scenarios with low label rates.

Keywords: Traffic Incident Detection · Semi-supervised Learning · Generative Adversarial Networks

1 Introduction

In the realm of intelligent traffic systems, traffic incident detection refers to accurately identifying unpredictable incidents such as traffic accidents, road maintenance, and severe congestion in terms of both location and time [1]. Ensuring the efficient operation of urban traffic and enhancing the safety of people's travel constitute one of the core functionalities of intelligent traffic systems, involving the automatic detection of traffic incidents. This involves promptly identifying and addressing these incidents to improve overall traffic flow. However, relying on traffic flow data for incident detection necessitates continuous data collection and labeling by professionals, which is resource-intensive. Current research has predominantly focused on deep learning methods, which typically require substantial labeled data for training, presenting a significant challenge in applying deep learning to traffic incident detection with limited labeled data [2].

To address the scarcity of label information in real-world scenarios due to difficulty in acquisition, semi-supervised learning has garnered attention. Semi-supervised learning

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 362–374, 2025.
https://doi.org/10.1007/978-3-031-78107-0_23

successfully alleviates the over-reliance on supervised information in machine learning, leveraging unlabeled samples to train a reliable classifier for effective predictions in target categories based on the popular assumption and clustering hypothesis. Various semi-supervised learning models employ different strategies for handling unlabeled samples, such as entropy minimization, consistency regularization, and data augmentation. Examples include the Mean Teacher model [3] and Virtual Adversarial Training (VAT) [4]. However, these semi-supervised learning methods are based on the assumption that the distribution of labeled and unlabeled data pairs is entirely identical, treating labeled or unlabeled data separately [5].

Due to the limited availability of labeled data during the training process of semi-supervised learning, overfitting is prone to occur. To better utilize unlabeled data, Bertholot et al. proposed the MixMatch method [6]. This method generates mixed samples by interpolating different samples through MixUp and interpolates mixed pseudo-labels for different samples. The authors also introduced the FixMatch model [7], which achieves state-of-the-art performance in semi-supervised learning benchmarks by using weakly augmented unlabeled images to generate high-confidence pseudo-labels and training the model with strongly augmented image versions. However, these interpolation methods are designed for image data, and discrete text data, interpolation needs to be performed in the corresponding hidden space, leading to the development of a new semi-supervised learning method for text data called MixText [8].

To further explore the latent information in unlabeled data by utilizing more rational training, data augmentation, and loss calculation strategies, integrating their complementary strengths, a semi-supervised traffic incident detection model is proposed based on the MixText framework. In the data augmentation module, as traffic incident datasets often exhibit significant imbalances and small scales, the application of GANs is proposed to balance and expand the dataset. A probability pseudo-mixing strategy is employed in the hidden space when performing Mix-up, assigning confidence to samples entering the mixture, thereby giving more weight to samples with higher confidence to enhance regularization. In terms of training strategy, unsupervised training is initially performed on all data, followed by supervised training on a subset of labeled data, and ultimately, semi-supervised fine-tuning is conducted to improve detection rates. In the semi-supervised fine-tuning phase, pseudo-labels are first predicted for unlabeled data, and confidence is assigned to these pseudo-labels. When labeled and unlabeled data enter the model's hidden layer, interpolation is performed based on the confidence ratio, iterating through training to extract latent information and obtain classification results. The model is experimentally validated on four real datasets and compared with baseline models. Through ablation studies, the effectiveness of each module is demonstrated. The results show that the proposed semi-supervised traffic incident detection model FPMT performs exceptionally well with very limited labeled data.

In summary, the main contributions of this paper are as follows:

- Proposing a novel semi-supervised traffic incident detection model that exhibits outstanding performance in scenarios with extremely low label rates.
- Introducing GANs in the data augmentation module to balance and expand the dataset.
- Optimizing the interpolation strategy in the hidden layer under the MixText framework to enhance regularization.

- Applying a training strategy that involves supervised training initially, followed by semi-supervised fine-tuning to improve detection rates.
- Conducting extensive experiments on four real datasets, demonstrating the effectiveness of the proposed semi-supervised traffic incident detection model and validating the effectiveness of each module.

2 Related Work

The development of Traffic Incident Detection traces back to 1965 when the California algorithm [9] utilized fluctuations in upstream and downstream traffic flow data to identify the occurrence of incidents. Subsequently, the standard deviation algorithm [10] employed standard deviation values to observe the average trends of preceding time intervals and the current transformation trends, thereby discerning whether a traffic incident has occurred. Following this, Bayesian algorithms [11], rooted in statistical theory, sequentially emerged. However, they exhibited an overreliance on past experiences, posing flexibility challenges. With the robust growth of machine learning post-1990, classical models such as Support Vector Machines (SVM) [12] and Random Forests [13] were applied to this task. Various artificial intelligence algorithms found applications in the field of Traffic Incident Detection, including Convolutional Neural Networks (CNN) [14] and Long Short-Term Memory Neural Networks [15] as part of deep learning methods.

The development of semi-supervised learning commenced in 2005 when Grandvalet and others proposed the entropy minimization method [16], becoming the most classic and commonly used deep semi-supervised learning algorithm and strategy. This approach effectively integrates unlabeled data in semi-supervised learning, demonstrating robust performance, especially when addressing violations of generated model error specifications or "cluster assumptions." Subsequent developments in semi-supervised learning models are intricately tied to four aspects: entropy minimization, consistency regularization, data augmentation, and pre-training fine-tuning.

2.1 Consistency Regularization

In 2018, Tarvainen et al. introduced the Average Teacher Model based on Consistency Regularization [3], significantly improving performance compared to previous methods. In the same year, Miyato et al. proposed the Virtual Adversarial Training method [4], which involves computing the gradient of the network to generate adversarial samples. These adversarial samples are designed to maximize the network's vulnerability, and by combining them with pseudo-labels derived from the original samples, the network can be correctly trained, maximizing its robustness against interference. In 2020, Sohn et al. presented the FixMatch method [7], which involves applying slight transformations to unlabeled samples for initial predictions and selecting samples with high confidence to assign pseudo-labels. Subsequently, these samples undergo more substantial transformations, and the consistency loss is computed between the pseudo-labels and the blurred predictions after strong transformations, thereby enhancing the learning effectiveness of the network.

2.2 Entropy Minimization

In 2013, Lee proposed the Pseudo-Label method [17], which has become the most widely used semi-supervised learning approach. Essentially, it also leverages the strategy of entropy minimization employed by the network predictions. The Pseudo-Label method primarily involves selecting samples with high confidence during the learning process, transforming their network predictions into pseudo-labels corresponding to the class with the highest predicted probability. These pseudo-labels are then utilized to assist in the network training process. In 2016, Laine and Aila introduced the Π-model [5]. This method utilizes two structurally identical but parametrically distinct network models, aiming for both networks to produce the same predictions for identical samples. Consequently, when one network generates incorrect labels for unlabeled samples during training, the other network can correct them. This consistency training strategy avoids the robustness issues associated with the aforementioned entropy minimization strategy.

2.3 Data Augmentation

In 2019, Wang et al. proposed a straightforward yet effective semi-supervised learning method called Augmented Distribution Alignment [18]. This method employs adversarial training and interpolation strategies to alleviate sampling biases arising from limited labeled samples in semi-supervised learning. It aligns the empirical distributions of labeled and unlabeled data. In the same year, Bertholot et al. introduced a novel semi-supervised learning model named MixMatch [6]. By unifying current mainstream semi-supervised learning methods, this model infers low-entropy labels on augmented, unlabeled examples and utilizes MixUp technology to blend labeled and unlabeled data. Cai et al. presented Semi-ViT [19], another semi-supervised learning model, introducing a probability pseudo-mixing mechanism for interpolating unlabeled samples and their pseudo-labels, enhancing the regularization effect.

2.4 Pretraining and Fine-Tuning

In 2018, Howard et al. introduced Universal Language Model Fine-tuning (ULMFiT) [20], incorporating key techniques for fine-tuning language models. In 2020, Ting et al.'s SimCLR [21] model demonstrated a significant improvement in accuracy when fine-tuning on only 1% of labels. Subsequent research utilized SimCLRv2 [22] for unsupervised pre-training of a large ResNet model, followed by supervised fine-tuning on a small set of labeled examples. Knowledge from unlabeled examples was distinguished to enhance and transfer task-specific knowledge.

3 Method

3.1 Fusion of Training Pipeline

The paradigm shift in the training pipeline has made significant strides in improving model performance in recent years. For instance, in the FixMatch framework, the pipeline has been altered to first undergo unsupervised pre-training followed by self-supervised

training fine-tuning. Similarly, in the SimCLRv2 framework, the approach involves initial unsupervised pre-training followed by supervised fine-tuning, ultimately employing knowledge distillation and transfer from unlabeled samples. In the training process of this study, following experimentation and exploration, a methodology akin to SimCLRv2 was adopted. Specifically, the decision was made to first conduct unsupervised pre-training on the entire dataset, then perform supervised fine-tuning on a subset of labeled data, and finally engage in semi-supervised training on both labeled and unlabeled data. Within the semi-supervised training framework, Probability MixText (PMT) was employed, incorporating probabilistic pseudo-mixing and GANs-based data augmentation techniques into the foundational MixText framework.

3.2 Probabilistic Pseudo Mixup

For the proposed data augmentation technique Mixup applied to image data, linear interpolation is performed at the pixel level of the input images. Specifically, it involves blending the pixel values of the original images in a certain proportion. Correspondingly, the labels are mixed in the same ratio, resulting in new samples and labels with blended features. The mixing ratio λ is derived from a Beta distribution, typically involving the random selection of two different samples, x_q and x_p, along with their corresponding labels from the dataset $\mathrm{X} = \{x_1, ..., x_m\}$ and labels $\mathrm{Y} = \{y_1, ..., y_m\}$. Formally, for $p, q \in [0, m]$, the mixing process is defined as follows:

$$\tilde{\mathrm{x}} = \lambda x_q + (1 - \lambda) x_p, \tag{1}$$

$$\tilde{\mathrm{y}} = \lambda y_q + (1 - \lambda) y_p. \tag{2}$$

However, due to the varying qualities of data and pseudo-labels generated, the simple random selection of the mixing ratio λ from the Beta distribution for poorly performing samples might lead to an undesired impact. This randomness could potentially allow low-quality data to influence high-quality data and affect loss calculations. To address this issue, the concept of probabilistic pseudo-mixing [19] is introduced.

Despite the lower quality of the data, it still holds valuable information. Probabilistic pseudo-mixing continues to involve random mixing of unlabeled data, but the mixing ratio λ is no longer randomly generated from a Beta distribution. During the semi-supervised training phase, pseudo-labels and corresponding losses are generated. The confidence is determined based on the loss information from the two samples involved in the mixing. Through this mechanism, samples with higher confidence have a higher proportion in the mixed samples, and consequently, the pseudo-labels have a higher proportion. This weighting allows higher confidence samples to contribute more significantly to the semi-supervised loss calculation. This mixing strategy enhances regularization and provides greater flexibility.

3.3 Data Augmentation

To address the high imbalance and insufficient scale of traffic incident data, in the data augmentation module, the decision was made to employ Generative Adversarial

Networks (GANs) to tackle these challenges. GANs are a common data augmentation technique capable of simulating the distribution of input data and capturing latent information to generate highly similar new data. Typically, GAN models consist of a generator and a discriminator, aiming to train the generator in such a way that the discriminator maximizes the probability of erroneously classifying generated samples as real samples. Previous research [23] has demonstrated that GANs produce high-quality data and have significantly contributed to advancements in various research domains.

3.4 MixText

The Probability Pseudo-Mixing (Tmix) technique was originally designed for image data and may not be directly applicable to text data due to its discrete nature. Therefore, interpolation is performed in the hidden space. In a BERT model with H layers, the process involves selecting x and x' from the dataset and inputting them into the first layer to obtain their hidden representations, denoted as h and h', respectively. Then, at an intermediate layer, denoted as layer E, the hidden representations h_E and h'_E of these two samples are mixed using the Mixup operation, generating a new sample h_m based on a random number λ drawn from a Beta distribution for each batch. The formula for obtaining the new sample h_m is as follows:

$$h_m = \lambda h_E + (1 - \lambda) h'_E, \tag{3}$$

simultaneously, the samples x and x' are input into the BERT model to obtain their corresponding pseudo-labels y and y', respectively. With the previously generated mixing ratio λ, the pseudo-label for the new sample is calculated as follows:

$$y_m = \lambda y + (1 - \lambda) y'. \tag{4}$$

This constitutes the framework of Tmix. Extending from this, MixText incorporates both labeled dataset X_L and unlabeled dataset X_U into the model training process. For labeled data, the model is trained using the supervised loss function, which is the cross-entropy loss:

$$L_x = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} log(p_{i,j}), \tag{5}$$

where, N denotes the batch size, C represents the number of classes, $y_{i,j}$ signifies the j-th element in the true label of sample i, and $p_{i,j}$ is indicative of the predicted probability by the model for the j-th class of sample i.

For unlabeled samples, the Kullback-Leibler Divergence Loss (KL Divergence Loss) is employed as the consistency loss. This helps ensure that the model produces similar outputs for similar inputs, enhancing the model's consistency L_u. The loss calculation formula is as follows:

$$L_u = D_{KL}(softmax(outputs_u) \| targets_u), \tag{6}$$

where $outputs_u$ is the model's output for unlabeled data, and $targets_u$ is the pseudo-label calculated based on the predicted probabilities $p_{i,j}$.

For mixed data, when both mixed samples come from the labeled dataset, the model is trained using the supervised loss. When both mixed samples come from the unlabeled dataset, the model is trained using KL Divergence Loss. When one mixed sample comes from the labeled dataset and the other from the unlabeled dataset, the model is trained using both the supervised loss and KL Divergence Loss. The formula is as follows:

$$L = L_x + w \cdot L_u, \tag{7}$$

where w is a weight used to balance the contributions of labeled and unlabeled samples.

The training process involves iterative mixing of labeled and unlabeled data with a certain probability, calculating the corresponding losses in each iteration.

3.5 FPMT

Building upon the MixText framework, the new semi-supervised traffic incident detection model, named FPMT, integrates the training strategy as a Fusion of Training Pipeline, the mixing strategy as a Probabilistic Pseudo Mixup, and the data balancing and augmentation strategy as GANs.

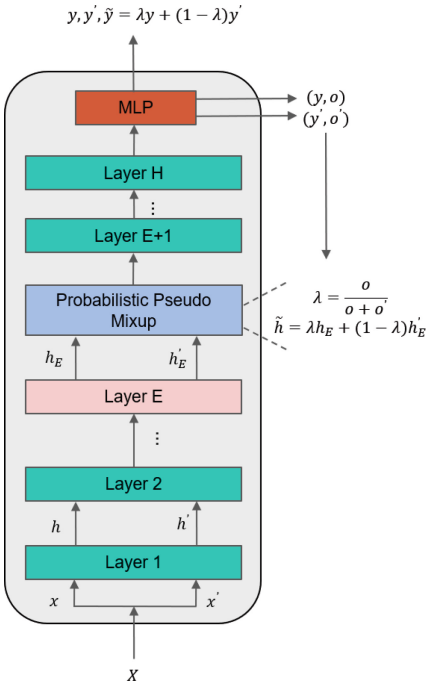


Fig. 1. Architecture of PTmix.

The framework of the Probability Pseudo Mixup model (PTmix) within FPMT is illustrated in Fig. 1. Samples x and x' are input into PTmix, and the output layer provides

their pseudo-labels and confidence scores (y, o) and (y', o'). The mixing ratio λ is determined based on the confidence proportions using the Formula 8. Subsequently, based on the values of λ and $1 - \lambda$, samples x and x' are mixed at the layer E of the model, producing hidden representations h_E and h'_E for the new sample, as well as mixing the pseudo-labels y and y' for the unlabeled data, resulting in the new sample $\tilde{h}$ and $\tilde{y}$.

$$\lambda = \frac{o}{o + o'}, \tag{8}$$

$$\tilde{\mathrm{h}} = \lambda h_E + (1 - \lambda) h'_E, \tag{9}$$

$$\tilde{\mathrm{y}} = \lambda y + (1 - \lambda) y'. \tag{10}$$

The semi-supervised fine-tuning stage of the FPMT model follows the framework of the PTmix model, as depicted in Fig. 2. Initially, the dataset X_O undergoes data augmentation using GANs to balance and expand the dataset. The augmented dataset is then partitioned into the labeled dataset X_L and the unlabeled dataset X_U. Both X_L and X_U are fed into PTmix, generating predicted labels Y_L for X_L, predicted labels Y_U for X_U, and predicted labels Y_M for mixed dats $\mathrm{X_M}$. The mixing strategy employed is probabilistic pseudo-mixing. Different loss functions are applied for calculating losses on different types of data, following the strategy outlined in MixText, as described in Sect. 3.4.

Additionally, the FPMT model adopts a training strategy involving initial unsupervised training on all data, followed by supervised fine-tuning and ultimately semi-supervised fine-tuning.

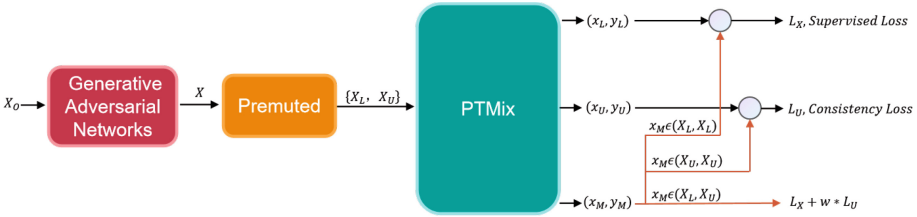


Fig. 2. Overall architecture of the proposed FPMT model.

4 Experiment

4.1 Datasets

To evaluate the proposed semi-supervised traffic incident detection model, four real-world datasets were utilized. These datasets include PeMS [24], I-880 [25], Whitemud Drive [26], and NGSIM [27]. PeMS is a California-specific traffic flow database, that collects real-time data from over 39,000 independent detectors. It includes parameters of

traffic flow, incident data, and weather information. I-880 originates from the renowned I880 highway traffic incident database in the United States, documenting traffic flow speed, occupancy data, and incident information on a 9.2-mile stretch of the highway. Whitemud Drive is a 28-km-long highway in Edmonton, Alberta, Canada, equipped with circular detectors on main lanes and ramps to gather traffic parameters. NGSIM, initiated by the United States Federal Highway Administration, gathers real-time vehicle trajectory data for driving behavior analysis, traffic flow analysis, microsimulation modeling, and vehicle trajectory prediction. These datasets provide valuable information for traffic flow prediction, model analysis, and urban traffic planning and management.

4.2 Comparing Method

To validate the effectiveness of FPMT, it was compared with several recent models during the experimental phase. Among these, BERT [28] is a bidirectional encoder representation model that achieves significant performance improvements across various natural language processing tasks by jointly pretraining on the left and right context of the text, without requiring extensive task-specific architecture modifications during fine-tuning. VAT [4] is a regularization method based on virtual adversarial loss, achieving high performance in semi-supervised learning tasks by measuring the local smoothness of the input conditional label distribution. UDA [29] is a new approach in semi-supervised learning that employs advanced data augmentation methods such as RandAugment and back-translation, replacing simple noise operations and significantly improving performance across six languages and three visual tasks. DSP [30], by guiding the teacher to generate more accurate pseudo-labels through student feedback and combining consistency regularization, significantly improves text classification performance.

4.3 Experimental Setting

As traffic incident detection is a binary classification task, the class parameter is fixed at 2. For the layer selection of probabilistic pseudo-mix-up in PMT, it was observed that the mixing performed better at the 9th layer after training PMT separately. The model's decoder is based on Bert-base-uncased, and the output is classified through an additional linear layer. The learning rate for the BERT model's encoder is set to 0.00001, and the learning rate for the additional linear layer is set to 0.001. During the semi-supervised fine-tuning phase, for each dataset, GANs are utilized to balance and augment the dataset. For the augmented dataset, in each category, the number of unlabeled samples is set to 5000, while the number of labeled samples is set to 50, 100, and 1500, achieving label rates of 1%, 2%, and 30%, respectively.

4.4 Result

In the experiments, the selected evaluation metrics include Classification Rate (CR), Detection Rate (DR), and F1-score. After fixing the number of unlabeled samples for each category at 5000, experiments were conducted on four real datasets with varying

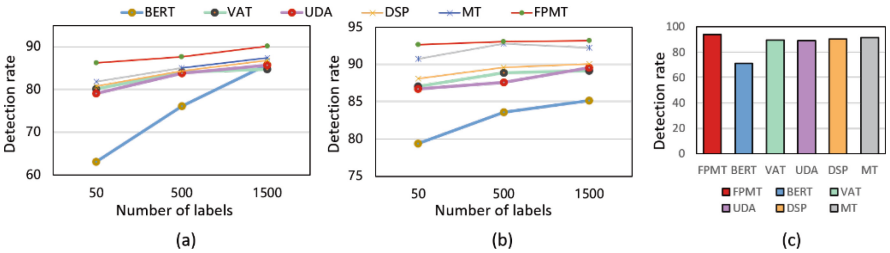


Fig. 3. (a) and (b) compare our FPMT model with baselines using different numbers of labeled samples (50, 100, 1500). Meanwhile, (c) represents the comparison with a fixed number of labeled samples set at 50.

numbers of labeled samples (50, 100, and 1500 for each category), and the results are presented in Table 1 and Fig. 3. Our FPMT model achieved superior performance compared to the contrasted models, demonstrating excellent performance even when the number of labeled samples is minimal. Particularly noteworthy is its performance on DR, where, even at a label rate of only 1%, it outperforms BERT at a 30% label rate. The model exhibits remarkable capability in enhancing DR. On the PeMS dataset, when the number of labeled data is the smallest, only 50, the proposed FPMT model achieves a detection rate 4.4% higher than MixText, demonstrating the best performance. On the I-880 dataset, the detection rate is 5.6% higher than MixText. The improvement on the other two datasets is not as significant, but the model still exhibits the best performance.

Table 1. Performance (Classification Rate (CR) (%), Detection Rate (DR) (%), and F1-score (%)) comparison with baselines. Models are trained with 50, 100, and 1500 labeled data per class.

Dataset	Model	50	500	1500
PeMS	BERT	71.3/63.1/70.6	83.8/76.1/81.3	89.3/85.8/89.6
	VAT	89.3/80.1/89.2	91.9/84.1/90.4	92.3/84.8/90.9
	UDA	88.9/79.1/87.4	91.9/83.8/90.1	92.4/85.7/90.7
	DSP	90.2/80.8/89.9	92.1/84.4/90.9	92.9/86.8/91.3
	MT	91.6/81.9/90.3	92.7/85.1/91.7	93.1/87.4/92.3
	FPMT	**93.7/86.3/91.7**	**94.3/87.7/92.8**	**95.5/90.2/94.7**
I-880	BERT	70.7/64.1/69.6	82.8/77.1/81.8	88.2/86.8/87.8
	VAT	88.7/82.1/87.7	90.4/86.5/87.9	91.8/87.1/90.1
	UDA	88.9/83.8/87.4	90.9/85.2/89.8	91.9/87.5/91.2
	DSP	89.5/82.5/89.2	91.4/86.3/91.2	92.6/88.4/91.4
	MT	90.9/82.7/89.9	91.9/87.3/91.6	92.9/89.3/92.1
	FPMT	**93.2/88.3/92.7**	**93.7/89.4/92.9**	**94.9/92.1/94.9**

(*continued*)

Table 1. (*continued*)

Dataset	Model	50	500	1500
Whitemud Drive	BERT	84.8/79.4/82.9	88.1/83.6/86.8	90.7/85.2/89.9
	VAT	92.7/87.1/90.4	94.5/88.9/92.7	95.4/89.2/93.1
	UDA	92.4/86.7/90.1	93.9/87.6/91.9	95.8/89.6/93.7
	DSP	93.8/88.1/92.3	94.7/89.6/93.4	95.9/90.1/94.2
	MT	96.7/90.8/94.9	98.1/92.8/97.9	98.7/92.3/98.2
	FPMT	**97.3/92.7/95.6**	**98.2/93.1/97.3**	**98.7/93.2/98.4**
NGSIM	BERT	80.8/76.4/82.9	85.4/83.6/86.8	85.9/83.7/87.1
	VAT	89.6/83.4/87.3	90.8/85.4/89.9	91.3/86.1/90.7
	UDA	88.4/82.8/86.5	90.5/83.9/88.4	91.7/85.3/89.8
	DSP	90.5/84.3/88.5	91.1/86.2/89.9	92.3/87.4/90.7
	MT	93.1/87.3/91.2	94.7/89.4/93.9	95.4/89.9/94.3
	FPMT	**94.8/90.4/92.2**	**95.9/91.3/93.5**	**96.8/92.3/94.7**

4.5 Ablation Experiments

The experimental results comparing the model PMT of FPMT during the semi-supervised fine-tuning stage with MixText on dataset PeMS and dataset Whitemud Drive are presented in Table 2. It can be observed that, with the improved mixing strategy, the model's performance across various aspects has been enhanced.

Table 2. Performance comparison with MixText and PMT.

Dataset	Model	50	500	1500
PeMS	MT	91.6/81.9/90.3	92.7/85.1/91.7	93.1/87.4/92.3
	PMT	92.1/83.2/90.9	93.9/86.2/92.3	94.3/88.5/93.4
	FPMT	**93.7/86.3/91.7**	**94.3/87.7/92.8**	**95.5/90.2/94.7**
Dataset	Model	50	500	1500
Whitemud Drive	MT	96.7/90.8/94.9	98.1/92.8/97.9	98.7/92.3/98.2
	PMT	**97.6**/91.3/93.5	**98.3**/92.9/**97.4**	**98.9**/92.7/98.3
	FPMT	97.3/**92.7**/**95.6**	98.2/**93.1**/97.3	98.7/**93.2**/**98.4**

5 Conclusion

In the research field of traffic incident detection, methods based on traffic data have made significant progress. However, popular deep-learning approaches heavily rely on data collection and labeling. To alleviate the re-source-intensive nature of data labeling,

this paper proposes a semi-supervised learning traffic incident detection model, FPMT, reducing the model's dependence on labeled data. The training pipeline involves pre-training in an unsupervised manner, followed by supervised fine-tuning, and ultimately semi-supervised training. The model incorporates GANs for balancing and augmenting the dataset and utilizes a data augmentation technique, probabilistic pseudo-mixing, at hidden layers to enhance the performance of the semi-supervised model. Comparative experiments with recent models on four real datasets demonstrate the effectiveness of the proposed model. The results show that the model achieves high performance even in scenarios with limited labeled data. As a future research direction, we plan to explore deep semi-supervised learning for traffic incident detection in open environments, simultaneously handling data from different domains or modalities, and leveraging unlabeled data for learning in situations with limited labeled data.

Acknowledgments. This work is supported by China NSFC Program under Grant NO. 61603257.

References

1. Xie, T., Shang, Q., Yu, Y.: Automated traffic incident detection: coping with imbalanced and small datasets. IEEE Access **10**, 35521–35540 (2022)
2. Eldele, E., et al.: Time-series representation learning via temporal and contextual contrasting. arXiv preprint arXiv:2106.14112 (2021)
3. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. Adv. Neural Inf. Process. Syst. **30** (2017)
4. Miyato, T., et al.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1979–1993. IEEE (2018)
5. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
6. Berthelot, D., et al.: Mixmatch: a holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
7. Sohn, K., et al.: Fixmatch: simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems, vol. 33, pp. 596–608 (2020)
8. Chen, J., Yang, Z., Yang, D.: Mixtext: linguistically-informed interpolation of hidden space for semi-supervised text classification. arXiv preprint arXiv:2004.12239 (2020)
9. Under, I.D.A.: On-line testing of the McMaster incident detection algorithm under recurrent congestion. In: Freeway Operations and High-Occupancy Vehicle Systems, p. 1
10. Samant, A., Adeli, H.: Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis. In: Computer-Aided Civil and Infrastructure Engineering, vol. 15, no. 4, pp. 241–250. Wiley Online Library (2000)
11. Yijing, H., et al.: Intelligent algorithms for incident detection and management in smart transportation systems. In: Computers and Electrical Engineering, vol. 110, p. 108839. Elsevier (2023)
12. Xiao, J., Liu, Y.: Traffic incident detection using multiple-kernel support vector machine. In: Transportation Research Record, vol. 2324, no. 1, pp. 44–52. SAGE Publications, Sage (2012)

13. Liu, Q., Lu, J., Chen, S.: Traffic incident detection using random forest (2013)
14. AlDhanhani, A., et al.: Multi-modal traffic event detection using shapelets. In: Neural Computing and Applications, vol. 35, no. 2, pp. 1395–1408. Springer (2023)
15. Han, X.: Traffic incident detection: a deep learning framework. In: 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE (2019)
16. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems, vol. 17 (2004)
17. Lee, D.-H.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, no. 2, p. 896. Atlanta (2013)
18. Wang, Q., Li, W., Gool, L.V.: Semi-supervised learning by augmented distribution alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1466–1475 (2019)
19. Cai, Z., et al.: Semi-supervised vision transformers at scale. In: Advances in Neural Information Processing Systems, vol. 35, pp. 25697–25710 (2022)
20. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
21. Chen, T., et al.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, PMLR, pp. 1597–1607 (2020)
22. Chen, T., et al.: Big self-supervised models are strong semi-supervised learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 22243–22255 (2020)
23. Yang, H., Zhou, Y.: Ida-gan: a novel imbalanced data augmentation gan. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 8299–8305. IEEE (2021)
24. Jiang, R., et al.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 7, pp. 8078–8086 (2023)
25. Li, H., et al.: Spatial and temporal prediction of secondary crashes combining stacked sparse auto-encoder and long short-term memory. In: Accident Analysis & Prevention, vol. 191, p. 107205 (2023)
26. Heidari, H., Zeng, B.: An optimized grey transition Verhulst method. In: Engineering Applications of Artificial Intelligence, vol. 120, p. 105870 (2023)
27. Gao, K., et al.: Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment. In: IEEE Transactions on Intelligent Transportation Systems (2023)
28. Devlin, J., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
29. Xie, Q., et al.: Unsupervised data augmentation for consistency training. In: Advances in Neural Information Processing Systems, vol. 33, pp. 6256–6268 (2020)
30. Li, S., et al.: Dual pseudo supervision for semi-supervised text classification with a reliable teacher. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2513–2518 (2022)

C2F-CHART: A Curriculum Learning Approach to Chart Classification

Nour Shaheen[1,2(✉)], Tamer Elsharnouby[2], and Marwan Torki[1,2]

[1] Faculty of Engineering, Alexandria University, Alexandria, Egypt
mtorki@alexu.edu.eg
[2] Applied Innovation Center, MCIT, Giza, Egypt
{nour.hesham,tamer.elsharnouby}@aic.gov.eg

Abstract. In scientific research, charts are usually the primary method for visually representing data. However, the accessibility of charts remains a significant concern. In an effort to improve chart understanding pipelines, we focus on optimizing the chart classification component. We leverage curriculum learning, which is inspired by the human learning process. In this paper, we introduce a novel training approach for chart classification that utilizes coarse-to-fine curriculum learning. Our approach, which we name C2F-CHART (for coarse-to-fine) exploits inter-class similarities to create learning tasks of varying difficulty levels. We benchmark our method on the ICPR 2022 CHART-Infographics UB UNITEC PMC dataset, outperforming the state-of-the-art results.

Keywords: Chart Classification · Curriculum Learning · Chart Understanding

1 Introduction

Charts are commonly used to represent features and relationships in data. They are also regularly used in scientific research. In the areas of machine learning, a researcher has to interpret loss curves, confusion matrices, data analysis plots, feature importance plots, and others. However, when dealing with visual representations, there is always the risk that individuals with vision impairment, low vision, or blindness are at a disadvantage. To increase the accessibility of charts, which are inherently visual, automatic pipelines for chart data extraction are needed. The chart data extraction process is called chart mining [6]. Often, the first step in this process is high-level chart classification. The division of chart images into specific categories can simplify further processing steps in the pipeline. This initial categorization can allow the following steps to either leverage the chart type, as meta-information about the image, or to assign different processing methods for each type instead of using the same method arbitrarily for all charts.

Past research [1,8,20] has investigated the use of deep learning methods for image classification and contrasted them to achieve a robust, highly accurate

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 375–391, 2025.
https://doi.org/10.1007/978-3-031-78107-0_24

chart classifier. Most of the research has been directed toward identifying the best model architecture for the task, whether a convolutional neural network (CNN) architecture or a transformer architecture.

Inspired by how humans learn, we use a curriculum learning (CL) [4] based approach. We extend the coarse-to-fine CL algorithm [18], which focuses mainly on classification tasks. Stretcu et al. [18] argue that, during classification, we can attribute the model's errors to similarities in class labels. Our motivation to use this approach as a building block is because one of the existing challenges in chart classification resides in the similarities between the different chart classes, or the *output space* of the model.

This inter-class similarity, usually considered a challenge, is leveraged by our training approach. Our CL setting allows us to construct learning tasks that are guaranteed to vary in difficulty by grouping similar classes. This allows us to construct simpler tasks, where we classify broader categories, and then to construct more complex ones, where we focus on distinguishing between specific types and classes. We can visualize this in Fig. 1. We refine the learning process across multiple levels of complexity and then combine the experience of multiple different learners at each level. We show that this optimizes the model's ability to discern nuanced differences in chart features that each learner might have picked up independently.

The contributions of this paper are as follows:

- We developed a novel training approach that, to the best of our knowledge, has not been used in chart classification before. We then used our approach to train the current state-of-the-art model architecture *Swin-Chart* [8].
- We ran an evaluative analysis to confirm that our approach outperforms the SOTA architecture on the ICPR 2022 CHART-Infographics UB Unitec PMC Dataset [7]. It also outperforms the ICPR 2022 CHART-Infographics competition winner on the same dataset. Our analysis also proved that our method exceeded the results of traditional coarse-to-fine CL.

The structure of the paper is as follows. In Sect. 2, we provide a concise overview of the past research conducted in chart classification using deep learning methods as well as curriculum learning techniques for image classification. Section 3 describes the dataset we used for benchmarking our results and why we selected it. In Sect. 4, we detail the method we developed. Our results, experiments, and comparison with other methods are provided in Sect. 5. Lastly, Sect. 6 concludes the paper and presents directions for future study.

2 Related Work

2.1 Deep Learning Methods for Chart Classification

This section focuses on previous work investigating the use of various deep learning methods for chart classification. We acknowledge that the use of both model-based methods and traditional machine learning methods for chart classification has been thoroughly investigated [1,12,20], yet each approach has its challenges.

Unlike a model-based approach, which models only a predefined set of chart types, deep learning methods are agnostic to the specific types encountered. They're also superior to traditional machine learning methods that require handcrafted features. Also, these methods aren't guaranteed to generalize well due to the varieties in chart image datasets.

Regarding deep learning methods, convolutional neural networks (CNNs) [13] have long been a staple of image classification in different domains. The ability of CNNs to capture hierarchical features from images through convolutional filters has allowed them to be extremely effective in feature extraction.

Liu et al. [14] introduced DeepChart, which combined CNNs for feature extraction and Deep Belief Networks (DBNs) for classification. Amara et al. [1] used a vanilla CNN-based model, inspired by the LeNet architecture [13] and tested it on their own dataset which is composed of 11 categories. Many subsequent papers [2,20] compare other CNN architectures such as different layer versions of ResNets [10], DenseNets [11], VGG Networks [16], Xception Modules [5], and EfficientNet [19].

In 2021, Bajic et al. [3] introduced a new addition to the CNN method of classifying charts: a Siamese CNN. They argue that when small datasets are used, a Siamese CNN outperforms a classic CNN in both classification accuracy and F1-score.

Finally, Dhote et al. [8] compared the use of several CNN architectures for this task on the same testing dataset. They compared and contrasted the ResNet-152, the Xception module, the DenseNet-121, and ConvNeXt [15], concluding that the Resnet-152 achieved the highest performance on the ICPR 2022 UB UNITEC PMC testing dataset out of all the other CNN architectures.

However, with the advent of image transformer models [9], it's fair to say CNNs have been surpassed in performance. Vision transformers treat images as sequences of patches. Instead of the localized feature maps produced by CNNs, transformer models leverage the self-attention mechanism to capture global dependencies. Dhote et al. [8] compared two transformers backbones: Swin-based and DeIT-based, and concluded that Swin transformers with different patch sizes outperform CNN-based architectures. Their state-of-the-art chart classification model, Swin-Chart, was, to our knowledge and previous to this work, the best performing transformer model architecture for chart classification on the aforementioned dataset.

2.2 Curriculum Learning

Curriculum learning (CL) was first introduced by Bengio et al. [4]. The intuition for it stemmed from the methods used by humans to learn information. Around the world, humans start by learning easier concepts before gradually moving towards more complex concepts later. Usually, the input data used to train machine learning models is not organized in any meaningful way. The samples are instead fed to the model in a random order, with easy and difficult samples shuffled and presented to the model with no heed to its training status or the difficulty of each data point.

Soviany et al. [17] propose in their survey of CL methods that increasing the complexity of the data, referred to as the *experience* of the model, is not the sole approach to implement curriculum learning. They contend that complicating any other machine learning component will produce a more involved objective function. Namely, this might be done by increasing the complexity of the model itself, by adding or activating neural units for example, or by increasing the complexity of the class of tasks the model is being trained on.

Coarse-to-fine curriculum learning [18], which is the main building block of our method, keeps the experience of the model consistent during training, but instead leverages the similarity between data classes to define a set of tasks $\{f_0, f_1, ...\}$ that are guaranteed to vary in difficulty.

The inspiration behind this coarse-to-fine technique originated from a specific aspect of human learning, where humans learn to break down specific, detailed tasks into simple milestones. Stretcu et al. [18] illustrate this method with the analogy of a child initially learning to identify dogs broadly as dogs, before later learning to differentiate between different dog breeds. Instead of relying on varying difficulty levels in the input data, the method itself ensures variability by progressively introducing tasks of increasing complexity as the model continues learning. This is achieved by clustering similar classes into broader categories, creating a hierarchical structure of class labels. Each task assigned to the model corresponds to a level within the hierarchy, with simpler tasks being at the top, where the categories are less specific.

3 Dataset

For training and testing, we used the datasets provided for the ICPR 2022 CHART-Infographics competition [7]. Both the training and testing sets are comprised of real charts from the PubMed Central that have been manually annotated. The testing set for the chart classification task, called *Split 1* in the data and the competition paper, is composed of 11,388 samples while the training set is composed of 22,923 samples. Table 1 describes the frequency of each class in both sets. We reserved 10% of the dataset's training split for validation purposes and only used 90% throughout our training process.

Another important motivator for selecting this dataset was the desire to conduct a comparative analysis with other research and quantify the improvements made. Since this dataset was used in the CHART-Info competition, we have real results of different deep learning methods to compare against.

4 Method

In this section, we present our hierarchical coarse-to-fine CL approach, leveraging a Swin Transformer model for chart classification. We previously described how traditional CL and coarse-to-fine CL differ in how they consider difficulty. Coarse-to-fine CL's main idea is to create tasks with increasing difficulty

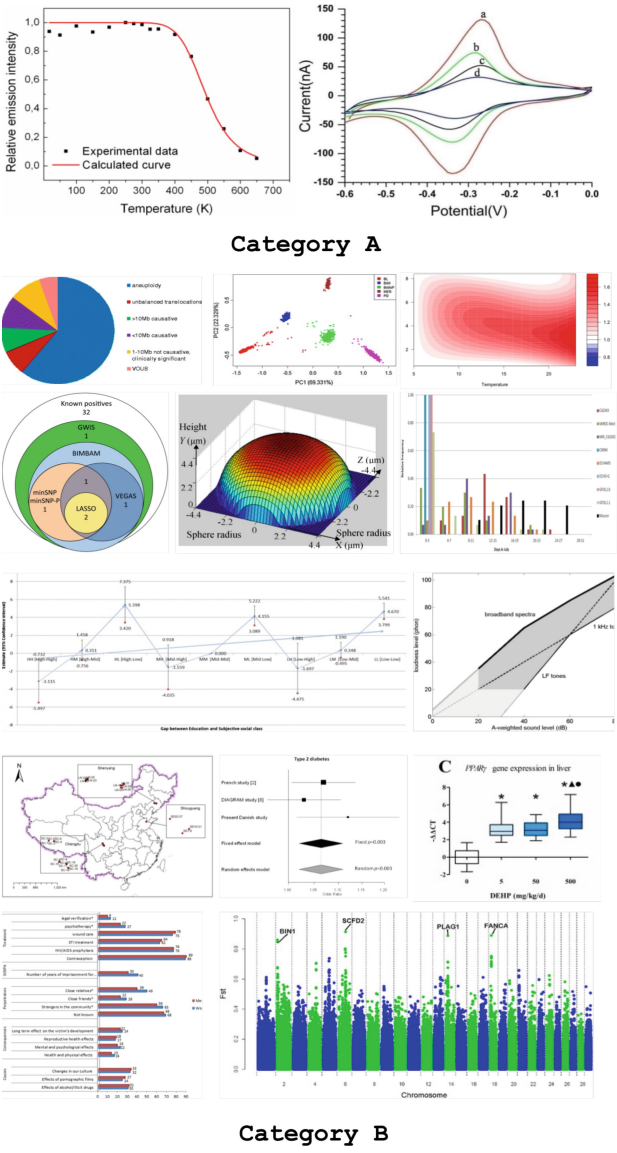


Fig. 1. Visualization of chart type clustering to construct tasks of varying difficulty.

$\{f_0, f_1, ...\}$ while not changing the order of the input data. However, our approach, which is described by Fig. 2, goes a step further and considers at which point in the learning process we should start teaching the model the more complex task. We also consider knowledge sharing between learners who shifted to the complex task at different points in that process. When the model is learning the simpler task f_0, the traditional approach would have us transferring

Table 1. Frequency of each chart type in the ICPR 2022 UB PMC Dataset

Chart Type	Train	Test
Area	172	136
Bar (horizontal)	787	425
Bar (vertical)	5,454	3,183
Box (vertical)	763	596
Heatmap	197	180
Interval (horizontal)	156	430
Interval (vertical)	489	182
Line	10,556	2,776
Manhattan	176	80
Map	533	373
Pie	242	191
Scatter	1,350	949
Scatter-line	1,818	1,628
Surface	155	128
Venn	75	131
Total	22,923	11,388

knowledge to the more complex task f_1 after a certain time (number of epochs) or when it performs best on f_0 (checkpoint with the highest validation score). Instead, we argue that choosing either the best performing model on f_0 or the final model after several epochs does not necessarily produce optimal results.

Instead, we choose to transfer knowledge from the top-K learners of f_0 so that each learner is then trained on f_1, producing K training paths. We then choose the top learner of each path of f_1, producing K final checkpoints. We argue that the subsequent sharing of the knowledge obtained by these K final checkpoints produces better results. Knowledge sharing here happens during inference time.

Our method involves three main steps: clustering, training/fine-tuning, and ensembling. We use minimizing cross-entropy (CE) loss as an objective function. We provide the detailed pseudocode of these three steps in Algorithm 1.

Step 1: Cluster To obtain a hierarchy of simple-to-complex tasks, we needed to first cluster classes based on similarity. We started by training a vanilla classification model using the current state-of-the-art architecture which, as described in [8], is a Swin Transformer pre-trained on the ImageNet dataset with an input size of 224 that they call Swin-Chart. We use this model to compute a coarse-to-fine class hierarchy H shown in Fig. 3. H depends on the similarity between the columns of the projection matrix in the output layer or *the predictor* of the Swin model. It is computed through affinity clustering, using the pairwise cosine

Algorithm 1. Hierarchical Coarse-to-Fine CL with Swin-Chart

Input: Swin Transformer θ with input image dimension 224 (SwinL_224)
Compute class hierarchy H using an auxiliary clustering function
Define auxiliary objective functions f_0 and f_1:
  f_0: Minimize CE loss for the 2 classes in level 1 of H
  f_1: Minimize CE loss for the 15 classes in level 2 of H
Initialize θ^0 with SwinL_224 weights pre-trained on ImageNet
for epoch = 1 to 100 **do**
  Train θ^0 on f_0
  Validate on holdout set
end for
Select top 5 checkpoints $\{\theta_0^0, \theta_1^0, \theta_2^0, \theta_3^0, \theta_4^0\}$ based on val. F1-scores
for each θ_i^0 **do**
  Initialize θ_i^1 with encoder parameters from θ_i^0 and random decoder parameters
  for epoch = 1 to 100 **do**
    Train θ_i^1 on f_1
    Validate on holdout set
  end for
end for
Add top checkpoint from each θ_i^1 to final model pool to combine $\{\theta_0^1, \theta_1^1, \theta_2^1, \theta_3^1, \theta_4^1\}$
Conduct combinatorial search to find optimal model combination
Output: Model combination with max. F1-score on holdout set

distances between the columns as a distance matrix [18]. The pseudocode for computing H and generating hierarchical clusters is provided in Algorithm 2.

Step 2: Train, Divide, and Fine-Tune. Using our two-level class hierarchy, we constructed two auxiliary tasks f_0 and f_1, one for each level. We define the objective function L_i used to train the i-th auxiliary task f_i as optimizing the maximum likelihood by minimizing cross-entropy (CE) loss, which is equivalent to minimizing the negative log-likelihood:

$$L_i = -\sum_j \log \left(\sum_{c \in C_i(y_j)} \exp\{f_i(x_j)\} \right) \tag{1}$$

where i denotes the hierarchy level we're working at. For each sample j, x_j and y_j represent the input data and its corresponding true label. The term $C_i(y_j)$ denotes the cluster in level i that the class y_j belongs to.

We started by initializing a new instance of the Swin-Chart architecture and training it on f_0 to produce top-K checkpoints from different points of the training journey, judged based on the average per-class F1-measure of a hold-out validation set. We chose K = 5, and so we obtained 5 level-1 models [θ_0^0 - θ_4^0] in order of validation scores. The output space of each θ_i^0 is simply the two main clusters in level 1, as shown in Fig. 3.

For each θ_i^0, we used the staged coarse-to-fine CL algorithm [18]: we initialized five Swin-Chart level-2 models [θ_0^1 - θ_4^1], whose encoder parameters were set as the

Algorithm 2. Generate Clusters Per Level for Hierarchical Coarse-to-Fine CL

Input: Number of classes K, training data, pre-trained baseline Swin model θ
Train Swin model θ on the training data
Extract the class embeddings from the final layer of θ
Let $W \in R^{E \times K}$ be the weight matrix of the final layer
Compute the distance matrix D using cosine distances between class embeddings:
for $k_1 = 1$ to K **do**
  for $k_2 = k_1 + 1$ to K **do**
    Compute cosine distance $d(k_1, k_2) = 1 - \cos(W_{\cdot k_1}, W_{\cdot k_2})$
    Update $D[k_1, k_2] \leftarrow d(k_1, k_2)$, $D[k_2, k_1] \leftarrow d(k_1, k_2)$
  end for
end for
Apply affinity clustering on the distance matrix D to form the hierarchy H
Initialize clustersPerLevel $\leftarrow$ []
for $l = 1$ to depth(H) **do**
  Initialize clustersPerLevel[l] $\leftarrow$ []
  for each node $n \in H$.nodesAtDepth(l) **do**
    Create cluster c by grouping leaves of the sub-tree rooted at n
    Append c to clustersPerLevel[l]
  end for
end for
Output: Class hierarchy H, clusters per level clustersPerLevel

encoder parameters of the corresponding θ_i^0 and whose predictor parameters were randomly initialized. We then fine-tuned all parameters of our level-2 models on the desired output space, the 15 classes in level 2, as shown in Fig. 3.

After the second fine-tuning step, we chose the checkpoint with the maximum validation score from each level-2 model as an ingredient for the final combination step, totaling K final models, judged based on the average per-class F1-measure.

Step 3: Combine. To combine our K ingredients, we considered two approaches: an ensembling method through averaging of predictions, and the model soups method [22]. For each method, we conducted a combinatorial search to choose the optimal model combination to ensemble or soup. We judged all combinations in both methods on their validation scores and chose our final model to be the model combination producing the maximum F1-score on the hold-out validation set.

5 Experiments

5.1 Setup

As mentioned before, we used the datasets provided for the ICPR 2022 CHART-Infographics competition for both training and testing. We benchmarked our results on the testing dataset, called ICPR 2022 UB Unitec PMC Dataset, and

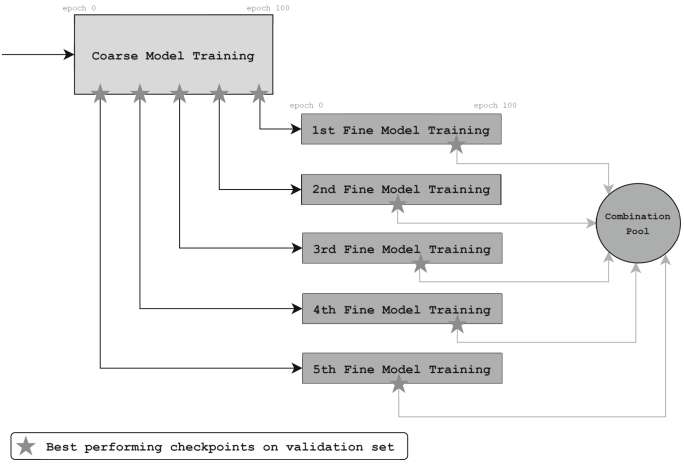


Fig. 2. Overview of our curriculum learning method.

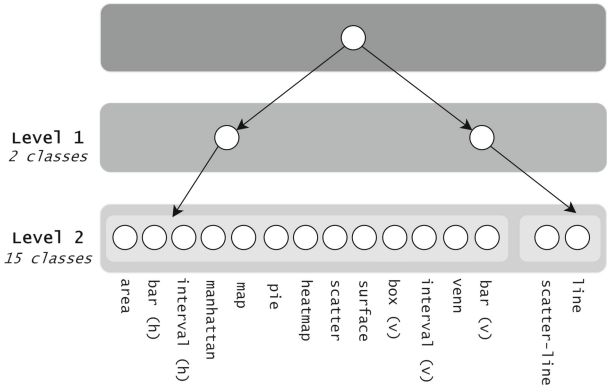


Fig. 3. Automatically computed coarse-to-fine class hierarchy of 2 levels on the 15 classes of the ICPR 2022 CHART-Infographics UB Unitec PMC Dataset.

compared them with previous work. Throughout the following experiments, we designated 10% of the dataset as a hold-out validation set, and used the remaining 90% in our clustering and fine-tuning steps.

In order to avoid attributing our improvements to a change in hyperparameters, we used an identical experimental setup as the one mentioned in [8]. This choice guarantees consistency and allows for a fair comparison between our approach and theirs.

To obtain the class hierarchy H in Fig. 3, we trained a Swin transformer, pre-trained on ImageNet, for 100 epochs, on a Tesla V100-SXM2-32GB GPU accelerator. We used the Pytorch framework with a learning rate of $1e-4$ and a batch size of 16. Our loss function was label-smoothing cross entropy loss. We

used the Adagrad optimizer. After training, we leveraged the columns in the projection matrix at the final layer of our classifier. Using each column's weights as a representation for the corresponding class, we computed the pairwise cosine distances matrix between all classes D. We subsequently used D as a similarity measure for affinity clustering to generate H.

We then trained the Swin classifier from ImageNet weights on level-1, whose output space is composed of two classes, with the same settings, for 100 epochs. We chose the parameter K specific to our method as 5, and thus saved the top-5 checkpoints resulting from the level-1 training. For each checkpoint, we initialized a model with the same encoder parameters and with randomized predictor parameters to be trained on level-2, whose output space is composed of 15 classes. We fine-tuned these 5 models as well for 100 epochs, using the same learning rate and batch size, and saved the top achieving checkpoint of each level-2 model.

In the final combination step, we compared the use of model soups and model ensembling. We conducted a combinatorial search to determine the subset of models that yielded the best performance on our validation set. The results of this investigation are mentioned subsequently in our ablation analysis.

5.2 Comparative Evaluation

We proceed to test the best-performing model, which we name C2F-CHART and evaluate our results in comparison with Swin-Chart, the current state-of-the-art method in [8], other deep learning methods evaluated in [8], as well as the ICPR 2022 CHART-Inforgraphics competition's results in [7]. As shown in Table 2, our testing precision, recall, and F1-score demonstrate superior performance to all competition participants and Swin-Chart.

Table 2. Comparative results on ICPR 2022 UB Unitec PMC Dataset.

Team/Method	Recall	Precision	F1-score
Swin-Chart [8]	**93.3**%	93.7%	93.2%
IIT_CVIT [7]	90.1%	92.6%	91.0%
Resnet-152 [10]	89.9%	90.5%	89.7%
ConvNeXt [15]	89.8%	90.6%	89.6%
UB-ChartAnalysis [7]	88.1%	90.0%	88.6%
DenseNet-121 [11]	87.9%	88.7%	87.5%
six_seven_four [7]	80.8%	86.5%	82.7%
CLST-IITG [7]	65.7%	70.4%	65.4%
C2F-CHART (Ours)	93.17%	**95.19%**	**93.98**%

5.3 Ablation Analysis

To explain our ablation analysis, we showcase the results of level-1 and level-2 model training. Table 3 shows the performance of the top 5 level-1 checkpoints on our validation set. It also shows the top achieving level-2 checkpoint trained from the corresponding level-1 model. We can observe that our highest validation score in level-2 doesn't necessarily result from the "top achieving" checkpoint at level-1. When we are referring to "top achieving" or "best performing" here, we are indicating the model with the highest F1-score, as evaluated on the hold-out validation set.

Table 3. Validation F1-scores of the top-5 Level-1 checkpoints and the max. validation F1-scores of the Level-2 model trained from each checkpoint.

Top-5 L1 Checkpoints	Max. Score at L2
98.7264%	95.4865%
98.6381%	**96.1122%**
98.5947%	95.4167%
98.5943%	95.6498%
98.5941%	95.5715%

Consequently, we define three settings of coarse-to-fine curriculum learning. Setting A represents the traditional curriculum learning approach of fine-tuning our top-achieving level-1 checkpoint, as shown in Table 3 and then testing its corresponding top-achieving level-2 checkpoint. Setting B describes taking the top-5 checkpoints trained on level-1, fine-tuning all 5 of them and then testing the top-achieving checkpoint out of all the subsequent models, even if it doesn't result from the level-1 model with the highest score, as is our case. Finally, Setting C describes our method of combination after fine-tuning level-2 using model ensembling on a subset of the 5 final models.

Table 4 compares the three different settings and shows how Setting C achieves the highest F1-score on our validation dataset, as well as the highest precision, recall, and F1-Score on our testing dataset.

Table 4. Comparison between different curriculum learning settings. The left two columns are on the validation set. The three right columns are on the test set.

Method	L1 Val. F1-Score	L2 Val. F1-Score	Recall	Precision	F1-Score
Setting A	**98.72%**	95.48%	92.98%	94.67%	93.53%
Setting B	98.63%	96.11%	92.56%	94.95%	93.6%
Setting C (Ours)	N/A	**96.27%**	**93.17%**	**95.19%**	**93.98%**

This leads us to consider the optimal approach for combining the models, and for choosing the most suitable subset of models to use, referred to subsequently as *ingredients*. We investigated the use of model soups and simple model averaging for our particular use case. In model soups, we aggregate the weights of the ingredients prior to inference, while in simple ensembling, we average the logits produced by each model in our ingredients pool.

To determine the optimal subset of models for the combination step, we ran a comprehensive combinatorial search on our 5 level-2 models shown in the second column of Table 3. For every combination of models (2-, 3-, 4-, and 5-model combinations), we calculated the validation F1-score of both model souping and simple ensembling. We also ran an "iterative greedy" version of model soups, as described in [21], where we allowed each ingredient to be added more than once. Finally, we select the combination that achieves the highest validation score as our final model.

Table 5. Max. validation F1-scores for each combination of models (using ensembling and souping). Subset is chosen from the 5 models with validation F1-scores: 95.48%, 96.11%, 95.41%, 95.64%, 95.57%.

Team/Method	Souping	Ensembling
2-model	95.80%	96.00%
3-model	95.25%	**96.28%**
4-model	95.08%	93.13%
5-model	93.86%	95.98%
Iterative greedy	96.11%	N/A

Table 5 compares between the maximum validation F1-scores for each number of models in both the ensembling and souping techniques, along with the validation score obtained through the iterative greedy souping method. We can conclude that in all combinations, model souping does not outperform the validation score of our highest participating ingredient, while ensembling often does.

Additionally, we investigated the use of another clustering technique to obtain a different hierarchical structure H, that we show in Algorithm 3. Stretcu et al. [18] also suggested using the confusion matrix of a trained classification model to calculate a distance matrix for the affinity clustering algorithm. Using our vanilla Swin classifier model, we obtained H by estimating the confusion matrix C from our dataset using a hold-out validation set. This involved calculating how often our vanilla model incorrectly predicted each class, and identifying the alternate class predicted instead. Given that C may not be symmetric, we followed the approach outlined in [18] and considered our symmetric confusion matrix to be the sum of C and its transpose. We then computed our hierarchy through affinity clustering, using the symmetric confusion matrix as a measure of similarity between classes. This resulted in a different cluster of classes, shown in Fig. 4.

Algorithm 3. Generate Class Hierarchy using Confusion Matrix

Input: Number of classes K, training data, baseline Swin model θ, validation dataset $\{(x_i, y_i)\}_{i=1}^{M}$
Train Swin model θ on the training data
Initialize confusion matrix C of size $K \times K$ with zeros
for each (x_i, y_i) in the validation dataset **do**
  Predict class probabilities $\hat{p}(y_i = c \mid x_i; \theta)$ using θ
  Increment $C[y_i, c]$ by $\hat{p}(y_i = c \mid x_i; \theta)$
end for
Normalize rows of C: $C[i, \cdot] \leftarrow C[i, \cdot] / \sum_{j=1}^{K} C[i, j]$
Compute symmetric confusion matrix $\hat{C} = C + C^{\top}$
Apply affinity clustering on the symmetric confusion matrix $\hat{C}$ to form hierarchy H
Output: Class hierarchy H

Table 6. Validation F1-scores of the top-5 Level-1 checkpoints and the max. validation F1-scores of the Level-2 model trained from each checkpoint.

Top-5 L1 Checkpoints	Max. Score at L2
98.51%	95.96%
98.36%	96.02%
98.35%	95.95%
98.30%	**96.32%**
98.25%	95.90%

We re-ran all of our previous experiments on this other cluster and achieved comparable results, shown in Tables 6, 7 and 8.

5.4 Qualitative Results

Figure 5 presents qualitative results comparing the three distinct coarse-to-fine CL settings we mentioned previously. The first row showcases the success cases

Table 7. Comparison between different CL settings. The left two columns are on the validation set. The three right columns are on the test set.

Method	L1 Val. F1-Score	L2 Val. F1-Score	Recall	Precision	F1-Score
Setting A	**98.51%**	95.96%	92.69%	94.03%	93.14%
Setting B	98.30%	96.32%	93.15%	94.24%	93.53%
Setting C (Ours)	N/A	**96.49%**	**93.40%**	**94.83%**	**93.93%**

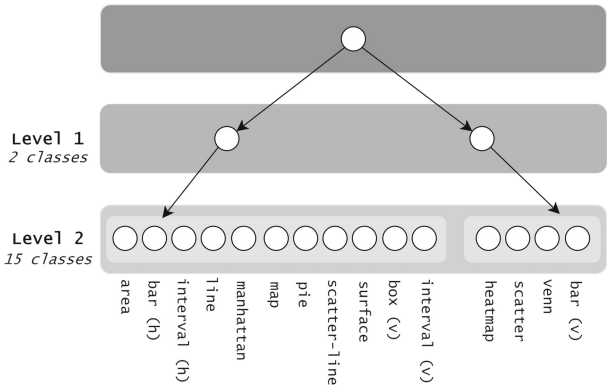


Fig. 4. Automatically computed coarse-to-fine class hierarchy using a confusion matrix as a distance matrix for clustering.

Table 8. Max. validation F1-scores for each combination of models (using ensembling and souping). Subset is chosen from the 5 models with validation F1-scores: 95.96%, 96.02%, 95.95%, 96.32%, 95.90%.

Team/Method	Souping	Ensembling
2-model	95.94%	96.17%
3-model	95.99%	96.39%
4-model	95.70%	**96.49%**
5-model	95.13%	96.29%
Iterative greedy	96.32%	N/A

of Setting A, that are also success cases in Settings B and C. In the second row, we showcase selected samples where Setting B exhibits superior performance compared to Setting A. Finally, in the third row, some samples where Setting C surpasses both are displayed.

We can observe from the figure that the charts where Setting C outperforms are charts whose types closely resemble other types. In the first figure of the third row, a short red curve between the thick black scatter dots transforms the figure from a *scatter* plot to the *scatter-line* plot correctly identified by Setting C. In the second chart, the notations identifying horizontal intervals are spaced, and thus were easily confused by Settings A and B as being scatter plot symbols. As well, in the final chart, the scatter plot symbols were misidentified as just indicators on the line, while Setting C correctly identified the figure as a *scatter-line* plot.

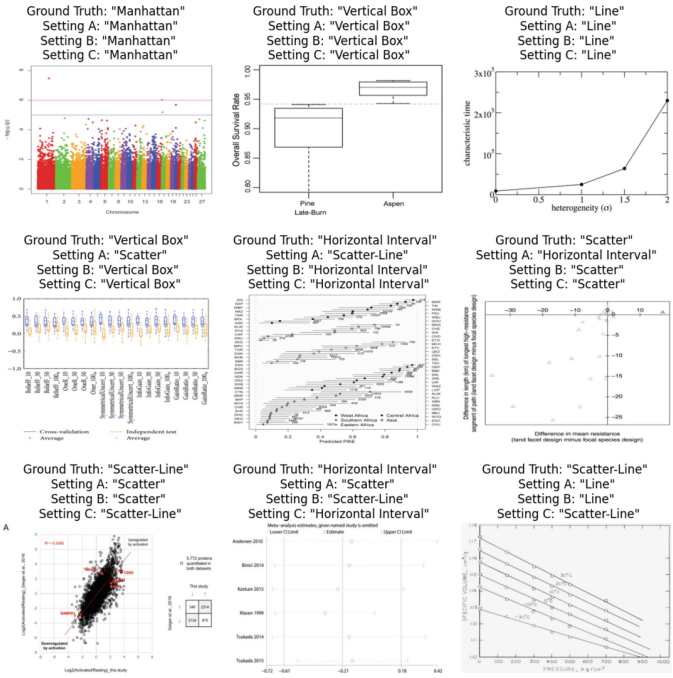


Fig. 5. Qualitative results comparing Settings A, B &C.

6 Conclusion and Future Work

We have implemented a novel approach to chart classification using a modified coarse-to-fine curriculum learning algorithm. Our method outperforms the current SOTA approaches on the ICPR 2022 CHART-Infographics UB Unitec PMC Dataset. We compared our method to traditional coarse-to-fine CL, transformer-based, and CNN-based chart classification approaches. Moving forward, we plan to explore the applicability and adaptability of our method beyond the current benchmark, across other datasets with more diverse chart types, to adequately evaluate its usability in real-world scenarios. Also, since our main interest lies in enhancing accessibility for people with visual impairments, we would like to contribute to an end-to-end chart understanding pipeline, which entails extending our research beyond just chart classification to more extensive accessibility features tailored specifically for visually impaired users.

Acknowledgments. The authors would like to thank the Applied innovation Center (AIC) of the Egyptian Ministry of Communication and Information Technology for funding the research presented in this paper.

References

1. Amara, J., Kaur, P., Owonibi, M., Bouaziz, B.: Convolutional neural network based chart image classification (05 2017)
2. Araújo, T., Chagas, P., Alves, J., Santos, C., Sousa Santos, B., Serique Meiguins, B.: A real-world approach on the problem of chart recognition using classification, detection and perspective correction. Sensors **20**(16) (2020). https://doi.org/10.3390/s20164370
3. Bajić, F., Job, J.: Chart classification using siamese cnn. J. Imag. **7**(11) (2021). https://doi.org/10.3390/jimaging7110220, https://www.mdpi.com/2313-433X/7/11/220
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 41–48. ICML '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1553374.1553380
5. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2017)
6. Davila, K., Setlur, S., Doermann, D., Kota, B.U., Govindaraju, V.: Chart mining: a survey of methods for automated chart analysis. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 3799–3819 (2021). https://doi.org/10.1109/TPAMI.2020.2992028
7. Davila, K., Xu, F., Ahmed, S., Mendoza, D.A., Setlur, S., Govindaraju, V.: Icpr 2022: Challenge on harvesting raw tables from infographics (chart-infographics). In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 4995–5001 (2022). https://doi.org/10.1109/ICPR56361.2022.9956289
8. Dhote, A., Javed, M., Doermann, D.S.: A survey and approach to chart classification. In: International Conference on Document Analysis and Recognition, pp. 67–82. Springer (2023)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
11. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2018)
12. Huang, W., Tan, C.L., Leow, W.K.: Model-based chart image recognition. vol. 3088, pp. 87–99 (07 2003). https://doi.org/10.1007/978-3-540-25977-0_8
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
14. Liu, X., et al.: Chart classification by combining deep convolutional networks and deep belief networks. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 801–805 (2015). https://doi.org/10.1109/ICDAR.2015.7333872
15. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s (2022)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
17. Soviany, P., Ionescu, R.T., Rota, P., Sebe, N.: Curriculum learning: a survey (2022)
18. Stretcu, O., Platanios, E.A., Mitchell, T.M., Póczos, B.: Coarse-to-fine curriculum learning. arXiv preprint arXiv:2106.04072 (2021)

19. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks (2020)
20. Thiyam, J., Singh, S., Bora, P.: Chart classification: a survey and benchmarking of different state-of-the-art methods. Int. J. Document Ana. Recogn. (IJDAR). **27**, 1–26 (06 2023). https://doi.org/10.1007/s10032-023-00443-w
21. Wasfy, O., Basiony, S., Torki, M.: Enhancing lidar semantic segmentation using model soups: Aiccsa23 (11 2023)
22. Wortsman, M., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time (2022)

Vision DualGNN: Semantic Graph Is Not Only You Need

Xiaolong Zheng[1], Jianming Wang[2], Zhitao Xiao[3], and Yukuan Sun[4(✉)]

[1] School of Software, Tiangong University, Tianjin, China
[2] Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tiangong University, Tianjin, China
wangjianming@tiangong.edu.cn
[3] Tianjin Key Laboratory of Optoelectronic Detection Technology and Systems, Tiangong University, Tianjin, China
xiaozhitao@tiangong.edu.cn
[4] Center for Engineering Intership and Training, Tiangong University, Tianjin, China
sunyukuan@tiangong.edu.cn

Abstract. Graph Neural Networks (GNNs) have shown great potential in visual tasks, yet they face challenges in effectively constructing and processing graphs. Vision GNN (ViG) was developed to tackle these issues by segmenting images into patches treated as nodes, with edges formed by connecting the nearest semantic neighbors. However, relying solely on semantic information for graph construction confines itself to a dispersed distribution of object neighbors, leading to inadequate graph processing. To address this issue, we propose Vision DualGNN(VDG), a novel dual graph neural network architecture that leverages both spatial and semantic information to construct and process graph representation of images. We apply a node encoder that transforms image patches into expressive node features. Additionally, we implement a dual-stream GNN that operates on both a spatial graph and a semantic graph. The spatial graph serves as a constraint for the semantic graph, enhancing the node features with spatial awareness. To verify the validity of our architecture, we have conducted our experiments on the ImageNet and CIFAR-100 datasets. And achieved state-of-the-art performance compared to other baseline models.

Keywords: dualgnn · graph convolution · semantic graph · spatial graph

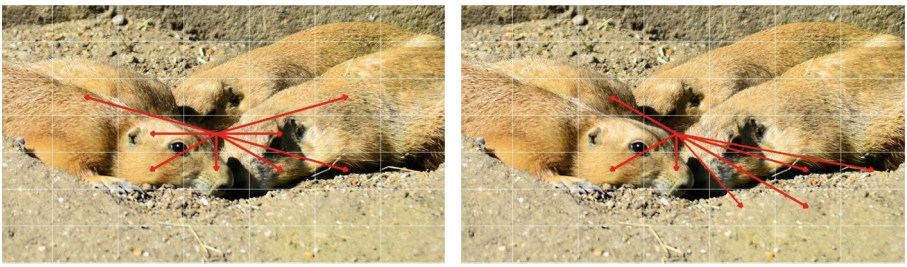
1 Introduction

For visual tasks, there are two main types of architectures: Convolutional Neural Networks (CNNs) [1,2] and Vision Transformers (ViTs) [4]. They encode an

Y. Sun: Supported by the National Natural Science Foundation of China (grant number 62072335) and the Introduction of Foreign Intelligence Project of shijiazhuang Science and Technology Bureau (Project No. 20240024).

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 392–404, 2025.
https://doi.org/10.1007/978-3-031-78107-0_25

image as either a fixed grid of pixels or a sequence of patches. Some methods fuse both the CNN and ViT based models [5,6] to enhance the image representation.



(a) (b)

Fig. 1. The above figures illustrate the node adjacency distribution, Fig(a) represents our method of consisting the semantic graph, where the neighbors of the central node are mostly objects. On the other hand, ViG's [10] method (Fig(b)) has more dispersed neighbors for the central node.

Nevertheless, CNN or Transformer based methods may not be flexible enough to capture the relationships between objects or parts of an object effectively. Graph Neural Networks (GNNs) [7] can model these relationships more comprehensively by introducing graph structures. Zhang *et al.* [8] and Han *et al.* [9] uses graphs and hypergraphs to capture relations between objects. Vision GNN(ViG) [10], represents an image as a graph, where each node corresponds to a patch of the image. Semantic distance is used to quantify the relations among all nodes, and forms edges between the most similar pairs. The graph structure is fed into a ViG model, which allows information flow and transformation among the nodes. A potent connectivity between parts of objects can be built through graph neural network, which would be more flexible and effective for visual perception. However, this network has limitations that are not addressed.

In vision tasks, objects are fundamental elements. However, ViG [10] simply splits an image into patches and treats them as nodes, which restricts the semantic diversity of the nodes. Moreover, the graph constructed by ViG is more dispersed in node distribution with limited spatial information. They add the positional embedding as ViT [4] do, while in the process of building graph structure, the positional embedding used ViG cannot fully express the spatial information in graph level. Figure 1(b) shows that ViG's model can select neighbors that are semantically similar but spatially irrelevant from the background. This may result in erroneous adjacency relations between node pairs, as they are solely based on semantic distance.

To address the limitations discussed above, we propose the Vision DualGNN (VDG) architecture, which incorporates both semantic and spatial information of nodes in graph level. We validate the effectiveness of our model through a series of experiments. In order to enable each node to capture richer semantic

information, we referenced DaVit's approach [11] to apply a node encoder composed of a channel attention and a spatial attention. To incorporate graph level of spatial information into the semantic graph of nodes, we feed these encoded nodes into a dual-stream GNN network. Motivated by the Guo [12], the dual-stream GNN model employs two parallel GNN layers to process two different types of graphs: a semantic graph based on semantic distance and a spatial graph based on spatial distance. The spatial graph is constructed by calculating the Euclidean distance of each node pairs in spatial domain and selecting the closest nodes as neighbors. The semantic graph is constructed by connecting the nearest semantic neighbors of each node. By doing so, the dual-stream GNN module can capture both semantic and spatial relationships more effectively between nodes, and enhance information dissemination and aggregation through interaction and fusion between the two GNN layers.

Our main contributions are as follows:

- We propose a dual-stream graph convolution architecture, which performs graph convolution operations on semantic flow and spatial flow respectively, and fuses features of the two streams to enhance the representation ability of graphs.
- We apply a node encoder, which leverages both channel attention and spatial attention to enhance the representation of nodes.
- We evaluated our model on public datasets such as ImageNet and CIFAR100 and compared it with the original Vision GNN model and other mainstream vision models. The experimental results show that our model has achieved remarkable improvement, our VDG-TI model achieved 77.2% top-1 accuracy, which is 3.3 points higher than the ViG-TI, the VDG-S and VDS-B also achieved excellent results.

2 Related Work

2.1 CNN and Transformer in Vision

Convolutional neural network (CNN) [1] is a deep learning model, which has achieved unprecedented success in the field of computer vision. The representative works include ResNet [1], VGG [2], etc. Vision Transformer (ViT) [4] is the first work that directly applies Transformer [3] to image data. On the basis of ViT, the subsequent work has carried out various improvements and optimizations. DaViT [11] is a dual attention vision transformer that alternates spatial window attention and channel group attention to capture long and short dependencies. Some works also introduce designing hierarchical structure [13], local self-attention [14], increasing depth and diversity, to improve the performance in vision tasks.

2.2 GNN in Vision

Graph nerual network(GNN) [15] can be powerful tools to resist adversarial perturbations [16], it can model the complex and irregular relationships among pixels, regions, or objects in images and videos. Some works [17] present subgraphs

to better explain graphs. Wu [18] utilize the structure features of the graph to better express the node adjacency relationship between objects. Jung [19] use graph contrastive learning for image translation. Vision GNN(ViG) [10] treats an image as graph structure, where nodes represent regions and edges capture their semantic relations. ViG [10] leverages graph convolution to learn powerful graph representations of images, which can be applied to various visual tasks. Vision HGNN (ViHGNN) [9] proposes to model the image as a hypergraph, where each hyperedge can connect multiple nodes. ViHGNN uses a fuzzy clustering method to dynamically construct and update the hypergraph structure, and applies hypergraph convolution to learn expressive hypergraph features.

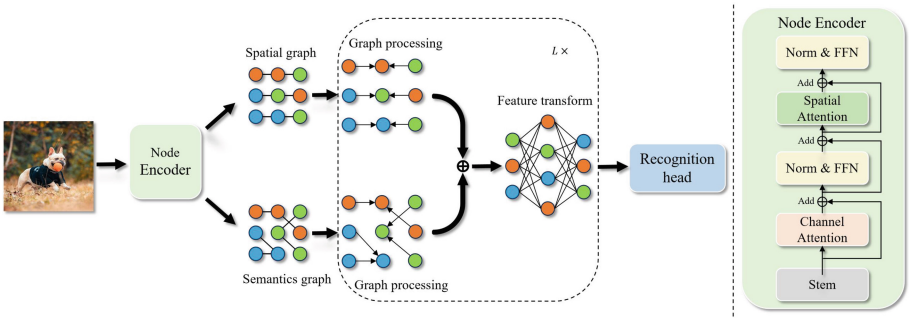


Fig. 2. Framework of VDG: VDG applies a node encoder module to generate diverse and informative features for each node. These encoded nodes are used to construct two different graphs: semantic graph and spatial graph, which reflects the semantic similarity and spatial neighborhoods of nodes, respectively. The spatial graph serves as a constraint for semantic graph to restrict node distribution of central objects. The two graphs are processed through a message passing network to learn graph representation of the image. we concatenate the output of two graphs to transform features for recognition.

3 Vision DualGNN

ViG [10] proposes a novel Visual GNN (ViG) architecture that represents an image as a graph structure, and apply a Grapher module to extract graph-level features. Here we present our updated method Vision DualGNN(VDG). The overall architecture of VDG is illustrated in Fig. 2. Given an input image with size $H * W * 3$, We followed the previous work to split the image into patches, where we have $\boldsymbol{X} = [\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}, ..., \boldsymbol{X}_N]$, each patch $\boldsymbol{X_i}$ with a feature dimension C, we define these patches as nodes. To enhance the expressiveness of the nodes, we apply a node encoder module to encode each node, which is essential for the graph level processing. We redesign the Grapher module by incorporating the DualGNN network as the main network structure. We reuse the semantic

distance flow as a component of the architecture. In addition, we add another gnn flow that exploits the spatial information. For each node X_i, we find its neighbors with the closest semantic distance and the shortest spatial distance, adding an edge e_{ji} to each node pairs, and we will get 2 graphs $G_1 = \{V, \xi_1\}$, $G_2 = \{V, \xi_2\}$, where ξ_1 represents edges with closest semantic distance and ξ_2 represents edges with shortest spatial distance, the 2 graphs will be processed through 2 flows of gnn, respectively.

To implement the structure with fewer parameters, we built the network following ViG's isotropic architecture, three versions with different models sizes were offered. The details are listed in Table 1

Table 1. Variants of our isotropic VDG architecture. The FLOPs are calculated for the image with 224×224 resolution. 'Ti' denotes tiny, 'S' denotes small, and 'B' denotes base.

Model	Depth	Dimension D	Params (M)	FLOPs (B)
VDG-Ti	12	192	9.1	1.8
VDG-S	16	320	29.4	6.5
VDG-B	16	480	64.5	14.4

3.1 Node Encoder

Feature encoding is crucial when converting an image into a graph structure. ViG [10] simply constructs the nodes by multi-layers of convolution, which limits the richness of the current node's representation of the corresponding region. Here we exhibits our node encoder that enrich node expresentations.

By sending the image to stem module, we get a feature map of size $C * H * W$, where each pixel of the last two dimensions is defined as a node. To enrich the semantic representation, we use the channel attention mechanism to update the features. The channel attention mechanism can adaptively increase the diversity of node expression, thus enhancing the differentiation and representation of each node. Specifically, we define the nodes $V = v_1, v_2, v_3, ..., v_N$. We first use a fully connected layer to get query, key and value of each node. Specifically, for a node v_i, we have:

$$v_q, v_k, v_v = split(\boldsymbol{W_c}(v_i)) \tag{1}$$

where $\boldsymbol{W_c}$ is the weight matrix, and we can get v_q, v_k, v_v of node v_i. By using the attention mechanism, we will get an updated node v_i

$$v_i = \boldsymbol{W_p}(softmax((v_q * scale) * v_k) * v_v) \tag{2}$$

where *scale* represents the scaling factor, $\boldsymbol{W_p}$ is the projection matrix.

After channel attention mechanism applied in nodes, we calculate the attention weight of all the other nodes, and we sum this attention weight vector

weighted with the features of all the other nodes to get an update node v_i representation, which synthesizes the information about the node from all other nodes.

$$v_i = \sum_{j=1}^{n} softmax((v_i * \boldsymbol{W_{q1}}) * (v_j * \boldsymbol{W_{q2}})) * v_i \tag{3}$$

where $\boldsymbol{W_{q1}}$, $\boldsymbol{W_{q2}}$ is the projection matrix of node v_i and v_j, so we can get a new feature vector for each node. The multi-head spatial attention module can consider different subspaces and all other neighbouring nodes at the same time, which enhances the information exchange and representation of nodes.

3.2 DualGNN

The ViG [10] model constructs the graph structure based on the semantic distance of nodes by splitting the image into nodes and connecting the nearest neighbors based on the feature similarity between nodes. However, this approach limits the node distribution and spatial information of objects, which is vital in visual tasks. For example, in image classification tasks, when recognising animal categories, the spatial relationships of parts such as the head, body and tail of an animal can provide useful clues. In the context of graph-level processing, the commonly used positional embedding in Vision Transformers (ViTs) architecture may not effectively provide spatial information as graph processing goes deeper. Adding spatial graph can help the semantic graph better understand the distribution of neighbor nodes, so as to reduce the wrong selection of background nodes with similar semantics when selecting neighbor nodes. Therefore, we propose a new way of constructing the graph structure. Use two different approaches to construct two graph structures, the semantic graph and the spatial graph.

For the semantic graph, we continue the previous work by calculating the feature semantic distance between each pair of nodes, connect each node to its nearest k neighbors based on the magnitude of the distance, i.e., we select the K nodes with the least distance as neighbouring nodes and connect them with an edge, the adjacency matrix can be expressed as :

$$A_s = \begin{cases} 1, & topk(s(v_i, v_j)) \\ 0, & others \end{cases} \tag{4}$$

where $s(v_i, v_j)$ means the semantic distance between node v_i and v_j.

For the spatial graph, we compute the spatial distance between each pair of nodes. We use the Euclidean distance in spatial domain to measure the distance between two nodes on the image, connect each node to its nearest neighbors based on the magnitude of the distance. Finally, we select the K nodes with the smallest distance as neighbour nodes and connect them with an edge, the adjacency matrix can be expressed as:

$$A_d = \begin{cases} 1, & topk(d(v_i, v_j)) \\ 0, & others \end{cases} \tag{5}$$

where $d(v_i, v_j)$ means the Euclidean distance between node v_i and v_j.

We take these two graph structures as inputs and feed them into two parallel graph convolution layers for message passing in graphs, respectively. We use the GIN as the basic GNN network to aggregate and update the features of each node, with the network goes deeper, increasing the number of nodes as the graph convolution layers get deeper. We splice the outputs of the two graph convolution modules to obtain fused feature that incorporates semantic and spatial information, which can be expressed as:

$$\boldsymbol{X_s} = GraphConv(\boldsymbol{X_{in}}\boldsymbol{W_{in}}, \boldsymbol{A_s}) \tag{6}$$

$$\boldsymbol{X_d} = GraphConv(\boldsymbol{X_{in}}\boldsymbol{W_{in}}, \boldsymbol{A_d}) \tag{7}$$

$$\boldsymbol{X_{out}} = (\boldsymbol{X_s}\boldsymbol{W_{out}}) \oplus (\boldsymbol{X_d}\boldsymbol{W_{out}}) \tag{8}$$

where $\boldsymbol{X_{in}}$, $\boldsymbol{X_{out}}$ represent the input and output node features, $\oplus$ represents concatenate operation, $\boldsymbol{W_{in}}$, $\boldsymbol{W_{out}}$ represent the input and output projection, and $\boldsymbol{A_s}$, $\boldsymbol{A_d}$ represents the semantic graph and spatial graph, respectively.

Table 2. Experimental Results on ImageNet

Model	Resolution	Params (M)↓	FLOPs (B)↓	Top-1↑	Top-5↑
ViT-B/16	384×384	86.4	55.5	77.9	-
DeiT-Ti [20]	224×224	**5.7**	**1.3**	72.2	91.1
DeiT-S [20]	224×224	**22.1**	4.6	79.8	95.0
DeiT-B [20]	224×224	86.4	17.6	81.8	95.7
ViHGNN-Ti [9]	224 × 224	8.2	1.8	74.3	92.5
ViHGNN-S [9]	224 × 224	23.2	5.6	81.5	**95.7**
ViHGNN-B [9]	224 × 224	88.1	19.4	**82.9**	**96.2**
ViG-Ti [10]	224×224	7.1	**1.3**	73.9	92.0
ViG-S [10]	224×224	22.7	**4.5**	80.4	95.2
ViG-B [10]	224×224	86.8	17.7	82.3	95.9
VDG-Ti(ours)	224×224	9.1	1.8	**77.2**	**93.8**
VDG-S(ours)	224×224	29.4	6.5	**81.8**	**95.7**
VDG-B(ours)	224×224	**64.5**	**14.4**	82.4	95.9

4 Experiments

4.1 Datasets and Experimental Settings

We use two public image datasets to evaluate the performance of our VDG model, ImageNet ILSVRC 2012 and CIFAR-100. We followed most of the ViG's

[10] setting for fair comparison, we train the model in 300 epochs, set the batch size to be 1024, use the adam optimizer, and the difference we set is the min learning rate of the base model to be 1e-6, all other settings are the same as the ViG. We implement the networks in iostropic architecture and train all our models on 8 NVIDIA A100 GPUs.

4.2 Baselines and Results

We followed ViG's work to establish the baselines, we do experiments mainly based on iostropic architecture, and compare our VDG against the recent isotropic models on ImageNet dataset, such as ViG [10], ViHGNN [9], ViT [4] and Deit [20]. We adopted a similar training strategy as ViG [10], and evaluated our model on two benchmark datasets, ImageNet and CIFAR100. The results are shown in Table 2 and Table 3.

We experimented our approach on the ImageNet dataset, comparing it with the original ViG [10] model. The results are shown in Table 2. It can be seen from the experimental results that our models of three sizes all achieve better results in performance or speed compared to ViG. For example, our VDG-B achieves 82.4% top-1 accuracy which is comparable to ViG-B model but with lower computing cost. This demonstrates that our model can utilize the spatial information more effectively from the image objects and leverage it to enhance the expressiveness of the model.

Table 3. Experimental Results on CIFAR-100

Model	Params(M)↓	Acc(%)↑
ResNet18 [1]	11.18	63.41
ResNet50 [1]	23.53	61.68
ResNeXt-50	25.03	84.42
ResNeXt-29-8×64d	34.4	82.23
ViT-Lite-6/4	3.19	73.33
NesT-B	97.2	82.56
ViG-S [10]	21.8	83.12
VDG-S(ours)	28.5	**84.63**

We also experimented our model on CIFAR-100. To get a baseline competition, we trained the ViG-S on CIFAR-100, using the same experimental setting as trained on ImageNet dataset expect for batch size. We set batch size at 256, the result can be seen in Table 3. Our model achieved higher accuracy than ViG-S as well as other competitive existing methods. These results shows that the VDG model capture more complex and irregular objects in images and to learn richer graph representations of the image.

To rigorously evaluate the generalizability of our proposed model architectures, we conducted a series of fine-tuning experiments using models pre-trained on the ImageNet dataset. Specifically, we utilized ViG-S and VDG-S, which were initially trained on ImageNet, as foundational models. These pre-trained models were then subjected to fine-tuning on the CIFAR-100 dataset for varying numbers of epochs: 5, and 10, respectively (Table 4).

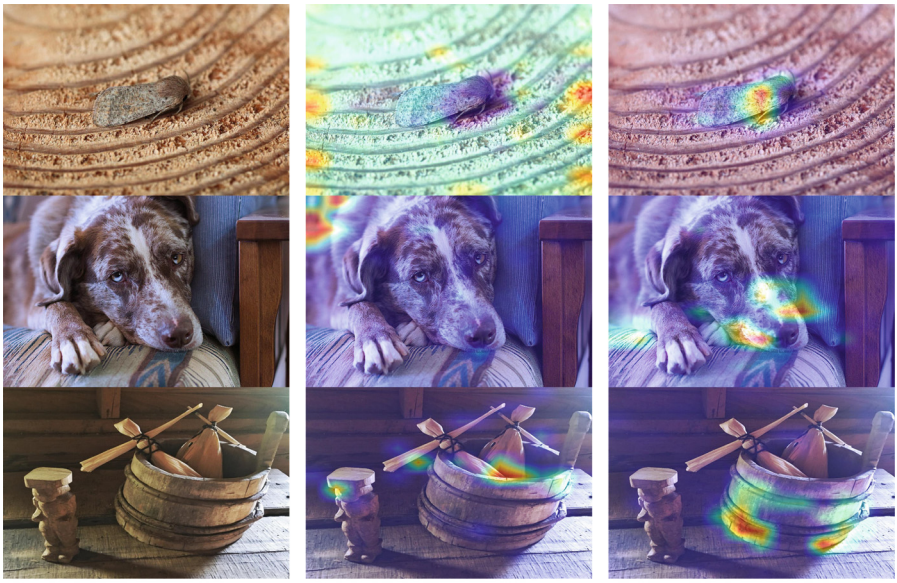
Table 4. Fine-tuning Results on CIFAR-100

Model	5 epochs(Acc%)	10 epochs(Acc%)
ViG-S	72.17	73.32
VDG-S	76.47	79.35

The results of these experiments were highly illuminating. Not only did our models display markedly faster convergence rates during the fine-tuning phase, but they also consistently outperformed competing models in terms of final classification accuracy on the CIFAR-100 dataset. This empirical evidence strongly supports the assertion that our proposed architectures possess exceptional generalization capabilities, capable of adapting effectively to new datasets with minimal fine-tuning. These findings underscore the versatility and robustness of our models, positioning them as promising candidates for a wide array of computer vision tasks requiring high degrees of adaptability and performance.

4.3 Visualization of Feature Map

To further understand why VDG works, we visualize feature map of our model. For this visualization, we use the features extracted from the final layer of the graph processing module, the information of spatial graph and semantic graph are fused together to form a global feature representation. We do comparisons between the ViG-S [10] model, and VDG-S model. Figure 3 shows the original image and Grad-CAM feature maps corresponding to the two models in different categories of images. We randomly selected three representative images from the dataset corresponding to three categories (insects, dogs, wooden barrels) as our comparison samples. As we can clearly see in Fig. 3(c), the VDG model is capable of capturing spatial relationships and topological structures within objects as well as semantic associations and similarities. This feature represents the ability to better focus on features of object itself in the image. At the same time, our VDG model makes use of the structure and topological characteristics of the graph, and can adaptively adjust adjacency matrix and node characteristics of the graph, so as to enhance the significance and differentiation of objects, rather than being distracted by background or other irrelevant information, so as to locate and identify objects in images more accurately.



(a) (b) (c)

Fig. 3. Visualization of the feature map for ViG-S and our VDG-S model. (a) represents the input image, (b) represents the feature map of ViG-S and (c) represents the feature map of ours VDG-S. We extracted the features before the classification head.

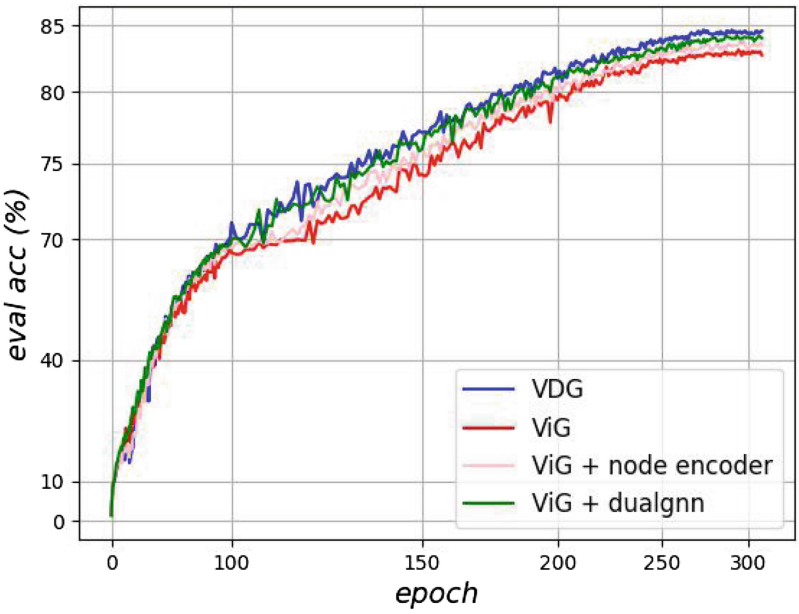


Fig. 4. Analysis of the role of each module in the model: experimental ablation methods for accuracy curves on CIFAR100

Table 5. Ablation Study on VDG

ViG	Node Encoder	DualGNN	Pos Embedding	Params(M)↓	Acc(%)↑
✓	✗	✗	✓	**21.8**	83.12
✓	✓	✗	✓	23.6	83.79
✓	✗	✓	✓	26.7	84.21
✓	✓	✓	✗	28.4	84.57
✓	✓	✓	✓	28.5	**84.63**

4.4 Ablation Study

To verify the contribution of each component of our model, we designed ablation experiments to compare the performance under different model configurations. We conducted experiments on the CIFAR100 dataset. Using ViG-S as the baseline model. The specific experimental results are shown in the Table 5, the training accuracy curve is shown in Fig. 4. As can be seen from the table, both of the modules we proposed have significantly improved the performance of the model. the node encoder module can effectively enhance the model's attention to image features and improve the expressive ability of nodes. The dualgnn module can utilize both the spatial and semantic information better to enrich the graph level representation of an image. Also, the positional embedding that added in ViG can provide limited spatial information to improve performance. As we can see from Fig. 4, by combining these two modules, our model achieves a better classification accuracy on the CIFAR100 dataset, exceeding the baseline model, proving the validity and superiority of our model.

5 Conclusion

In this work, we introduce Vision DualGNN, a novel framework designed to harness both the semantic richness and spatial context inherent in visual data. By innovatively integrating dual graphs—one emphasizing semantic relationships, the other focusing on spatial configurations—our model demonstrates a remarkable capability to encode complex visual scenes with enhanced accuracy and flexibility. This dual perspective not only enriches the representation of individual nodes but also facilitates a more nuanced understanding of their interrelations within the broader context of an image. Yet, we acknowledge the presence of limitations that warrant further exploration. Specifically, the current strategy for node selection may not fully exploit the potential of our graph structures, potentially leading to suboptimal performance in certain cases.

Looking ahead, our research agenda will prioritize refining node selection methodologies to ensure they are more principled and context-aware, thereby optimizing graph construction. Simultaneously, we aim to streamline the computational architecture without compromising on performance, seeking to make Vision DualGNN more accessible and scalable for widespread deployment.

Through these enhancements, we envision a future where our model can seamlessly integrate into various computational ecosystems, delivering state-of-the-art results with unparalleled efficiency.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (No. 62072335) and the Introduction of Foreign Intelligence Project of Shijiazhuang Science and Technology Bureau (Project No. 20240024).

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
2. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: ACPR, pp. 730–734 (2015)
3. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
4. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: ICLR (2021)
5. Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A.: A Fast hybrid vision transformer using structural reparameterization. In: ICCV, FastViT (2023)
6. Wang, A., Chen, H., Lin, Z., Han, J., Ding, G.: Revisiting mobile CNN From ViT Perspective. In: CVPR, RepViT (2024)
7. Kofinas, M., et al.: Graph neural networks for learning equivariant representations of neural networks. In: ICLR (2024)
8. Zhang, Z., et al.: Object relational graph with teacher-recommended learning for video captioning. In: CVPR, pp. 13275–13285 (2020)
9. Han, Y., Wang, P., Kundu, S., Ding, Y., Wang, Z.: Vision HGNN: an image is more than a graph of nodes. In: ICCV, pp. 19878–19888 (2023)
10. Han, K., Wang, Y., Guo, J., Tang, Y., Enhua, W.: Vision GNN: an image is worth graph of nodes. NeurIPS **35**, 8291–8303 (2022)
11. Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., Yuan, L.: DaViT: dual attention vision transformers. In: ECCV, pp. 74-92 (2022). https://doi.org/10.1007/978-3-031-20053-3_5
12. Guo, L., Sun, Z., Zhang, Q., Chen, H.: Dual-Encoding Transformer for Graphs. ICLR, Reach the Remote Neighbors (2023)
13. Tang, S., Zhang, J., Zhu, S., Tan, P.: Quadtree attention for vision transformers. In: ICLR (2022)
14. Pan, X., Ye, T., Xia, Z., Song, S., Huang, G.: Slide-Transformer: hierarchical vision transformer with local self-attention. In: CVPR, pp. 2082–2091 (2023)
15. Xia, Z., Yang, H., Wang, B., Jia, J.: GraphGuard: provably robust graph classification against adversarial attacks. In: ICLR (2024)
16. Zhao, K., Kang, Q., Song, Y., She, R., Wang, S., Tay, W.: Adversarial robustness in graph neural networks: a Hamiltonian approach. In: NeurIPS, pp. 3338–3361 (2023)
17. Tan, Q., Zhang, X., Liu, N., et al.: Bring your own view: graph neural networks for link prediction with personalized subgraph selection. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 625–633 (2023)
18. Wu, Q., Yang, C., Zhao, W., He, Y., Wipf, D., Yan, J.: DIFFormer: scalable (graph) transformers induced by energy constrained diffusion. In: ICLR (2023)

19. Jung, C., Kwon, G., Ye, J.C.: Patch-wise graph contrastive learning for image translation. In: AAAI (2024)
20. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)

Enhancing Graph-Based Clustering with the Regularity Lemma

Jian Hou[1(✉)], Juntao Ge[1], Huaqiang Yuan[1], and Marcello Pelillo[2,3]

[1] School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China
houjian@dgut.edu.cn
[2] DAIS, Ca' Foscari University, Venice 30172, Italy
[3] European Centre for Living Technology, Ca' Foscari University, Venice 30123, Italy

Abstract. Graph-based clustering has been shown to be promising, partly due to the rich data relationship encoded in affinity graphs. However, the graph representation also means a large computation and storage load for large-scale datasets. Several previous works show that it is promising to improve graph-based clustering based on Szemerédi's regularity lemma, which roughly states that each graph can be partitioned into a small number of random-like graphs. We find in experiments that the results of these methods are sensitive to the involved parameters, and therefore propose a thorough investigation of the influence of several parameters on clustering results and discuss the reason behind their behaviors. As a result, we find out some clues as the determination of these parameters in practical applications. In experiments on a number of real datasets, we find that with proper parameters, the regularity lemma is able to improve both the clustering quality and computation efficiency significantly. Furthermore, experiments show that two relatively old-fashioned algorithms are enhanced to outperform recent state-of-the-art ones. This work goes a step further in extending the application of the regularity lemma from pure theoretical to practical realms.

Keywords: graph-based clustering · regularity lemma · reduced graph

1 Introduction

Graph-based clustering is one of the most popular clustering approaches, with the spectral clustering (SPC), affinity propagation [3] and dominant set (DSet) [19] algorithms belonging to this type. On one hand, graph-based clustering requires as input the pairwise data similarity matrix, which encodes rich data distribution information. On the other hand, the well-established results in graph theory can be applied to solve graph-based clustering problems, with spectral clustering as a typical example. Therefore it is not surprising that graph-based

This work is supported by the National Natural Science Foundation of China under Grant No. 62176057.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 405–420, 2025.
https://doi.org/10.1007/978-3-031-78107-0_26

clustering achieves impressive results. However, the graph representation indicates the similarity matrix of size $n \times n$ for n data points, which further means a large computation and storage load for large datasets. For the purpose of reducing the complexity of the clustering process, different algorithms, e.g., Nyström method [8], optimal embedding [21], out-of-sample method [18] and kNN graph.

Several previous works [20,22,24] proposed a promising approach to enhance graph-based clustering with Szemerédi's regularity lemma [25]. This lemma shows roughly that every graph can be partitioned into subgraphs, where the majority of subgraph pairs behave like random bipartite graphs. By treating each subgraph as a node and defining the edge weights based on subgraph pairs, we obtain a reduced graph, which is a compact representation of the original one. The final clustering results can then be obtained by performing clustering with the reduced graph and mapping labels back to the original one. We find in experiments that this method is sensitive to three parameters in regular partitioning. Therefore we make a thorough investigation on how the clustering results are influenced by these parameters and provide an explanation as to the influences. We therefore find out the appropriate ranges of these parameters in practical application. With proper parameters, this method is shown to improve the clustering results of the SPC and DSet algorithms evidently with significantly less running time, demonstrating the potential of the regularity lemma in dealing with large-scale clustering. Following [12], this work achieves the investigation on two more representative graph-base clustering algorithms.

Our contributions in this paper are as follows. First, we investigate the influence of regular partitioning parameters on clustering results, which is helpful to the parameter tuning in practical applications. Second, we explain why the regularity lemma is able to improve the clustering results of the DSet and SPC algorithms evidently with significantly less running time. Third, we show with experiments on real datasets that traditional graph-based clustering algorithms can be enhanced with this lemma to outperform recent state-of-the-art ones.

In Sect. 2 we introduce the regularity lemma and some previous works. Then in Sect. 3 we present in details how to improve graph-based clustering based on the regularity lemma. Section 4 is devoted to extensive experimental validation and comparisons. Finally, we arrive at some conclusions in Sect. 5.

2 Related Works

In this part we firstly introduce the regularity lemma and the regular partitioning method, and then review a few practical applications of this lemma.

2.1 Regularity Lemma

Szemerédi's regularity lemma is one of the best known achievements in extremal graph theory. This lemma shows that every graph can be partitioned into subgraphs so that subgraph pairs behave like random bipartite graphs. Therefore it is convenient to carry over results of random graphs to ordinary graphs.

Let $G = (V, E)$ denote an undirected graph without self-loops, with V being the vertex set and E denoting the edge set. With $A \subseteq V$, $B \subseteq V$ and $A \cap B = \emptyset$, the edge density between A and B is defined as

$$d(A, B) = \frac{e(A, B)}{|A||B|}, \tag{1}$$

with $e(A, B)$ denoting the count of edges connecting one endpoint in A to the other in B, and $|\cdot|$ stands for the number of vertices in a set.

Before presenting the regularity lemma, we need the following concepts.

Definition 1. *Regular pair. Given $A \subseteq V$, $B \subseteq V$, $A \cap B = \emptyset$, and a positive constant ε, if for every $X \subset A$ and $Y \subset B$ such that $|X| > \varepsilon|A|$ and $|Y| > \varepsilon|B|$ we have $|d(X, Y) - d(A, B)| < \varepsilon$, then we call (A, B) as an ε-regular pair.*

Since the edge density between two relatively large subsets is similar to that between the regular pair, we see that edges between a regular pair are distributed quite uniformly.

Definition 2. *Equitable partition. Given a partition P of the vertex set $V = V_0 \cup V_1 \cup \cdots \cup V_k$. If $|V_1| = |V_2| = \cdots = |V_k|$, then P is called an equitable partition.*

The class V_0 is called the exceptional class, which may be empty, and is only used to guarantee that all the other classes have the same cardinality.

Definition 3. *Regular partition. Given an equitable partition P of the vertex set $V = V_0 \cup V_1 \cup \cdots \cup V_k$, with V_0 being the exceptional class. If $|V_0| < \varepsilon|V|$ and all but at most εk^2 pairs (V_i, V_j), $1 \leq i < j \leq k$, are ε-regular, then the partition P is an ε-regular partition.*

Now we are ready to introduce the regularity lemma below.

Lemma 1. *(Regularity Lemma [25]) For every positive real ε and positive integer m, there exist positive integers $N = N(\varepsilon, m)$ and $M = M(\varepsilon, m)$ such that every graph $G = (V, E)$ with $|V| > N$ has an ε-regular partition into $k + 1$ classes, where $m \leq k \leq M$.*

Roughly speaking, the regularity lemma states that every sufficiently large graph could be partitioned such that every subgraph pair could be regarded as a regular pair, i.e., the edges between two subgraphs have a uniform distribution. Based on this special distribution, we are able to obtain a reduced graph from the original graph, which is shown to inherit the major structure of the original graph with a much smaller size. The details are presented in Sect. 3.

2.2 Practical Applications

Szemerédi's regularity lemma has been extended in several aspects, including weaker regularity notions [16], stronger regularity notions [16,26], the variants for sparse graphs [23] and for hypergraphs [7]. However, the application of the

regularity lemma has been limited to proving pure theoretical results for a long time, and a possible reason is that this lemma applies to graphs of astronomically large sizes.

Probably the first practical application of the regularity lemma is presented in [24]. In order to apply this lemma to relatively small graphs, [24] no longer insisted on obtaining provably regular partitions. Instead, a few simple heuristics are proposed to generate approximately regular partitions, and a reduced graph is then built from the original one. Then DSet clustering is performed with the reduced graph and we map the labels to the original graph. Experiments showed that this algorithm outperforms the original one on several datasets.

Following [22,24] also applies the regularity lemma to pairwise clustering. It adopts a similar idea as [24], i.e., using a few modifications to obtain approximately regular partition and then doing clustering on the reduced graph. This work tests the regular partitioning methods presented in both [2,9], and uses spectral clustering as the base algorithm. In addition, this work tests different parameter values in experiments. Experimental results confirm that the regularity lemma is able to improve spectral clustering on some datasets.

In [5], the regularity lemma is used to summarize large graphs with noises and search graphs efficiently and robustly. A few heuristics are proposed to obtain approximately regular partitions from relatively small graphs, which are then used to build the essential structure of graphs.

As pointed in [4,6,17,20], Szemerédi's regularity lemma is a promising approach to big data analysis. Although at present its practical applications are very limited due to severe constraints imposed to accommodate arbitrary graphs, it has been shown that even crude approximation may work well for relatively small graphs occurring in practice. Therefore more efforts are needed to benefit from this theoretic result in practice.

3 Enhancement Based on the Regularity Lemma

The method to enhance graph-based clustering based on the regularity lemma can be described briefly as follows. Firstly, we partition the original graph into subgraphs with the regularity lemma. In the second step, we build a reduced graph based on the obtained subgraphs. Next, we perform graph-based clustering with the reduced graph. In the last step, we map the obtained labels from the reduced graph to the original one. In the four steps, the third and fourth ones are easy to implement, and therefore in the following we only describe the first and second steps in details.

3.1 Partitioning the Graph

The original proof of the regularity lemma in [25] didn't provide a method to obtain the regular partition. The first algorithm to create regular partitions on arbitrary graphs was presented by Alon et al. in [2]. After that, [9] proposed another method which is shown in [22] to perform similar to the one of Alon et al. in [2]. In this paper the algorithm of Alon et al. is adopted.

It is necessary to introduce the following lemma before presenting the algorithm of Alon et al.

Lemma 2. *(Alon et al. [2]) Let H be a bipartite graph with equal classes $|A| = |B| = n$. Let $2n^{-1/4} < \varepsilon < \frac{1}{16}$. There is an $O(n^{2.376})$ algorithm that verifies that H is ε-regular or find two subsets $A' \subset A$, $B' \subset B$, $|A'| \geq \frac{\varepsilon^4}{16}n$, $|B'| \geq \frac{\varepsilon^4}{16}n$, such that $|d(A,B) - d(A',B')| \geq \epsilon^4$.*

This lemma means roughly that one can either verify that a pair (A, B) is regular, or provide certificates that it is not. On this basis, the regular partitioning algorithm of Alon et al. is described below.

1) Create the initial partition: Divide the set V of vertices arbitrarily into an equitable partition P_1 with classes $V_0, V_1, \cdots, V_b$, with $|V_1| = \lfloor n/b \rfloor$ and therefore $|V_0| < b$. Denote $k_1 = b$.
2) Check regularity: For every pair (V_r, V_s) of P_i, $1 \leq r < s \leq k_i$, verify if it is ε-regular or find $X \subset V_r$, $Y \subset V_s$, $X \geq \frac{\varepsilon^4}{16}|V_r|$, $Y \geq \frac{\varepsilon^4}{16}|V_r|$, such that $|d(X,Y) - d(V_r,V_s)| \geq \varepsilon^4$.
3) If at most $\varepsilon\binom{k_i}{2}$ pairs are not verified as ε-regular, then P_i is a regular partition. The partitioning process is terminated.
4) Otherwise, apply a refinement algorithm [2] to obtain a partition P' with $1 + k_i 4^{k_i}$ classes.
5) Let $k_{i+1} = k_i 4^{k_i}$, $P_{i+1} = P'$, $i = i + 1$, and go to Step 2.

The Step 4 shows that in the partitioning process, each iteration increases classes exponentially, and therefore this method applies only for graphs of astronomically large sizes. In order to make this method applicable for practical datasets of relatively small sizes, some modifications to the original method has been proposed in [5,20,22,24]. In this paper we adopt the following modifications. First, Step 2 needs to find out all the irregular pairs, thereby resulting in the exponential growth of classes. Therefore for each class, we consider at most one irregular pair containing this class, allowing to generate a constant number of subclasses in Step 4. In addition, the degree-based greedy method [5] is adopted to generate certificates to check the regularity of pairs. Finally, the number of iterations and that of classes required to obtain a provable regular partition are too large to be considered. Therefore the iteration is terminated if the class size is smaller than a threshold. With these modifications, we obtain only approximately regular partitions in general.

There are three parameters involved in the regular partitioning process. The first is ε in defining the regular pair, and the second is b which denotes the number of classes in initial partition. The third parameter is the maximum compression ratio $\epsilon = |R|/|G|$, with $\lfloor 1/\epsilon \rfloor$ representing the minimum class size. We will investigate how these three parameters influence clustering results with experiments on real datasets.

3.2 Building the Reduced Graph

Based on the regularity lemma, a graph can be partitioned such that the edges between each pair of subgraphs have a quite uniform distribution. If we treat

each subgraph as a vertex, and calculate the edge weights based on Eq. (1), we obtain a new graph. This graph is typically much smaller than the original one, and is commonly referred to as the *reduced graph*. The reduced graph can be obtained formally as follows. Given the original graph $G = (V, E)$ partitioned into subgraphs (V_1, E_1), (V_2, E_2), $\cdots$, (V_k, E_k). Each subgraph is transformed to a vertex in the reduced graph R, and if two subgraphs form a regular pair and their edge density is greater than a threshold d_0, the two corresponding vertices in R are adjacent. Note here that the exceptional class V_0 is excluded from building the reduced graph.

One important ability of the reduced graph is to inherit many properties of the original graph with a much smaller size than the latter [13]. With an integer t and a graph $R = (V_R, E_R)$, we construct a graph $R(t)$ as follows. Each vertex $x \in V_R$ is mapped to a set V_x of t independent vertices in $R(t)$. With two vertices $x, y \in V_R$, if (x, y) is an edge in R, then all the $u \in V_x$ and $v \in V_y$ in $R(t)$ are adjacent. Given a positive integer m and $d_0 > \varepsilon > 0$, we further build a graph G as follows. Every vertex in R is mapped to m vertices in G, and every edge in R is mapped to a regular pair whose edge density is greater than d_0. Here we observe that R is a reduced graph of G.

The Key Lemma [13] shows how the reduced graph inherits properties of the original one. It states roughly that every small subgraph of $R(t)$ lies also in G. Noticing that $R(1) = R$, wee see that a small subgraph of a reduced graph R lies also in the original graph G. As the number of subgraphs in R is much smaller than that in G, we can regard R as a compressed, compact version of G, which contains the essential structures of G. Unlike traditional sampling of data points, here the graph compression is accomplished by sampling of edges or larger structures. Since data point sampling can be used in large-scale clustering, here we use the reduced graph obtained by structure sampling to do clustering.

With Eq. (1) we are able to calculate edge weights of R for unweighted G. However, in application to graph based clustering, the original graph is edge-weighted, with the weight equaling to the similarity of two data points. Therefore we revise the edge weights of the reduced graph R to be

$$d_w(X, Y) = \frac{\sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} w(x_i, y_j)}{|X||Y|}, \tag{2}$$

where $x_i \in X$ and $y_i \in Y$.

3.3 Procedures

Enhancing graph-based clustering with the regularity lemma can be accomplished as follows. Given the set of n data to be clustered, we calculate the pairwise similarity matrix W, which encodes the information of the original graph $G = (V, E)$. With the partitioning method provided in Sect. 3.1, we partition the vertex set V into a regular partition $V_0, V_1, \cdots, V_k$. Based on $V_1, V_2, \cdots, V_k$ the reduced graph R is built with the method provided in Sect. 3.2, where we

Table 1. Characteristics of datasets.

dataset	NP	ND	NC	dataset	NP	ND	NC	dataset	NP	ND	NC
Appendicitis	106	7	2	Iris	150	4	3	Seeds	210	7	3
Arcene	200	10000	2	Leaves	1600	64	100	Segment	2310	19	7
Balance-scale	625	4	3	Libras	360	90	15	Thyroid	215	5	3
Ecoli	336	7	8	Olivertti	400	28	40	USPS	11000	256	10
Glass	214	9	6	SCC	600	60	6	Wine	178	13	3

calculate edge weights using Eq. (2). With R as input, we do graph-based clustering and obtain the labels $l_1, l_2, \cdots, l_k$. Then we do the label mapping from the reduced graph to the original one, i.e., the data in V_i are assigned the label l_i. As to the data in V_0 which are left unclustered, each of them is simply assigned to the nearest cluster.

4 Experiments

To demonstrate the performance of graph-based clustering algorithms enhanced by the regularity lemma, we conduct experiments on 15 real datasets taken from the UCI machine learning repository. We use only real datasets based on the observation in [22] that the regularity lemma does not perform well with synthetic datasets, as the regularity lemma relies on the somewhat random distribution of edges in real-world datasets. The 15 datasets are taken from different domains, and vary significantly in dataset size (from 106 to 11000), in data dimension (from 4 to 10000) and in number of clusters (from 2 to 100). We believe these significant diversities in the characteristics of datasets help make the experimental results and conclusions more reliable. The characteristics of these datasets are shown in Table 1, where we denote the dataset size by NP, data dimension by ND, and number of clusters by NC.

In graph-based clustering, we adopt the dominant set (DSet) algorithm [19] and spectral clustering (SPC) in our experiments. Different from the well-known SPC algorithm, the DSet algorithm defines a cluster as a maximal subset of data points with internal coherency and extracts clusters sequentially to accomplish the clustering. While these two algorithms are based on different principles, our experiments show that the influences of parameters on clustering results are similar with both algorithms. Therefore we expect the conclusions in this paper are also applicable to other graph-based algorithms [24].

In the following we firstly study the influence of these parameters ε, b and ϵ on the clustering results. After that, we compare the enhanced DSet and SPC algorithms to the original ones. Finally, we compare the enhanced DSet and SPC algorithms to some recent clustering algorithms.

4.1 Sensitiveness to Parameters

The regular partitioning involves three parameters, which have a significant influence on the clustering results. The work in [24] didn't provide much details on the setting of parameters. In [22] the authors tested some values of ε and b in experiments, but they didn't study the influence of these parameters on the clustering results. Consequently, no instructions on how to determine parameters are provided. In this paper we intend to study the influence of three parameters on clustering results extensively. As a result, we provide some insights on the selection of parameters, going a step further towards the practical application of the regularity lemma in clustering.

In the three parameters of regular partitioning, ε stands for the portion of a class of points used in checking pair regularity. As defined in Definition 1, if $|d(X,Y) - d(A,B)| < \varepsilon$ holds for every $X \subset A$ and $Y \subset B$ such that $|X| > \varepsilon|A|$ and $|Y| > \varepsilon|B|$, then the pair (A,B) is regarded as an ε-regular pair. Therefore the range of this parameter is $0 < \varepsilon < 1$. From Definition 1 we observe that with a large ε which is close to 1, the subsets X and Y will be close to A and B, respectively. In this case, $d(A,B)$ is close to $d(X,Y)$ and the condition $|d(X,Y) - d(A,B)| < \varepsilon$ holds trivially. In other words, a large ε means a relaxed constraint on the regularity of a regular pair, which degrades the clustering results, as shown in the experiments of [22]. Therefore the testing values of the parameter ε are selected to be 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6.

With the original graph G and corresponding reduced graph R, the parameter $\epsilon = \max(\frac{|R|}{|G|})$ denotes the maximum allowed compression ratio, with $0 < \epsilon < 1$. As we intend to reduce the computation load, we expect the reduced graph to be as small as possible. Meanwhile, obtaining a large R itself is computationally expensive, and in experiments we found that $\epsilon = 0.2$ already results in a very large computation load. Therefore in experiments we test only small values 0.02, 0.05, 0.1 and 0.2.

The parameter b in Step 1 of the regular partitioning in Sect. 3.1 denotes the initial number of classes, i.e., the initial cardinality of the reduced graph R. This parameter is set as small values 2 to 7 in [22]. As the cardinality of R is increased in each iteration, we are interested to see if a large initial cardinality b help accelerate the iteration by skipping the first some iterations with smaller cardinalities. Therefore in experiments we test b with both small and large values, including 4, 8, 16, 32, 64, 128, 256, 512 and 1024, with the constraint that $b < |G|$.

We firstly find the best values of the three parameters for a dataset by grid search, and then show how clustering results vary w.r.t. a parameter by fixing the other two parameters. With the enhanced DSet algorithm, we report the clustering results from different parameters in Fig. 1, where the clustering results are measured by NMI (normalized mutual information).

The top row of Fig. 1 shows that when the parameter ε rises, the running time declines in general. This observation is consistent with our analysis that it is relatively easy to achieve regular partitions with a large ε. However, we also observe that the increase of ε degrades the clustering results with some datasets, and ε seems to have little influence on the results with other datasets, with the

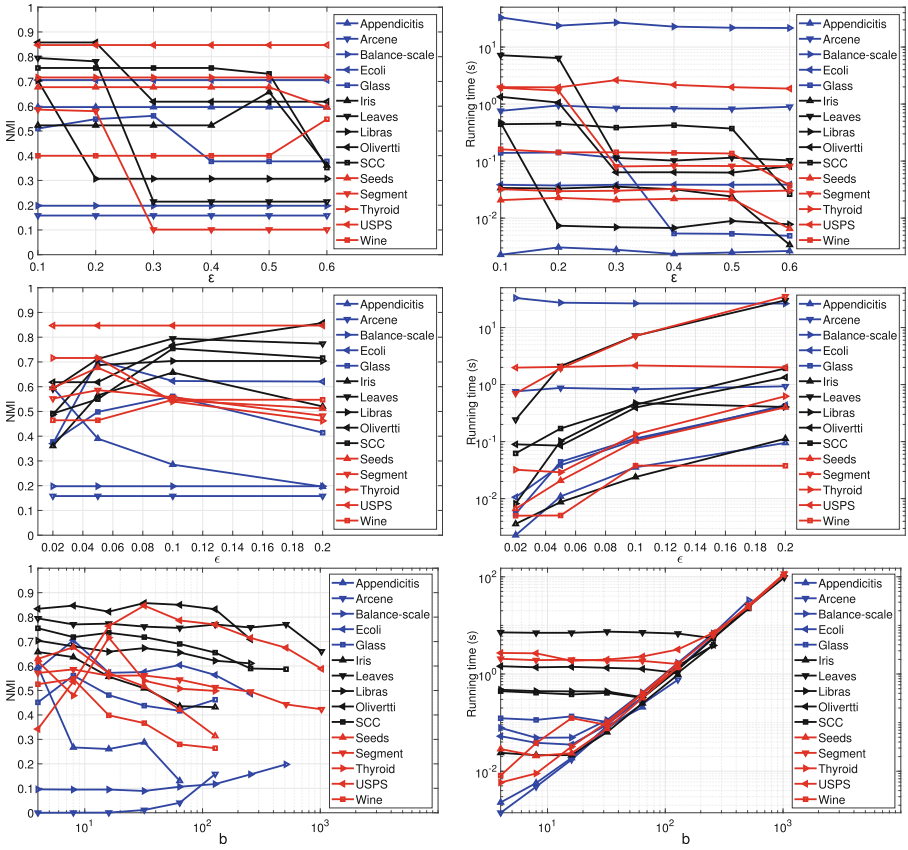


Fig. 1. Performance of the enhanced DSet algorithm, with different parameters.

Wine dataset as the only exception. In summary, although large ε help accelerate regular partitioning, the best clustering quality is usually obtained with small ε's, e.g., 0.1 and 0.2.

As to the observation that a large ε doesn't result in good results, our explanation is as follows. The regular partitioning requires a pair of classes to be ε-regular, and a regular pair requires that for every $X \subset A$ and $Y \subset B$ with $|X| > \varepsilon|A|$ and $|Y| > \varepsilon|B|$, we have

$$|d(X,Y) - d(A,B)| < \varepsilon. \tag{3}$$

On one hand, the large ε means the large right side of Eq. (3), making the inequality easy to hold. On the other hand, since $|X| > \varepsilon|A|$ and $|Y| > \varepsilon|B|$, a large ε means that X is more close to A and Y is more close to B. This further results in a small value in the left side of Eq. (3), making the inequality easy to hold. Both these two aspects show that a large ε reduces the difficulty in achieving regular pairs and therefore accelerates regular partitioning. However, a

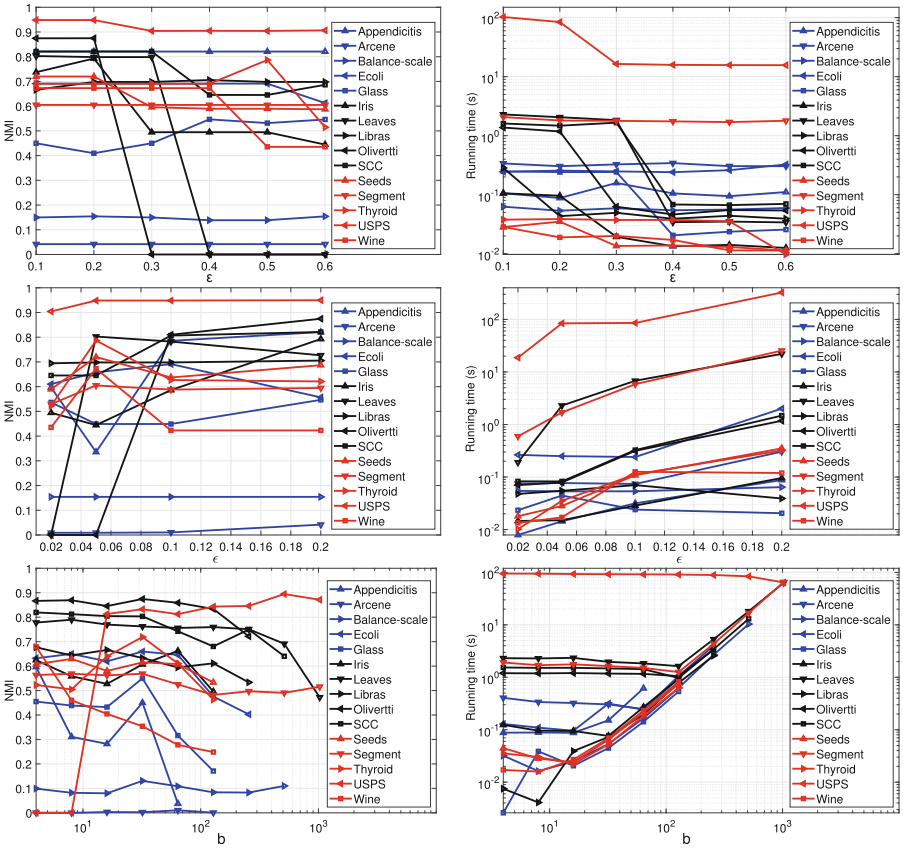


Fig. 2. Performance of the enhanced SPC algorithm.

large ε makes X similar to A and Y similar to B, respectively. While this makes the condition in Eq. (3) easy to satisfy, the relaxed condition also means less uniform distribution of edges between a regular pair. As a result, the reduced graph becomes a less accurate representation of the original graph, and the clustering results are worsened.

In the middle row of Fig. 1, a larger ϵ typically results in a longer running time. Our explanation is as follows. With a large ϵ, the allowed number of classes is large. A provable regular partition usually takes a very large amount of iterations to reach, and in our experiments the iteration are terminated by the upper bound in most cases. Therefore a large ϵ usually means more iterations and more running time to obtain the partition. Meanwhile, we observe that a larger ϵ results in better and worse results on different datasets, and little variance to the results on the remaining ones. In our opinion, these complex behaviors should be attributed to the variances in data distribution of different

real datasets. Generally, if we consider both the running time and clustering results, it seems appropriate to adopt medium ϵ's, e.g., 0.05 and 0.1.

As to the parameter b, we have tested both very small and very large values, with the aim to see if it is possible to accelerate the iteration with a large b. The bottom row of Fig. 1 shows that the running time does decrease slightly with the increase of b, on condition that b is rather small compared with the graph cardinality. After that, the increase of b is usually accompanied by the increase of running time. We try to explain this observation as follows. In the case that b is small compared with the graph cardinality, the increase of b is able to skip the first some iterations and therefore reduce the running time slightly. The regularity optimization skipped in this process will be compensated for in subsequent iterations. With a large b, however, the running time saved in skipping the first some iterations will become smaller than that increased in subsequent iterations, which are with much larger k and take much more running time. As to the clustering quality, we observe that large b's roughly decrease the clustering quality, except for the large dataset USPS. Our explanation is that a large b means many iterations skipped, and the regularity optimization skipped is too large to be compensated for in subsequent iterations. Considering both the running time and clustering results, we recommend to use a small b below 10. Meanwhile, it seems appropriate to select a large b for large graphs, as the best result on the USPS datasets is obtained at $b = 32$.

With the enhanced SPC algorithm, we report the clustering results of different parameters in Fig. 2. By comparing between Fig. 1 and Fig. 2, we observe that our discussions with the DSet algorithm are also applicable to the SPC algorithm. In other words, the influences of parameters with both algorithms are similar, and the best parameter options of the two algorithms are also similar. Therefore we skip the detailed discussions of Fig. 2.

4.2 Comparing with Original Algorithms

To demonstrate if the regularity lemma really improves graph-based clustering, we compare the original DSet algorithm to the enhanced one in Fig. 3. It is shown that the enhanced version generates better results on 14 out of the 15 datasets with both the NMI and RI (Rand Index) criteria, and both algorithm performs very similar on the remaining 1 dataset (Arcene with the NMI criterion or Leaves with the RI criterion). Meanwhile, the enhanced version saves running time compared with the original one on 13 out of the 15 datasets. To conclude, our experiments show that the regularity lemma is able to improve the clustering results of the DSet algorithm and saves the running time simultaneously.

We compare the original SPC algorithm to the enhanced one in Fig. 4. It can be observed that the enhanced SPC algorithm performs better than the original one on 14 datasets, and very similar to the latter one on the remaining 1 dataset (Iris with the NMI criterion or Arcene with the RI criterion). In computation efficiency, the enhanced version takes less running time than the original one on 9 datasets. In conclusion, the SPC algorithm can be enhanced based

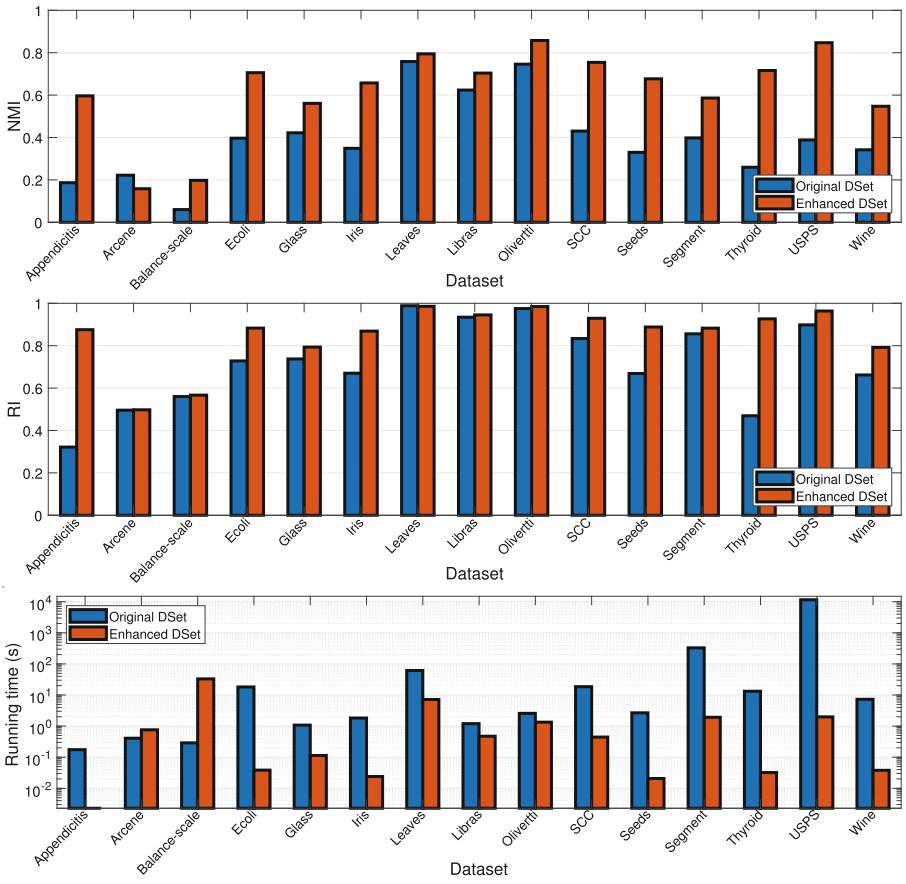


Fig. 3. Comparison between the original and enhanced and DSet algorithm. The enhanced version generates better results with less running time on the majority of datasets.

on the regularity lemma in both clustering quality and computation efficiency simultaneously.

In the following we discuss why the regularity lemma is able to improve the DSet and SPC algorithms to generate better results with less running time. Since the reduced graph is usually much smaller than the original one, clustering on the reduced graph saves running time naturally. With our modifications to regular partitioning and the limit set by the parameter ϵ, the regular partitioning won't take much running time. These two factors together enable the enhanced DSet and SPC algorithms to take less running time than the original ones on the majority of datasets.

As to why the enhanced DSet and SPC algorithms generate better results than the original ones, our explanation is as follows. In a regular partition, each

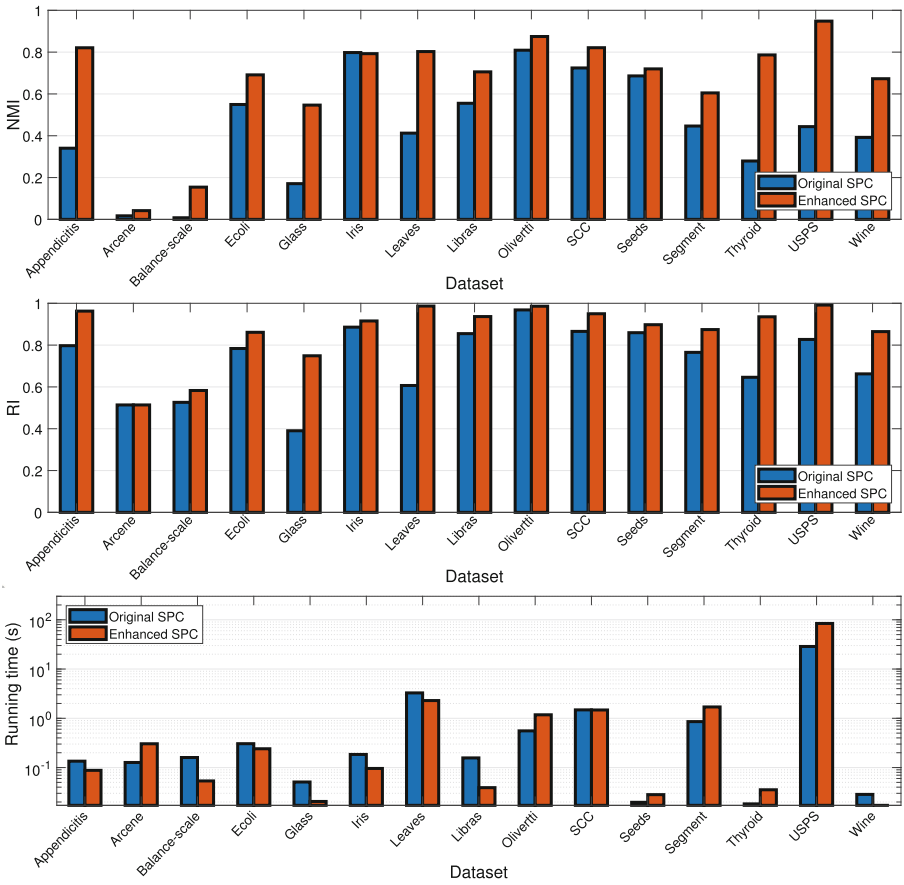


Fig. 4. Comparison between the original and enhanced SPC algorithm. The enhanced version generates better results with less running time on the majority of datasets.

pair of classes behave like a regular pair, and the edges between two classes are distributed quite uniformly. This special distribution of edges allows us to treat a class as a supernode, and then derive the edge weight between two such supernodes based on the edges between the two corresponding classes, thereby generating the reduced graph. In a sense, we can regard the reduced graph as being obtained by sampling of edges or of larger structures, in contrast to ordinary sampling of nodes. In this way, the reduced graph removes many repetitive and redundant structures of the original graph, thereby preserving the key structure information in the original graph in a compressed, compact manner. By simplifying the overall data distribution, this compact representation is beneficial to distinguishing between clusters. Consequently, the small reduced graph is able to generate better clustering results than the original graph.

Table 2. Clustering results (NMI) comparison on 15 datasets.

	MDPC+	LDPSC	DPC-FSC	FHC-LPD	DPCSA	DenMune	3W-DPET	E-DSet	E-SPC
Appendicitis	0.22	0.20	0.05	0.26	0.26	0.28	0.00	0.60	**0.82**
Arcene	0.07	0.04	0.00	0.02	0.05	**0.21**	0.00	0.16	0.04
Balance-scale	**0.23**	0.03	0.02	0.01	0.13	0.21	0.08	0.20	0.15
Ecoli	0.61	0.60	0.55	0.59	0.67	0.68	0.61	**0.71**	0.69
Glass	0.26	0.36	0.03	0.30	0.34	0.43	0.37	**0.56**	0.55
Iris	0.71	0.87	0.03	0.65	**0.89**	0.80	0.76	0.66	0.79
Leaves	0.38	0.73	0.65	0.52	0.69	0.71	0.72	0.79	**0.80**
Libras	0.42	0.65	0.07	0.63	0.64	0.67	0.49	0.70	**0.71**
Olivertti	0.41	0.91	0.90	0.84	**0.92**	0.88	0.81	0.86	0.87
SCC	0.77	0.81	0.81	0.80	0.79	**0.85**	0.71	0.75	0.82
Seeds	0.66	0.68	0.71	0.66	0.67	**0.73**	0.63	0.68	0.72
Segment	0.66	0.65	0.01	0.69	**0.70**	0.68	0.00	0.59	0.60
Thyroid	0.17	0.22	0.11	0.27	0.30	0.53	0.44	0.72	**0.79**
USPS	0.39	0.66	0.38	0.62	0.56	0.81	0.45	0.85	**0.95**
Wine	0.58	0.30	**0.84**	0.56	0.75	0.47	0.82	0.55	0.67
mean	0.44	0.52	0.34	0.49	0.56	0.60	0.46	0.62	**0.67**

Table 3. Clustering results (RI) comparison on 15 datasets.

	MDPC+	LDPSC	DPC-FSC	FHC-LPD	DPCSA	DenMune	3W-DPET	E-DSet	E-SPC
Appendicitis	0.75	0.67	0.50	0.77	0.77	0.77	0.68	0.88	**0.96**
Arcene	**0.54**	0.53	0.50	0.51	0.53	**0.54**	0.50	0.50	0.51
Balance-scale	0.59	0.54	0.44	0.43	**0.60**	0.59	0.55	0.57	0.58
Ecoli	0.83	0.80	0.77	0.81	0.87	**0.88**	0.82	**0.88**	0.86
Glass	0.50	0.71	0.27	0.70	0.62	0.76	0.68	**0.79**	0.75
Iris	0.77	0.95	0.34	0.73	**0.96**	0.89	0.88	0.87	0.92
Leaves	0.64	0.98	0.93	0.87	0.95	0.98	0.98	**0.99**	**0.99**
Libras	0.78	0.91	0.13	0.91	0.90	0.93	0.89	**0.95**	0.94
Olivertti	0.63	**0.99**	0.98	0.96	**0.99**	0.98	0.98	0.98	**0.99**
SCC	0.90	0.87	0.91	0.87	0.90	0.92	0.87	0.93	**0.95**
Seeds	0.87	0.87	0.88	0.87	0.86	0.89	0.84	0.89	**0.90**
Segment	0.86	0.72	0.15	0.87	0.87	**0.91**	0.14	0.88	0.87
Thyroid	0.58	0.58	0.48	0.58	0.65	0.81	0.74	0.93	**0.94**
USPS	0.70	0.90	0.70	0.89	0.80	0.96	0.83	0.96	**0.99**
Wine	0.71	0.60	**0.93**	0.77	0.88	0.74	**0.93**	0.79	0.86
mean	0.71	0.77	0.59	0.77	0.81	0.84	0.75	0.85	**0.87**

4.3 Comparison with Recent Algorithms

We finally compare the enhanced DSet (E-DSet) and SPC (E-SPC) algorithms to some recent works, including MDPC+ [10], LDP-SC [15], DPC-FSC [14], FHC-LPD [11], DPCSA [27], DenMune [1] and 3W-DPET [28]. We report the comparison in Table 2 and Table 3. For clarity, we highlight the best result(s) on a dataset with bold fonts, and the second-best ones with underline.

In Table 2 we observe that, with the NMI criterion, the enhanced DSet and SPC algorithms perform the best or second-best on 10 out of 15 datasets, and their average results on 15 datasets rank the first and second in all the algorithms. In Table 3 with the RI criterion, these two enhanced algorithms generate the best or second-best results on 11 datasets, and again they rank the first and second in

all the algorithms in average results. These comparisons show that the enhanced DSet and SPC algorithms are able to improve relatively old-fashioned algorithms to outperform recent state-of-the-art works.

5 Conclusions

While Szemerédi's regularity lemma was shown to be promising in improving graph-based clustering in previous works, the parameter tuning problem hasn't been dealt with in our knowledge. To solve this problem, we investigate how the three parameters in regular partitioning impact on the clustering results with extensive experiments on real datasets. We find out appropriate ranges of these parameters, which can be helpful in practical applications. We then provide an explanation of the parameters' behaviors, and discuss why the dominant set and spectral clustering algorithms enhanced with the regularity lemma is able to generate better results than the original ones with less running time. Finally, experiments show that the enhanced dominant set and spectral clustering algorithms outperforms some recent works, which further shows that the regularity lemma is promising in big data analysis.

Although our work goes a step forward in applying the regularity lemma in real-world tasks, there are still some problems to be solved. First, the regular partitioning process is still time-consuming. Second, it is still not clear how the approximation degree of the regular partition influences the clustering results. In the future we will work in these two directions to generalize the application of the regularity lemma in real-world tasks further.

References

1. Abbas, M., El-Zoghabi, A., Shoukry, A.: DenMune: density peak based clustering using mutual nearest neighbors. Pattern Recogn. **109**, 107589 (2021)
2. Alon, N., Duke, R.A., Lefmann, H., Rödl, V., Yuster, R.: The algorithmic aspects of the regularity lemma. J. Algorithms **16**(1), 80–109 (1994)
3. Brendan, J.F., Delbert, D.: Clustering by passing messages between data points. Science **315**, 972–976 (2007)
4. Curado, M., Escolano, F., Hancock, E., Nourbakhsh, F., Pelillo, M.: Similarity analysis from limiting quantum walks. In: Third International Workshop on Similarity-Based Pattern Recognition, pp. 38–53 (2015)
5. Fiorucci, M., Pelosin, F., Marcello: Separating structure from noise in large graphs using the regularity lemma. Pattern Recogn. **98**, 107070 (2020)
6. Fiorucci, M., Torcinovich, A., Curado, M., Escolano, F., Pelillo, M.: On the interplay between strong regularity and graph densification. In: The 11th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition, pp. 165–174 (2017)
7. Fischer, E., Matsliah, A., Shapira, A.: Approximate hypergraph partitioning and applications. In: 48th Annual IEEE Symposium on Foundations of Computer Science, pp. 579–589 (2007)
8. Fowlkes, C.C., Belongie, S.J., Chung, F.R.K., Malik, J.: Spectral grouping using the nyström method. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 214–225 (2004)

9. Frieze, A.M., Kannan, R.: A simple algorithm for constructing szemerédi s regularity partition. Electron. J. Combin. **6**, 1–7 (1999)
10. Guan, J., Li, S., He, X., Chen, J.: Clustering by fast detection of main density peaks within a peak digraph. Inf. Sci. **628**, 504–521 (2023)
11. Guan, J., Li, S., He, X., Zhu, J., Chen, J.: Fast hierarchical clustering of local density peaks via an association degree transfer method. Neurocomputing **455**, 401–418 (2021)
12. Hou, J., Ge, J., Yuan, H.: Efficient affinity propagation clustering based on szemerédi's regularity lemma. In: The 17th International Conference on Knowledge Science, Engineering and Management, pp. 449–461 (2024)
13. Komlós, J., Shokoufandeh, A., Simonovits, M., Szemerédi, E.: The regularity lemma and its applications in graph theory. In: Khosrovshahi, G.B., Shokoufandeh, A., Shokrollahi, A. (eds.) Theoretical Aspects of Computer Sciences: Advanced Lectures, pp. 84–112. Springer, Berlin (2002). https://doi.org/10.1007/3-540-45878-6_3
14. Li, Y., Sun, L., Tang, Y.: DPC-FSC: an approach of fuzzy semantic cells to density peaks clustering. Inf. Sci. **616**, 88–107 (2022)
15. Long, Z., Gao, Y., Meng, H., Yao, Y., Li, T.: Clustering based on local density peaks and graph cut. Inf. Sci. **600**, 263–286 (2022)
16. Lovász, L., Szegedy, B.: Szemerédi's lemma for the analyst. Geom. Funct. Anal. **17**(1), 252–270 (2007)
17. Nourbakhsh, F.: Algorithms for graph compression: Theory and experiments. Ph.D. thesis, Ca' Foscari University of Venice (2015)
18. Pavan, M., Pelillo, M.: Efficient out-of-sample extension of dominant-set clusters. In: Neural Information Processing Systems, NIPS 2004, pp. 1057–1064 (2004)
19. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. IEEE Trans. Pattern Anal. Mach. Intell. **29**(1), 167–172 (2007)
20. Pelillo, M., Elezi, I., Fiorucci, M.: Revealing structure in large graphs: Szeremédi's regularity lemma and its used in pattern recognition. Pattern Recogn. Lett. **87**, 4–11 (2017)
21. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.M.: Optimal cluster preserving embedding of nonmetric proximity data. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1540–1551 (2003)
22. Sárközy, G., Song, F., Szeremédi, E., Trivedi, S.: A practical regularity partitioning algorithm and its applications in clustering. arXiv: 1209.6540v1 (2012)
23. Scott, A.: Szemerédi's regularity lemma for matrices and sparse graphs. Comb. Probab. Comput. **20**(11), 455–466 (2011)
24. Sperotto, A., Pelillo, M.: Szemerédi's regularity lemma and its applications to pairwise clustering and segmentation. In: The 6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 13–27 (2007)
25. Szemerédi, E.: Regular partitions of graphs. In: Colloques Internationaux CNRS 260-Problèmes Combinatoires et Théorie des Graphes, pp. 399–401 (1976)
26. Tao, T.: Szemerédi's regularity lemma revisited. Contrib. Discret. Math. **1**(1), 8–28 (2006)
27. Yu, D., Liu, G., Guo, M., Liu, X., Yao, S.: Density peaks clustering based on weighted local density sequence and nearest neighbor assignment. IEEE Access **7**, 34301–34317 (2019)
28. Yu, H., Chen, L., Yao, J.: A three-way density peak clustering method based on evidence theory. Knowl. Based Syst. **211**, 106532 (2021)

IPD: Scalable Clustering with Incremental Prototypes

Jayasree Saha[1(✉)] and Jayanta Mukhopadhyay[2]

[1] IIIT, Hyderabad, India
jayashree.saha@research.iiit.ac.in
[2] IIT, Kharagpur, India
jay@cse.iitkgp.ac.in

Abstract. *DBSCAN*, a fundamental density-based clustering method, is well-known for its ability to discern clusters of diverse shapes. However, its effectiveness diminishes with the scale of datasets, as the storage and processing of labels become impractical. Conversely, centroid-based algorithms like K-means adeptly handle raw data by assigning them to the nearest centroids, yet struggle with non-spherical clusters. This study aims to amalgamate the strengths of these distinct clustering methodologies. Proposing the Incremental Prototype-based DBSCAN (IPD) algorithm, our study addresses the challenge of identifying clusters with arbitrary shapes within large datasets. IPD not only identifies clusters but also selects representatives for each, with a primary focus on establishing a stability criterion within the prototype space. Experimental results demonstrate the efficacy of the approach, showcasing perfect convergence and scalability, even with extremely large synthetic datasets containing shaped clusters. The code for the IPD algorithm is available at https://github.com/Jayasree-Saha/IPD.

Keywords: *DBSCAN* · Prototype · sampling · Incremental clustering · Stability · large-scale data

1 Introduction

Clustering is the process of grouping similar objects into one cluster and dissimilar objects into separate clusters based on some similarity or dissimilarity functions. It is a well-known unsupervised tool used in several machine learning and data mining applications where ground truth is unavailable [21]. However, the current era of Big Data has introduced new challenges to the existing machine learning and data mining approaches [15,33]. Due to automatic capability of knowledge extraction, clustering becomes obvious choice for handling big data.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78107-0_27.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 421–441, 2025.
https://doi.org/10.1007/978-3-031-78107-0_27

Density based clustering is one of the most popular paradigm in the machine learning and data mining community. Ester et al. [6] introduced Density-Based Spatial Clustering of Applications with Noise (*DBSCAN*). The idea is to group data in the high-density region of the feature space. It requires two parameters: i) scanning radius ϵ, and ii) a density threshold $MinPts$. It has the capability of recognizing clusters on complex manifolds, having arbitrary shapes. It is not limited to identifying only "spherical" clusters as any centroid-based algorithm. However, the effectiveness of DBSCAN with very large datasets is hindered by its computational complexity [4]. While Ester et al. [6] proposed a running time of $\mathcal{O}(nlogn)$ using an appropriate index structure for data in d-dimensional Euclidean spaces, where n is the number of objects, Gan and Tao [9] have recently demonstrated that this claim does not hold for $d > 3$; DBSCAN actually requires at least $\Omega(n^{\frac{4}{3}})$. Yet, algorithms with $\mathcal{O}(n)$ time complexity struggle to scale and handle datasets with millions or billions of entries. Thus, employing DBSCAN with very large datasets remains challenging and necessitates a more efficient solution. Nevertheless, it is intuitive that such large datasets often contain redundancy or repetition of similar features, suggesting the possibility of summarizing them with only a fraction of the complete dataset. In such instances, prototype-based approaches [27,28] are highly suitable. Typically, prototypes are initially generated utilizing the leaders clustering algorithm [12] [13] in linear time, followed by the application of clustering algorithms on these prototypes. Some researchers [17,20] address the challenge of large datasets in the DBSCAN method by reducing the number of neighborhood queries, where each query consumes $\mathcal{O}(n)$ time if no index structure is employed. Since algorithms with $\mathcal{O}(n)$ time complexity still struggle with scalability issues when dealing with large datasets, sampling-based solutions are becoming more attractive. In this case, the complexity of computation can be reduced on the sample space since it reduces individual neighborhood query as well as total number of such queries. In this study, we aim to construct a prototype using a sampling-based method and iteratively delineate the cluster structure. The appropriateness of the prototype is ascertained incrementally. The primary challenges in this stage include i) determining a suitable structural representation for depicting a cluster structure, and ii) validating that the cluster structure derived from the sample space aligns well with the entire dataset. In our study, we develop a stability criterion for assessing the cluster structure within the sample space in an incremental manner. For convex clusters, it is appropriate to have a single cluster representative. However, the system struggles to assign data points to the nearest centroid when dealing with closely spaced, arbitrarily shaped clusters. Tong et al. [26] suggest that boundary points serve as potential candidates for representing a cluster in a prototype-based clustering algorithm. Therefore, they may be suitable for representing arbitrary cluster structures. Our major contributions are as follows:

– We introduce a density based clustering method which is based on the basic principle of *DBSCAN*. It uses a fraction of the whole dataset to produce a suitable cluster structure. Hence, it reduces each query processing time and

the total number of queries. The main benefits of our algorithm is that it identifies cluster structure in terms of cluster representatives. In real time, one may not be interested in knowing the labels of all data points but a few for large scale data. Our strategy makes the algorithm more robust and efficient for handling large scale data in real time scenario.
- We introduce an incremental scheme to identify cluster structures by employing a stability criterion. Our technique relies on the sampled items on each iteration for querying. Hence, it reduces the number of queries.
- Our method introduces a strategy for having multiple representatives for each cluster. It facilitates a new instance to be classified to a cluster having any arbitrary shape using the nearest neighbor rule. This labeling scheme is more robust and accurate compared to the single representative scheme as in partition based methods.

The remainder of the paper is structured as follows: Sect. 2 discusses related works, while Sect. 3 provides a detailed description of our proposed algorithm. Evaluation results are presented in Sect. 4, and we conclude our work in Sect. 5.

2 Related Work

DBSCAN is widely used density based clustering method. It builds clusters by examining neighborhood queries across all data objects, discerning their core attributes, and identifying chains of density-connected objects. We present a concise overview of recent study on DBSCAN in handling large-scale datasets.

Variants of *DBSCAN* for Large Scale Data. The *DBSCAN* algorithm detects arbitrary-shaped clusters based on chosen parameters ϵ and $MinPts$. However, its quadratic time complexity renders it impractical for handling big data. To address this limitation, various variants of *DBSCAN* have emerged over the years, aimed at making it suitable for large-scale and high-dimensional datasets. One such variant, proposed by Tong et al. [26], integrates Scalable Clustering Using Boundary Information (*SCUBI*) into *DBSCAN*. *SCUBI* identifies boundary points in the dataset and groups them into clusters using clustering techniques, with the remaining points assigned cluster labels based on their proximity to the boundary points. Hybrid clustering techniques [27,28] combine prototype selection from the dataset with clustering algorithms. The leaders clustering algorithm [12] is often utilized to derive prototypes efficiently. However, it may not capture density information adequately, leading to approximate solutions. Grid-based approaches [11,32] divide the dataset into grids to facilitate neighbor queries. While efficient, they face challenges with high-dimensional data due to neighbor explosion and redundant distance computations. Improved algorithms, such as those proposed by Bonchoo et al. [3] and Chen et al. [4], address these limitations by employing techniques like bitmap indexing and $\frac{\epsilon}{2}$-norm ball identification to optimize neighbor queries and merging operations. Additionally, other *DBSCAN* variants are tailored for big data processing. AnyDBC [20] and IncAnyDBC [19] utilize active learning and iterative processing to

refine cluster structures efficiently. Some methods leverage Graphics Processing Units (GPUs) [16] or map-reduce algorithms [13,18] to exploit parallel architectures for faster computation. However, challenges such as memory management and scalability remain significant concerns in these approaches.

Notion of Stability in Clustering. Cluster analysis hinges on two critical factors: 1) determining whether data exhibits properties of clustering [1], and 2) evaluating the stability of clustering results [29] Clusterability assesses the extent of inherent clustering within the data, and this evaluation is typically conducted prior to applying clustering algorithms. Conversely, stability analysis occurs post-clustering, focusing on the robustness of the obtained partition. In this study, we concentrate on stability and explore current trends in this area. Stability is commonly quantified by assessing the variation in clustering outcomes under minor data perturbations [2]. Typically, stability is computed by analyzing pairwise similarities or dissimilarities between clusterings obtained from subsampled data. This concept is utilized not only for cluster validity assessment [22] but also for determining the optimal number of clusters associated with a dataset [7] Wang [30] proposes a method to determine the appropriate number of clusters via cross-validation. By randomly dividing the dataset into training and validation sets, clustering models are trained and validated, with the instability metric computed based on discrepancies in cluster assignments. This process is repeated for various cluster numbers, with the optimal number determined by minimizing instability. In our work, we leverage instability to evaluate the validity of cluster structures in the incremental approach. However, devising cluster validation criteria within a sample-based framework [24] poses challenges, particularly in ensuring that clustering results are not artifacts of the sampling process. Furthermore, the stability of clustering solutions tends to improve as the sample size increases. Shamir et al. [25] investigate consistency, central limit, and regularity conditions as general sufficient conditions to guarantee the reliability of clustering stability estimators in large sample regimes.

3 IPD

The core principle of our proposed algorithm *IPD* revolves around creating a prototype that streamlines the processing of large datasets. Leveraging the DBSCAN principle, it demonstrates the ability to identify intricate cluster shapes. Moreover, the algorithm continuously updates the clustering structure in response to incoming data points or changes in the dataset, ensuring its adaptability to real-time dynamic environments. The main objective revolves around stabilizing the clustering structure in prototype space to replicate the clustering structure of the original dataset. The pseudo-code for *IPD* is summarized in Algorithm 1. As *IPD* is built upon the foundation of the DBSCAN method, we have provided fundamental terminology and the primary algorithm in the supplementary materials.

Algorithm 1: *Incremental Prototype based* DBSCAN *(IPD)*

Input: Dataset: X; DBSCAN parameters: ϵ, MinPts; $|S_{prototype}|$: γ; $|S_{inc}|$: β; threshold: τ
Output: Cluster Labels: $\mathbb{C}$, Representatives: $\mathcal{R}$
$LOS \leftarrow \{\text{"unknown"}\}_{i=1}^{n}$, $N_\epsilon \leftarrow \{\{\}\}_{i=1}^{n}$;
$LOC \leftarrow \{\}, LOB \leftarrow \{\}, LON \leftarrow \{\}$, $\Delta \leftarrow \infty$;
$S_{\text{prototype}} \leftarrow$ Sample γ points randomly from X, $X_{rem} \leftarrow X - S_{\text{prototype}}$;
$\eta \leftarrow \text{EIM}(S_{\text{prototype}}, \epsilon, \text{MinPts}, \eta = 1)$;
$nextId \leftarrow \text{DBSCAN}(S_{\text{prototype}}, \epsilon, \eta)$;
$C \leftarrow$ labels of $S_{\text{prototype}}$ assigned by DBSCAN;
$\alpha \leftarrow \text{computeTestSize}(C, n)$;
$S_{\text{test}} \leftarrow$ Sample α points randomly from X_{rem}; $X_{rem} \leftarrow X_{rem} - S_{\text{test}}$;
$\Omega_{\text{test}} \leftarrow$ Obtain Cluster labels for S_{test} as described in Definition 3;
while $\Delta > 0$ *and* $|X_{rem}| > 0$ **do**
  $S_{\text{inc}} \leftarrow$ Sample β points randomly from X_{rem}; $X_{\text{rem}} \leftarrow X_{\text{rem}} - S_{\text{inc}}$;
  if $\eta < MinPts$ **then**
    $\eta \leftarrow \text{EIM}(S_{\text{prototype}}, \epsilon, \text{MinPts}, \eta)$;
    RC(LOC, LOS, η, N_ϵ);
  $nextId \leftarrow \text{incDBSCAN}(S_{\text{inc}}, \epsilon, \eta, nextId)$;
  $\Omega'_{\text{test}} \leftarrow$ Obtain Cluster labels for S_{test} as described in Definition 3;
  $\Delta \leftarrow \text{computeStability}(\Omega'_{\text{test}}, \Omega_{\text{test}})$;
  $\alpha' \leftarrow \text{computeTestSize}(C, n)$;
  if $\alpha' - \alpha > 0$ *and* $\alpha' - \alpha > |X_{rem}|$ **then**
    $S_{\text{test}}^{\text{new}} \leftarrow$ Sample $(\alpha' - \alpha)$ points from X_{rem};
    $S_{\text{test}} \leftarrow S_{\text{test}} \cup S_{\text{test}}^{\text{new}}$; $X_{\text{rem}} \leftarrow X_{\text{rem}} - S_{\text{test}}^{\text{new}}$;
    Compute $\mathcal{R}$ from C' as defined in Definition 5;
    $\Omega_{\text{test}} \leftarrow$ Obtain Cluster labels for S_{test} as described in Definition 3;
  $S_{\text{prototype}} \leftarrow S_{\text{prototype}} \cup S_{\text{inc}}$;
  $C' \leftarrow$ labels of $S_{\text{prototype}}$ assigned by incDBSCAN;
  if $\Delta == 0$ *and* $\eta < MinPts$ **then**
    $\Delta \leftarrow 1$, $\eta \leftarrow$ MinPts;
    RC(LOC, LOS, η, N_ϵ);
Compute $\mathcal{R}$ from C' as defined in Definition 5;
$\mathcal{C} \leftarrow$ Each point of X_{rem} is labeled with the label of nearest representative $r \in \mathcal{R}$;
$\mathbb{C} \leftarrow \mathcal{C} \cup C'$;

3.1 Prototype Creation

We first choose a sample ($S_{prototype}$) from the list of unprocessed points in the original dataset as described in Definition 1.

Definition 1. *We define a sample $S_{prototype}$ such that $S_{prototype} \subset X$ and each object in $S_{prototype}$ is sampled randomly with i.i.d and without repetition, such that $|S_{prototype}| = \gamma$. The remaining points are $X_{rem} = X - S_{prototype}$. $S_{prototype}$ acts as initial prototype for the clustering*

Then, we apply *D*BSCAN [6] on this sample. i.e., the query of a point only searches ϵ-neighborhood in the sample space but not in the original data space.

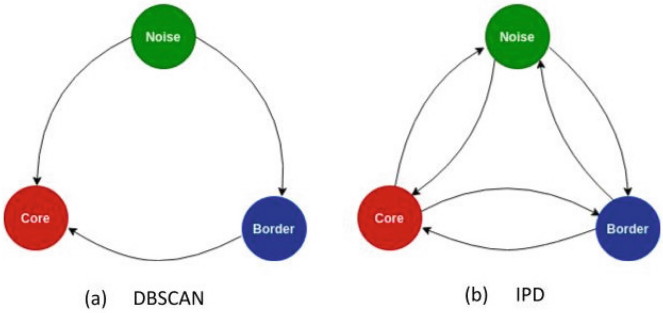


Fig. 1. The state transition diagram of a point

Each point undergoes state transition during the execution of *DBSCAN* as shown in Fig. 1. Intuitively, the size of ϵ-neighborhood of a point in the sample space is small compared to the original data space. Therefore, the value of the parameter $MinPts$ in *DBSCAN* may not be appropriate to identify the "core" point when *DBSCAN* applied in the sample space. To resolve the issue, we introduce parameter η to control the value of $MinPts$ in the sample space. This scheme reduces the computation time for neighborhood queries and builds the structure of a tentative cluster on sample space. In this step, we create a graph $G_p(V, E)$ as defined in Definition 2, based on the outcome of *DBSCAN*.

Definition 2. *We define a graph* $G_{prototype} = (V, E)$ *such that* $v \in V$ *represents a point in* $S_{prototype}$*. We have tagged each vertices with any of three states: "core", "border", and "noise". These states are decided when* DBSCAN *is applied on* $S_{prototype}$*. For every* $e(u, v) \in E$*,* u *is in* ϵ*-neighborhood of* v*. We assign "volatile-yes" (v-yes), "volatile-weak"(v-weak), and "volatile-no"(v-no) state to an edge* $e(u, v)$ *if* u *and* v *are both core, if only one of them is core, and both are not core, respectively. Each maximal set of "volatile-yes" and "volatile-weak" connected vertices forms a cluster.*

We store the neighbors for each point m (a vertex in $G_{\text{prototype}}$) , denoted as $N_\epsilon(m)$, for determining the core property of m. This data structure is extremely useful in the incremental step to avoid unnecessary queries of processed points. Since the state of each vertex v in $G_{\text{prototype}}$ may change in the subsequent steps, a few lists throughout the algorithm are maintained. These lists are called list of *state* (LOS), list of *core* (LOC) , list of *border* (LOB), and list of *noise* (LON). LOS denotes the current state of a point. The length of the LOS is n. This list maintains the current state of a point. They are labeled to "unknown" state at the beginning of the algorithm. *DBSCAN* changes the state of a few points

(points present in $S_{prototype}$) which is reflected in the *State* list. *LOC, LOB*, and *LON* store points which are currently core, border and noise respectively. These are temporary lists whose sizes change several times in the life span of *IPD*.

3.2 Incremental Processing

In this step, we randomly sample β points from the remaining dataset. We update the existing cluster structure while processing them.

Definition 3. *We define sample S_{inc} where each object is sampled randomly with i.i.d and without repetition from X_{rem} such that $|S_{inc}| = \beta$. We modify $G_{prototype} = (V, E)$ to $G'_{prototype}(V', E')$ such that $V' = V \cup V_1$, where $v \in V_1$ represents a point in S_{inc} and $E' = E \cup E_1$, where $e(u, v) \in E_1$ such that $u \in V'$, $v \in V_1$ and $dist(u, v) <= \epsilon$. Also, we allocate state to edges as described in Definition 10.*

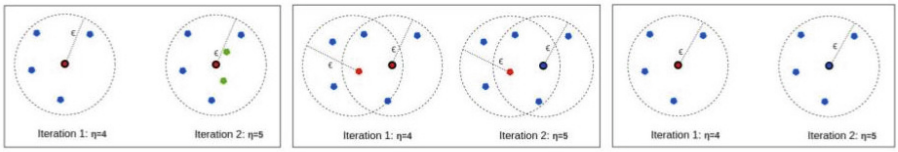


Fig. 2. Transition of the core property of a point outlined with black color. Circle represents ϵ-neighborhood area, blue and green points are non-core points, and red is the core point. Green points are added in the second iteration of incremental processing. (a) depicts the scenario when new points are added in the neighborhood of a core point in the next iteration. (b) and (c) depict the situation when there is no increment in the neighborhood of a core point in the next iteration. (Color figure online)

Since we add more points into the prototype, the size of the ϵ-neighborhood of a point in the sample space may increase. Hence, we increase the value of η successively until $\eta = MinPts$. With this scheme, the core property of a point incurs several transitions as depicted in Fig. 2.

When $MinPts$ is large, the number of iterations in incremental processing increases. To reduce such complexity, we sampled a fraction of the prototype. We observe the number of neighbors within ϵ of the sampled data points and take their mean. We use this number to increment of η till $\eta < MinPts$. Otherwise, we increase η by 1. In the incremental step merging of clusters can occur along with the addition of new cluster. If the new point is a core point and its neighborhood contains points that belong to different clusters, then algorithm triggers merging of those clusters. **Re-evaluation of "core" property.** At every iteration, we increment the value of η that judges the "core" property. Therefore, core property needs to be re-evaluated before processing S_{inc}. We have detailed the process in the supplementary materials.

Algorithm 2: *incDBSCAN*

Input : Unprocessed Sample: S_{inc}
Parameters for incremental DBSCAN: $\epsilon, MinPts$
New Cluster Id: $\hat{k}$
Output: Cluster Labels : $\mathbb{C}$, List : *State*, Next Cluster Id: $\hat{k}$
Lists: *state* , *core*, *border*, *noise*;
for $p \in S_{inc}$ **do**
  if *p is not processed* **then**
    $Found, k \leftarrow incExpandCluster(p, \epsilon, MinPts, \hat{k})$;
  if *(Found == True) and* $(k == \hat{k})$ **then**
    $\hat{k} \leftarrow \hat{k} + 1$;
return $\hat{k}$;

Lemma 1. *With the increment of η, if the state of a vertex u looses "core" property, it may induce many state changes of points in its ϵ-neighborhood.*

Proof. Let u be chosen as a "core" point when $\eta = t$ and $|N_\epsilon(u)| = t$. In the next iteration η becomes $t + r$ ($r \geq 1$) and u becomes either "border" or "noise". $\exists v \in N_\epsilon(u)$ whose state is "border" and u is the only "core" point in its ϵ-neighborhood. i.e., $|N_\epsilon(v)| < t$ and there is no "core" point in the ϵ-neighborhood of v. Therefore, v becomes a "noise". Each core $\rightarrow$ border transition may induce border $\rightarrow$ noise transition. □

Lemma 2. *There is no further state changes for border and noise points with the increment of η unlike core points.*

Proof. Let u be chosen as a "border" point when $\eta = t$ and $|N_\epsilon(u)| = t - m$ where $1 \leq m < t$. When η becomes $t + 1$ in the next iteration, the size of $|N_\epsilon(u)|$ remains the same and the state is also not changed. Therefore, u can not affect others in its ϵ-neighborhood. The same situation holds for a "noise" point. Therefore, there is no further state changes only with the increment η. □

Since the "border" or "noise" point does not induce further state changes, new points can be queried without changing their actual state when η is incremented. However, the size of ϵ-neighborhood of $v \in S_{\text{prototype}}$ may increase while we run queries for every point in S_{inc}. Hence, we can re-verify "border" or "noise" property of v while processing S_{inc}. We have detailed the process in the supplementary materials.

Update the Prototype Graph. During this step, the graph $G_{\text{prototype}}$ is modified so that it includes all merging as a result of new queries prompted by new samples S_{inc} and new clusters within the new samples. At each iteration, *IPD* randomly chooses a set (S_{inc}) of β points from the remaining unprocessed dataset and run queries among their neighbors in $S_{\text{prototype}}$ and S_{inc}. With state changes, $G'_{\text{prototype}}$ may produce variation in cluster structure in $G_{\text{prototype}}$. The cluster structure $\mathbb{C}$ of $G_{\text{prototype}}(V, E)$ is updated to $\mathbb{C}'$ of $G'_{\text{prototype}}(V \cup V_1, E')$ whose formation leads to the following conditions:

1. If $\exists v \in V_1$ such that $\text{dist}(u, v) <= \epsilon$ where $u \in V$ and u has a "core" state and belong to C_m cluster, then v also belongs to C_m cluster.
2. If $\exists v \in V_1$ and v has a "core" state and $\exists u \in N_\epsilon(v)$ such that $u \in V$ and has "noise" state. Then, maximal set of "v-yes" and "v-weak" connected vertices of v is assigned to a new cluster $C_p \notin \mathbb{C}$ and v's state is changed to "border" state.
3. If $\exists v \in V_1$ and ϵ-neighborhood of v contains core vertices u and w such that $u \in C_m$ and $w \in C_n$ where $C_m \neq C_n$, then C_m and C_n are merged to C_m $(m < n)$ and v with its maximal set of "v-yes" and "v-weak" connected vertices (belong to S_{inc}) have been assigned to C_m.

Lemma 3. *If two core points u and v are directly density connected such that $u \in S_{prototype}$, $u \in c_1$, and $v \in S_{inc}$, there exists a path of vertices in $G'_{prototype}$ that connects $\forall w \in c_1$ and v. Similarly, $\forall z \in S_{inc}$ if z is density reachable to v, then z is density connected to u and z belongs to the cluster c_1.*

Proof. Let $u \leftrightarrow x_1 \leftrightarrow x_2 \cdots \leftrightarrow x_m \leftrightarrow w$ be a chain of core points connecting u and w (Definition 6, 10). After performing query on v, if $u \in N_\epsilon(v)$, then $dist(u, v) \leq \epsilon$. Therefore, $v \leftrightarrow u \leftrightarrow x_1 \leftrightarrow x_2 \cdots \leftrightarrow x_n \leftrightarrow w$ i.e., v is density reachable from w. Similarly, $\forall z \in S_{inc}$, if $z \leftrightarrow y_1 \leftrightarrow y_2 \cdots \leftrightarrow y_p \leftrightarrow v$, then after querying on v, z becomes density reachable to u. $z \cdots \leftrightarrow v \leftrightarrow u \cdots \leftrightarrow w$ forms a subset of the maximal set of density connected points. Hence, $z \in c_1$ according to Definition 7. □

Similar to Lemma 3, if $v \in S_{\text{inc}}$ and $u, w \in S_{\text{prototype}}$ but $u \in c_m$ and $w \in c_n$ such that $c_m \neq c_n$, then two clusters merge to a single cluster since $u \leftrightarrow v \leftrightarrow w$. Algorithm 2 and Algorithm 3 togetherly describes the processing of each incremental step.

Re-Verification of Border and Noise. The size of ϵ-neighborhood of any point $v \in S_{prototype}$ may increase while querying each point in S_{inc}. This leads to promotion of state for vertices from *noise* $\rightarrow$ {*border, core*} or *border* $\rightarrow$ *core*.

3.3 Selection of Cluster Representatives

In a partition-based clustering algorithm, clusters are represented by the corresponding cluster centers. Data points are assigned to their nearest cluster centers. However, the scheme of a single representative for a cluster has a serious flaw. Border points of a cluster may get assigned to another nearby cluster. The immediate solution is to choose multiple representatives for each cluster. The problem gets resolved to a great extent when representatives belong to the border area. Hence, we aim to select core points that depict the contour of the arbitrary-shaped clusters. In density-based clustering, core points are the indicator of the high-density region of the cluster. Therefore, we consider core points as the representative of the respective cluster. Since our algorithm develops by processing a fraction of the dataset, the total number of core points is limited.

According to our design of algorithm, test sample S_{test} may be relabeled many times. If the number of representatives is equal to the number of core points, then that quantity could be a bottleneck in the sample space for a large dataset. Therefore, we need to efficiently select very few representatives from the set of core points such that the shape of each cluster could be well predicted by the set of representatives. This in turn reduces the execution time for labeling remaining unprocessed points. Hence, we need to consider the impact of the size of representatives in labeling test sample S_{test} for a very large-scale dataset where n is counted in millions. Hence, we define a heuristic in Definition 4 to sieve a limited number of representatives. The first condition of $\Phi(q)$ selects a few core points which are nearer to the border points as representative. The second condition of $\Phi(q)$ determines the centroid. Definition 5 defines the criterion of a representative.

Definition 4. *Let* $\varrho_q^k = |N_\epsilon(q)|$ *such that* $q \in C_k$, *and* $\varrho_{max}^k = max(\varrho_q^k | q \in C_k)$. *Then, a* Φ *function is defined to choose representatives for each cluster* $C_k \in C$ *as follows:*

$$\Phi(q) = \begin{cases} 1 \text{ if } \frac{\varrho_q^k}{\varrho_{max}^k} \leq \tau \\ 1 \text{ if } \frac{\varrho_q^k}{\varrho_{max}^k} = 1 \\ 0 \text{ Otherwise,} \end{cases} \tag{1}$$

Where τ *is a user defined threshold.*

Definition 5. *We define representatives* $\mathcal{R}$ *for the partition* C *that correspond to graph* H *such that* H *follows Definition 2 and Definition 4.*

$$\forall c \in C, R_c = \{v \mid v \in V' \text{ and } v \text{ has "core" state and } \Phi(v) = 1\} \tag{2}$$

At the end of the incremental step, unprocessed points are assigned to their nearest core points. This scheme has two advantages: 1) it reduces the computation cost of processing the whole dataset as in *DBSCAN*, and 2) the accuracy of labeling border points increases as compared to any centroid-based cluster representative system.

3.4 Stopping Criterion

In this step, we discuss the stopping criterion of the incremental step. We sample S_{test} randomly from the remaining unprocessed points with i.i.d and without repetition. We check whether derived labels of items in S_{test} vary in the consecutive incremental step. For this we measure the change of labels in the pairs of points of S_{test} as discussed in Definition 6.

Definition 6. *We design a stability criterion (in terms of instability) to check whether the structure of clusters of* $G'_{prototype}$ *resembles the structure of clusters of* G_{oracle}. *We create two sets of cluster labels* Ω_{test} *and* Ω'_{test} *for each point in* S_{test}. Ω_{test} *and* Ω'_{test} *are created using the representatives of* $G_{prototype}$ *and*

Algorithm 3: *incExpandCluster*

Input : Point: p, Processed sample: $S_{\text{prototype}}$, Unprocessed sample: S_{inc}; New Cluster Id: $\hat{k}$; Parameters for incremental DBSCAN: $\epsilon, MinPts$; Lists: LOS, LOC, LOB, LON

Output: Cluster Labels : $\mathbb{C}$, List : LOS, Next Cluster Id: $\hat{k}$

$N_\epsilon \leftarrow \text{RangeQuery}(p, \epsilon, S_{\text{prototype}} \cup S_{\text{inc}})$;
if $|N_\epsilon(p)| < MinPts$ **then**
  Determine whether p is a noise or border and update $state(p)$ and $label(p)$ accordingly;
  return $\hat{k}$;
state$(p) \leftarrow$ "core", $LOC \leftarrow LOC \cup p$, $label(p) \leftarrow \hat{k}$;
$seeds = []$;
for $x \in N_\epsilon(p)$ **do**
  if x *is not visited* **then**
    $seeds \leftarrow seeds \cup x$;
  else
    RNB(LON, LOS, η, N_ϵ, x); `// Refer to Algorithm ??`
    $\mathcal{L}_{Merge} \leftarrow \mathcal{L}_{Merge} \cup label(x)$;
for $q \in seeds$ **do**
  $N_\epsilon \leftarrow \text{RangeQuery}(q, \epsilon, S_{\text{prototype}} \cup S_{\text{inc}})$;
  if $|N_\epsilon(q)| > MinPts$ **then**
    $LOS(q) \leftarrow$ "core", $LOC \leftarrow LOC \cup q$, $label(q) \leftarrow \hat{k}$;
    for $x \in N_\epsilon(q)$ **do**
      if x *is unprocessed* **then**
        $seeds \leftarrow seeds \cup x$; `//` x `was not in` $seed$
      else
        RNB(LON, LOS, η, N_ϵ, x); `// Refer to Algorithm ??`
        $\mathcal{L}_{Merge} \leftarrow \mathcal{L}_{Merge} \cup label(x)$;
  else
    $LOS(q) \leftarrow$ "border", $LOB \leftarrow LOB \cup q$, $label(q) \leftarrow \hat{k}$;
$\mathcal{L}_{Merge} \leftarrow \mathcal{L}_{Merge} \cup \hat{k}$;
if $|\mathcal{L}_{Merge}| > 1$ **then**
  Merge all clusters in $\mathcal{L}_{Merge}$;
return $\hat{k} + 1$;

$G'_{prototype}$ *respectively. The labeling scheme uses the following rule: Label* $u \in S_{test}$ *with* $\hat{k} \in \mathbb{C}$ *where* $\mathbb{C}$ *is a partition such that*

$$\hat{k} = \underset{label(v) \in \mathbb{C}}{\text{argmin}} \; dist(u, v) \; where \quad v \in \mathcal{R} \tag{3}$$

However, if there is no cluster structure in the graph, i.e., every point is a "noise", then every point in S_{test} *is labeled with* -1*. The* $\mathbb{C}$ *represents partition*

of either $G_{prototype}$ or $G'_{prototype}$ and $\mathcal{R}$ represents the set of cluster representatives.
Instability measure Δ is defined using following function:

$$\Delta = \frac{1}{\binom{|S_{test}|}{2}} \sum_{1 \le i < j \le |S_{test}|} V_{ij}(\Omega_{test}, \Omega'_{test}) \tag{4}$$

where V_{ij} measures the instability in S_{test}

$$V_{ij}(\Omega_{test}, \Omega'_{test}) = I(I(\Omega_{test}(x_i) = \Omega_{test}(x_j)) + I(\Omega'_{test}(x_i) = \Omega'_{test}(x_j)) = 1) \tag{5}$$

where $I(.)$ is an indicator function.
If x_i and x_j gets same cluster label in Ω_{test} and different cluster label in Ω'_{test} (where $x_i, x_j \in S_{test}$), or vice versa, then $I(\Omega_{test}(x_i) = \Omega_{test}(x_j)) + I(\Omega'_{test}(x_i) = \Omega'_{test}(x_j)) = 1$. Equation 5 represents the XOR *operation. V_{ij} in Eq. 4 enumerates number of pairs in S_{test} which are labeled differently in two consecutive iterations.*

The method initially captures the clustering structure based on items observed within the sampled space, potentially detecting only a subset of clusters. However, with each iteration, the shapes of these clusters are refined. As samples are iteratively added, the prototype accumulates exemplary samples from each existing cluster, allowing it to accurately describe their shapes in the oracle. Once the prototype is well-formed with exemplary samples from the oracle, the algorithm identifies representatives capable of defining each cluster structure in the prototype space. Consequently, test samples cannot change their cluster label. This instability function plays a crucial role in capturing this scenario. Following terms are used in estimating the value of α in the subsequent lemmas and corollaries.
k : Number of clusters.
n : Total number of points.
t : fraction of test samples.
p_i: Prob. of the ith cluster, $i = 1, 2, \ldots, k$.

Lemma 4. *Probability that there exist at least two samples of the ith cluster in the test samples is given by $(nt-1)^2 p_i^2 t^2$.*

Proof. Prob. that a sample of i cluster in the test set: $p_i t$. Number of test samples (α): nt. Prob. that there exist any two or more samples of the i cluster in the test data set:

$$\begin{aligned} & 1 - (1 - p_i t)^{nt} - \binom{nt}{1} (p_i t)(1 - p_i t)^{(nt-1)} \\ &= 1 - (1 - p_i t)^{(nt-1)}(1 + (nt-1)p_i t) \\ &\approx 1 - (1 - (nt-1)P_i t)(1 + (nt-1)p_i t) \\ &= 1 - (1 - (nt-1)^2 p_i^2 t^2 \end{aligned}$$

$$= (nt-1)^2 p_i^2 t^2$$

Q.E.D

Lemma 5. *Probability that for every cluster there exist two or more samples is* $(n't-1)^{2k}(\Pi_{i=1}^{k} p_i^2)t^{2k}$.

Proof. Probability that there exist at least two samples of the ith cluster in the test samples is given by $(n't-1)^2 p_i^2 t^2$.
Hence, the probability that for every cluster there exist two or more samples is $\prod_{i=1}^{k}((n't-1)^2 p_i^2 t^2) = (n't-1)^{2k}(\Pi_{i=1}^{k} p_i^2)t^{2k}$ □

Corollary 1. *The number of test samples for ensuring that there exist at least two samples of each cluster with probability P is obtained from:*

$$(n't-1)^{2k}(\Pi_{i=1}^{k} p_i^2)t^{2k} = P$$

One may put $P = 1$, for the theoretical minimum number. Practically P can be kept high.

Corollary 2. *For uniform probability distribution of clusters $p_i = \frac{1}{k}$. Hence, the theoretical minimum number is given by the solution of the following equation:*

$$\begin{aligned} & n't^2 - t - k = 0 \\ & \Longrightarrow t = \tfrac{1}{2}\left(\tfrac{1}{n'} \pm \sqrt{\tfrac{1}{n'^2} + \tfrac{4k}{n'}}\right) \\ & \approx \tfrac{1}{2n'} + \sqrt{\tfrac{k}{n'}} \end{aligned} \tag{6}$$

This step aims to provide a measure of the 'goodness' of the prototype. This strategy verifies whether the prototype can portray the cluster structure of the original dataset. We use the notion of "clustering stability" to measure the 'goodness' of the prototype. The notion of stability ensures that the cluster structure present in the graph replicates the cluster structure of the dataset X.

Lemma 6. *The algorithm converges when no new cluster emerges.*

Proof. Since we add new samples to the prototype in every iteration, it may reveal new cluster. However, there are two plausible cases. This new cluster may be a part of the existing cluster but it is not density reachable from them at that stage. Again, this cluster may reveal truly a new cluster of the original dataset. However, this condition will stop within finite iteration since number of clusters are constant for the dataset. If new cluster is part of the existing cluster and the condition of density reachability hold for the original dataset with the given ϵ and $MinPts$, then they will be eventually merged within finite iteration. Otherwise, they remain as separate cluster. Addition of new samples can not merge them and eventually they become stable.

In the extreme case, merging may lead to a single cluster. Therefore, the instability measure Δ will be zero. Otherwise, cluster stability is achieved, and addition of new points can not bring new information to the prototype. Hence, Δ becomes zero and iteration stops. □

It is assumed that the cluster structure of $G'_{\text{prototype}}$ replicates the cluster structure of X when the notion of stability is achieved. However, the success of this step depends on the size of $|S_{inc}| = \beta$, and the selected items in S_{test}. If β is very small, then algorithm may converge even if all the clusters have not evolved. However, such a situation ends in having many points as noise if initial size of prototype is also very low. New samples fail to provide variation in the prototype. Hence, Δ reaches to zero. Hence, we have used the principle used in [23] for estimation of sample size to determine β. We provide respective theorems and proofs in the supplementary materials. Again, it is important for S_{test} to have points from every cluster which is the best fit to X. As we are aiming to deal with large datasets, random sampling is the easy solution to create S_{test}. Therefore, the value of $\alpha = |S_{test}|$ is crucial to achieving the requirement. This step aims to provide a measure of the 'goodness' of the prototype. This strategy verifies whether the prototype can portray the cluster structure of the original dataset. We use the notion of "clustering stability" to measure the 'goodness' of the prototype. The notion of stability ensures that the cluster structure present in the graph replicates the cluster structure of the dataset X (Fig. 3).

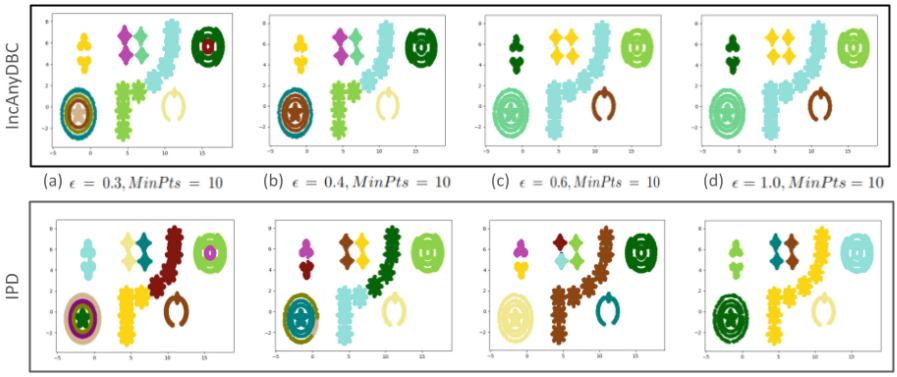


Fig. 3. Clustering results by *IPD* and by *IncAnyDBC* on artificial-flower dataset.

Rectification on the Size of Test Sample for Large Data. Since the test samples are labeled using the nearest neighbor principle, it could be a bottleneck for our algorithm when applied on very large-scale data (instances $>> 10^5$). Similarly, k also can be a bottleneck for the system. The number of clusters present in the prototype can be for computing α. However, the value of k for initial prototype will be very high. Because, small $MinPts$ is used which creates a very large number of groups. Hence, k is fixed to a reasonably high value.

Theorem 1. *For very large data, time complexity of* IPD *is independent of* n.

Proof. DBSCAN requires $\mathcal{O}(\gamma^2)$ for neighborhood queries. Re-evaluation of "core" property consumes $\mathcal{O}(\gamma)$. Each incremental step uses $\mathcal{O}((\beta)^2 + \gamma\beta)$ for

neighborhood queries . It requires $\mathcal{O}(\gamma)$ for processing border and noise list. The termination condition requires $\mathcal{O}(\alpha^2 + \alpha k)$. Hence, time complexity to identify representatives of the clusters in data space requires $\mathcal{O}(\gamma^2 + (\gamma + (\beta^2 + \gamma\beta) + \gamma + \alpha^2 + \alpha k) \times t_{ipd})$, where t_{ipd} is the total number of iterations of the incremental processing step. Usually for large data, $\gamma, \alpha, \beta, k << n$. Hence, time complexity is independent of n. □

Table 1. Performance of IPD for measuring clustering quality using small scale synthetic datasets. '*' indicates other non predominating cluster structure.

Dataset	Parameters		NMI ($\mathcal{L}_Z, \mathcal{L}_{IPD}$)		#clusters (frequency)
	$(\epsilon, MinPts)$	α, β, τ	$Z = \text{dbscan}$	$Z = \text{gt}$	
Aggregation	2, 5	12%, 12%, 0.5	0.96 ± 0.04	0.92 ± 0.04	5 (16%), 6 (42%), 7 (24%)
Compound	2.28, 11	15%, 10%, 0.5	0.84 ± 0.04	0.82 ± 0.02	3 (10%), 4 (64%), 5(20%)
D31	0.65, 5, 0.5	10%, 10%, 0.5	0.82 ± 0.02	0.86 ± 0.02	31(12%), *
K30	2.09, 6, 0.5	15%, 10%, 0.5	0.99 ± 0.01	0.98 ± 0.01	30 (58%), 31 (32%), 32(10%)
t4	10, 7, 0.5	15%, 10%, 0.5	0.67 ± 0.02	-	22 (28%), 24 (18%), *

4 Experiments

We conduct several experiments to evaluate the effectiveness and advantage of the proposed method. We validate our algorithm by performing experiments on synthetic datasets. This helps to understand the aim of our work. The experiments are carried out on a workstation with 128G RAM Centos 64 bit OS, and Python 2.7 programming environment. We have used *Euclidean* distance to measure the similarity between two points in every comparing methods since it is frequently used in the literature. We compare results using normalized mutual information (NMI) [14]. We have detailed a heuristic for parameter estimation for our proposed algorithm in the supplementary materials.

4.1 Datasets

We have chosen dataset with the purpose of analyzing our algorithm on the basis of quality of clusters and capability of handling large-scale data. We consider a few synthetic simulations [5,8,10,31] to test the applicability of our method for detecting arbitrary shaped clusters. They are widely used for cluster analysis in several research papers. Additionally, we have created two synthetic datasets namely Aquanimal and artificial-flower that can be treated as large-scale data to check the scalability of our method. We have also checked its performance on a popular real world datasets, MNIST and PAMPA2. The detailed description of datasets could be found in supplementary material.

Aquanimal. We have created nine shaped clusters in two dimension. Each of them contains approx. $500,000$ data points. The total number of data points are 4.65×10^6. We have used QGIS software to create shapes in .geojson format and fill each shape using uniform distribution in Numpy.

Artificial-Flower. We have used similar approach as aquanimal to create this dataset. We have created 29 shapes using QGIS. However, we introduced a little complexity in the dataset. For example, we have created concentric half circles and we keep their distance small. Also, we add overlapping between two clusters in flower like shapes. We add noise in the set of concentric circles. We add connection between two distant clusters which are having flower like shape. The total number of data points are $320,050$.

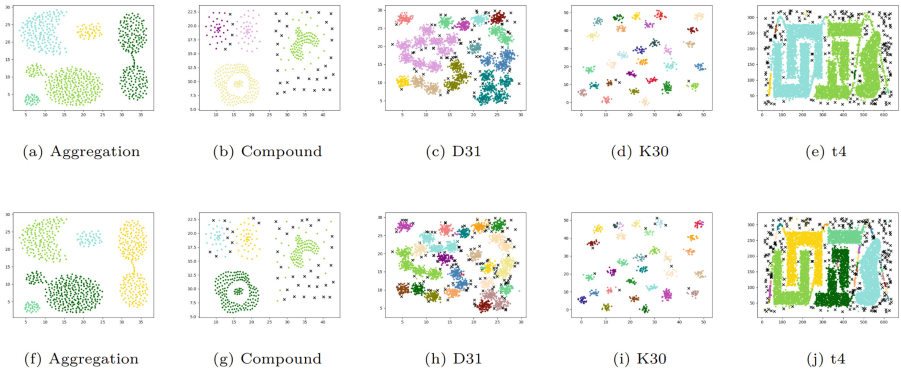
PAMAP2 is a physical activity monitoring dataset having $n = 1,921,431$ and $d = 39$.

MNIST is a digit dataset having $n = 70000$ and $d = 784$.

4.2 Quality of Clustering

We examine quality of clustering of our method with respect to DBSCAN and the ground-truth (gt). We use normalized mutual information (NMI) [14] to compare two labels set $\mathcal{L}_Z, \mathcal{L}_{IPD}$ where $\mathcal{L}$ represents labels of data-points and $\mathcal{L}_Z$ represents either Labels predicted by DBSCAN or the available ground-truth. We have chosen small scale synthetic dataset to examine the performance of IPD in identifying clustering structure. Since *IPD* is a sampling based method, we run the algorithm 50 times and report mean and standard deviation of the metric. Table 1 depicts the results and Fig. 4 provide the 2D visualization of the clustering outcome. It suggests that outcome of *IPD* is not exactly similar to *DBSCAN* for all the datasets. Figure 4 suggests that IPD can capture more detailed clustering structure compared to *DBSCAN* for a given ϵ and $MinPts$. For example, *IPD* captures 22 clusters, and *DBSCAN* captures 12 clusters for D31. However, there are 31 clusters present in an overlapping manner. Similarly, *IPD* captures detailed shaped clusters for t4 compared to DBSCAN. Additionally, Table 1 indicates that *IPD* can identify various rational cluster structures for a dataset. We have demonstrated such occurrences on a synthetic dataset in the supplementary material, along with the effects of other hyperparameters.

Analysis of Clustering for Large Dataset. *IPD* processes a fraction of the dataset to generate representative points for the cluster structures present in the data. Unprocessed points are labeled using the 1-NN rule. In contrast, *IncAnyDBSCAN* processes the whole dataset to determine cluster labels. *IncAnyDBSCAN* and *IPD* have quite different motivations. In IncAnyDBSCAN, the authors attempt to provide results similar to *DBSCAN* for large datasets in reasonable time. In contrast, *IPD* attempts to identify representatives that can accurately reflect the existing clustering structure for the whole dataset. Hence, runtime for *IPD* and *CNAK* reflects the time needed to identify representatives.

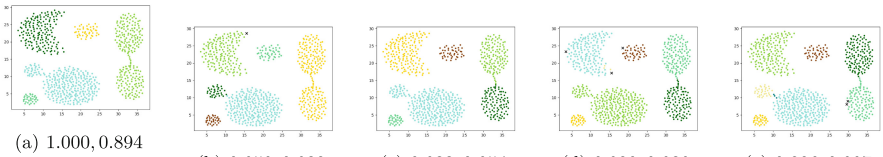


(a) Aggregation (b) Compound (c) D31 (d) K30 (e) t4

(f) Aggregation (g) Compound (h) D31 (i) K30 (j) t4

Fig. 4. Clustering results of DBSCAN ((a)-(e)) and IPD ((f)-(j)) on small scale synthetic dataset

Convergence Analysis. The successful execution depends upon the fact that Δ (refer to Sect. 3.4) should reach zero within fewer iteration. i.e., the method should converge after processing a fraction of the dataset such that it stops within a reasonable time. This also indicates that our method finds a stable cluster structure. Otherwise, the method stops when all data has been processed. To study the convergence, we have measured Δ and detected cluster number for each iteration. We have shown respective plots for Aquanimals and Artificial-flower datasets in supplementary material.

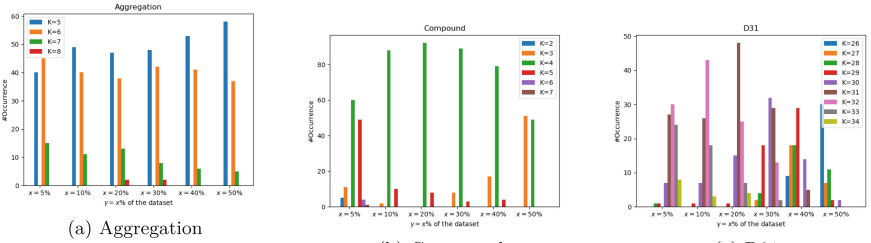


(a) 1.000, 0.894 (b) 0.950, 0.932 (c) 0.933, 0.954 (d) 0.920, 0.939 (e) 0.886, 0.997

Fig. 5. Several clustering structures of the *Aggregation* dataset. (a)-(e) show predictions by IPD. Parameters used: $\epsilon = 2$, $MinPt = 2$, $\gamma = 12\%$ of the dataset, $\beta = 12\%$ of the dataset, and $\tau = 0.5$. (nmi_{dbscan}, nmi_{gt}) are reported in the subcaption where nmi_{dbscan} compares the prediction of IPD and the prediction of DBSCAN for the same ϵ and $MinPts$, and nmi_{gt} compares the prediction of IPD with the ground truth.

4.3 Effect of Randomness on Clustering Structure

Since *IPD* is a sampling-based method, random behavior could be observed. To understand the effect of randomness we have executed IPD several times with the same parameters. We have chosen *Aggregation* dataset to perform such a test and we observe interesting characteristics of our algorithm. We have kept $\epsilon = 2$,



(a) Aggregation (b) Compound (c) D31

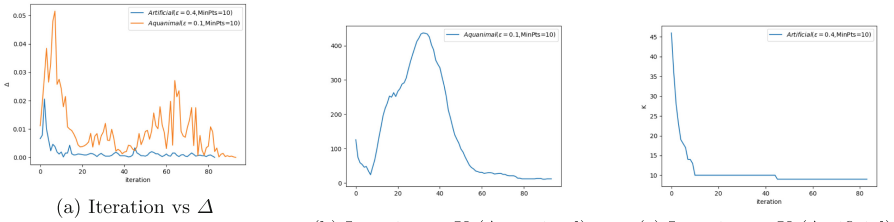
Fig. 6. Effect of γ on clustering result.

$MinPts = 5$, $\gamma = 12\%$ of the dataset, $\beta = 12\%$ of the dataset, and $\tau = 0.5$ for the experiment and run the algorithm for 50 times. We depict a few clustering structure produced by *IPD* in 50 such executions in Fig. 5. We observe that our algorithm produces $K = 5$, $K = 6$ and $K = 7$ for 16%, 42% and 12% times respectively. We have shown normalized mutual information between prediction of *IPD* and *DBSCAN* and prediction of *IPD* and the ground truth in Fig. 5. This phenomenon suggests that our method is effective in capturing multiple suitable clustering structures.

4.4 The Effect of initial Prototype Size γ on IPD

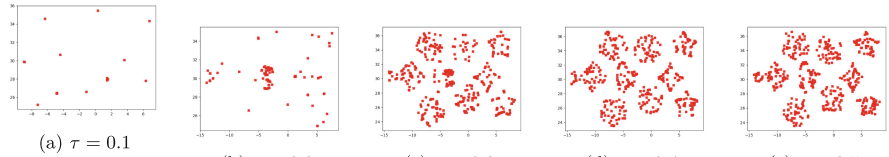
To study the effect of γ, we run our algorithm 100 times on each of the chosen sizes of γ. We have shown the type of cluster structure (in terms of cluster number) obtained for various values of γ in Fig. 6. We have chosen *Aggregation*, *Compound* and *D31* datasets for this experiment. Figure 6a suggests that estimated cluster number $K = 5$ predominates when the initial size of the prototype is large. *DBSCAN* also provides a similar cluster structure. However, with the small size of the initial prototype, several cluster structure is observed. We observe the similar phenomenon for *Compound* dataset in Fig. 6b. On the other hand *D*31 shows an interesting behavior in Fig. 6c. When the initial prototype size is small, the resulting cluster structure contains more than 31 clusters most of the time. But, the resulting cluster structure contains clusters between 26 to 31, when that size is large. Inter-cluster distance is small in D31. With the large size of samples, a few clusters appear as a single cluster. Hence, a small number of clusters predominates. We draw the following observation from this fact: Our method reveals the existence of hierarchy on the dataset on several runs with an initial prototype of small size. The clustering structure of the initial prototype is obtained by *DBSCAN*. Therefore, a large size prototype tends to produce a similar cluster structure as in *DBSCAN*.

Analysis on the Size (α) of Test Dataset. The size of α depends on both the number of clusters k and the size of the dataset n. For a large dataset, the test size becomes significantly large. It causes a large number of computations during the generation of test labels at every iteration. Hence, we use the size



(a) Iteration vs Δ (b) Iteration vs K (Aquanimal) (c) Iteration vs K (Artificial)

Fig. 7. Convergence on large-scale datasets (Aquanimal and Artificial)



(a) $\tau = 0.1$ (b) $\tau = 0.2$ (c) $\tau = 0.3$ (d) $\tau = 0.4$ (e) $\tau = 0.5$

Fig. 8. Selected representative for Aquanimal at several thresholds (τ)

of a subset of the dataset. Here, we use $n = 50,000$. Similarly, K may also be a bottleneck for the system. We use the number of clusters present in the prototype for computing α. We also fix K to a reasonably high value. Here, we use $K = 50$. Although this strategy eliminates the dynamic nature of the test size, it ensures scalability (Fig. 7).

The Effect of Threshold τ to Select Representatives τ is an important parameter that helps to select the number of representatives from each cluster. With the higher value of τ, the number of representatives increases. But, this creates a bottleneck while handling large-scale datasets. ?? depicts that NMI reaches stability at $\tau = 0.3$. NMI does not change significantly with $\tau > 0.3$. However, with increasing τ, run time increases. Figure 8 depicts that with high increasing τ, representatives can draw the contour of the clusters. The number of representatives controls the quality of the cluster. It is a trade-off between the number of representatives and execution time.

5 Conclusion

This paper introduced the prototype-based incremental DBSCAN clustering algorithm, IPD, which aims to select representatives for arbitrarily shaped clusters for large datasets. Our approach emphasizes the significance of real-time cluster representatives. The core strategy of IPD involves creating prototypes, assigning cluster structures, seeking feedback, and iteratively modifying prototypes and cluster structures until convergence is achieved. By selecting a subset of data points to construct a stable cluster structure fitting the original

dataset, IPD minimizes query consumption. However, achieving high-quality clusters relies heavily on selecting appropriate DBSCAN parameters. Our experiments demonstrate that IPD is proficient in capturing multiple cluster structures present within datasets, showcasing its versatility and efficacy in diverse clustering scenarios (Table 2).

Table 2. Runtime comparison on large scale data

Dataset	size		ϵ	Minpts	runtime (in s)			resultant K		
	#data-points,	#features			incAnyDBC	IPD	CNAK	incAnyDBC	IPD	CNAK
Artificial flower	3.2×10^5	2	1	10	175	19	126	6	7	6
Aqua animal	4.65×10^6	2	0.4	10	3141	33	424	7	9	8
PAMAP2	1,921,431	39	200	100	129	116	692	1	1	1
MNIST	70000	768	1000	10	4644	4320	198	32	22	1

References

1. Adolfsson, A., Ackerman, M., Brownstein, N.C.: To cluster, or not to cluster: an analysis of clusterability methods. Pattern Recogn. **88**, 13–26 (2019)
2. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, vol. 2002, 6–17 (2002)
3. Boonchoo, T., Ao, X., Liu, Y., Zhao, W., Zhuang, F., He, Q.: Grid-based DBSCAN: indexing and inference. Pattern Recogn. **90**, 271–284 (2019)
4. Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., Du, J.: Block-DBSCAN: fast clustering for large scale data. Pattern Recogn. **109**, 107624 (2021)
5. Veenman, C.J., Reinders, M.J.T., Backer, E.: A maximum variance cluster algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **24**(9), 1273–1280 (2002)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231. KDD 1996 (1996)
7. Fang, Y., Wang, J.: Selection of the number of clusters via the bootstrap method. Comput. Stat. Data Anal. **56**(3), 468–477 (2012)
8. G. Karypis, E.H. Han, V.K.: Chameleon: a hierarchical 765 clustering algorithm using dynamic modeling. IEEE Trans. Comput. **32**(8), 68–75 (1999)
9. Gan, J., Tao, Y.: DBSCAN revisited: mis-claim, un-fixability, and approximation, pp. 519–530. SIGMOD 2015 (2015)
10. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Trans. Knowl. Discov. Data **1**(1), 4–es (2007)
11. Gunawan, A., de Berg, M.: A faster algorithm for DBSCAN. Master's thesis (2013)
12. Hartigan, J.A.: Clustering Algorithms. Wiley (1975)
13. He, Y., et al.: MR-DBSCAN: an efficient parallel density-based clustering algorithm using MapReduce. In: IEEE 17th International Conference on Parallel and Distributed Systems, pp. 473–480 (2011)

14. Horibe, Y.: Entropy and correlation. Trans. Syst. Man Cybern. **SMC-15**(5), 641–642 (1985)
15. Ianni, M., Masciari, E., Mazzeo, G.M., Mezzanzanica, M., Zaniolo, C.: Fast and effective big data exploration by clustering. Futur. Gener. Comput. Syst. **102**, 84–94 (2020)
16. Loh, W.K., Yu, H.: Fast density-based clustering through dataset partition using graphics processing units. Inf. Sci. **308**, 94–112 (2015)
17. Luchi, D., Rodrigues, A.L., Varejão, F.M.: Sampling approaches for applying DBSCAN to large datasets. Pattern Recogn. Lett. **117**, 90–96 (2019)
18. Lulli, A., Dell'Amico, M., Michiardi, P., Ricci, L.: NG-DBSCAN: scalable density-based clustering for arbitrary data. Proc. VLDB Endow. **10**(3), 157–168 (2016)
19. Mai, S., et al.: Incremental density-based clustering on multicore processors. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3023125
20. Mai, S.T., Assent, I., Storgaard, M.: AnyDBC: an efficient anytime density-based clustering algorithm for very large complex datasets. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1025–1034. KDD 2016 (2016)
21. Perner, P. (ed.): Machine Learning and Data Mining in Pattern Recognition. Springer-Verlag, Berlin Heidelberg (2009). https://doi.org/10.1007/978-3-642-03070-3
22. Roth, V., Lange, T., Braun, M., Buhmann, J.: A resampling approach to cluster validation. In: Compstat, pp. 123–128 (2002)
23. Saha, J., Mukherjee, J.: CNAK: cluster number assisted k-means. Pattern Recogn. **110**, 107625 (2021)
24. Shamir, O., Tishby, N.: Cluster stability for finite samples. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, pp. 1297–1304. NIPS 2007 (2007)
25. Shamir, O., Tishby, N.: On the reliability of clustering stability in the large sample regime. In: Advances in Neural Information Processing Systems, vol. 21 (2009)
26. Tong, Q., Li, X., Yuan, B.: A highly scalable clustering scheme using boundary information. Pattern Recogn. Lett. **89**, 1–7 (2017)
27. Viswanath, P., Pinkesh, R.: L-DBSCAN : a fast hybrid density based clustering method. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 912–915 (2006)
28. Viswanath, P., Suresh Babu, V.: Rough-DBSCAN: a fast hybrid density based clustering method for large data sets. Pattern Recogn. Lett. **30**(16), 1477–1488 (2009)
29. Volkovich, Z., Barzily, Z., Morozensky, L.: A statistical model of cluster stability. Pattern Recogn. **41**(7), 2174–2188 (2008)
30. Wang, J.: Consistent selection of the number of clusters via crossvalidation. Biometrika **97**(4), 893–904 (2010)
31. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **C-20**(1), 68–86 (1971)
32. Zhao, Y., Zhang, C., Shen, Y.D.: Clustering high-dimensional data with low-order neighbors. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), pp. 103–109 (2004). https://doi.org/10.1109/WI.2004.10128
33. Zhou, L., Pan, S., Wang, J., Vasilakos, A.V.: Machine learning on big data: opportunities and challenges. Neurocomputing **237**, 350–361 (2017)

Mitigating the Impact of Noisy Edges on Graph-Based Algorithms via Adversarial Robustness Evaluation

Yongyu Wang(✉) and Xiaotian Zhuang

JD Logistics, Beijing 101111, China
wangyongyu1@jd.com

Abstract. Given that no existing graph construction method can generate a perfect graph for a given dataset, graph-based algorithms are often affected by redundant and erroneous edges present within the constructed graphs. In this paper, we view these noisy edges as adversarial attack and propose to use a spectral adversarial robustness evaluation method to mitigate the impact of noisy edges on the performance of graph-based algorithms. Our method identifies the points that are less vulnerable to noisy edges and leverages only these robust points to perform graph-based algorithms. Our experiments demonstrate that our methodology is highly effective and outperforms state-of-the-art denoising methods by a large margin.

Keywords: Graph · Denoising · Adversarial Robustness

1 Introduction

For many graph-based algorithms, the initial phase entails the construction of a graph from the provided dataset [1]. This graph is structured such that each node corresponds to an individual data point, while the edges delineate the interrelations among these points. Given the intrinsic uniqueness of each dataset, it is impractical to expect a universal graph construction method that can precisely cater to all datasets. Consequently, the graphs generated may contain a significant number of erroneous and superfluous edges, commonly referred to as noisy edges. These noisy edges can profoundly degrade the performance of graph-based algorithms [2,4].

Currently, the most widely used method for graph construction is the k-nearest neighbor (k-NN) graph. In a k-NN graph, each node is connected to its k nearest neighbors. This method possesses a strong capability to capture the local manifold [3], which is why it has become the predominant graph construction technique for the majority of graph-based algorithms. However, k-NN graph has a tendency to include noisy edges [5]. Within datasets, the distribution and characteristics of points are neither uniform nor consistent, rendering the use of a uniform k value for all points imprecise [2]. Furthermore, the metric used to measure the distance between two points is also problematic. Whether it is

Y. Wang and X. Zhuang—Contributed equally and are co-first authors.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 442–452, 2025.
https://doi.org/10.1007/978-3-031-78107-0_28

Euclidean distance, cosine similarity, or any other distance metric, each has its limitations [6]. Consequently, the distances measured are not always accurate. Therefore, the k nearest neighbors in a k-NN graph may not necessarily be the ones that should be connected.

Addressing this issue is extremely challenging, primarily because, for the task of capturing the underlying structure of a dataset using a graph, there is no ground truth solution. Thus, only heuristic methods are available to remove noisy edges from the graph. Among the most representative works in this area is the consensus method proposed by [2], which extracts consensus information from a given k-NN graph. In this method, edges with a consensus value below a certain threshold are pruned. However, this approach, while removing noise, also eliminates a substantial number of non-noisy, useful edges. [4] proposed a spectral framework to detect non-critical, misleading, and superfluous edges in the graph. However, the gains in algorithmic solution quality are still relatively modest.

This paper introduces a novel method aimed at enhancing the noise resilience of graph-based algorithms. Unlike approaches that attempt to remove noisy edges from the graph, our method employs a spectral adversarial robustness evaluation method to identify a small amount of robust nodes that exhibit strong resistance to noise. We then utilize only these robust nodes to complete the graph analysis tasks. Our approach not only enhances the solution quality of graph-based algorithms but can also help to reduce the computational cost of these algorithms. The majority of graph-based algorithms have a time complexity of at least $O(n^2)$, and many are $O(n^3)$, such as spectral clustering algorithms [1], where n is the number of nodes in the graph. Unlike traditional approaches that involve all nodes, our method only requires the robust nodes, thus substantially lowering the computational cost.

The main contributions of this work are as follows:

1. We view noisy edges in graphs as adversarial attacks and propose to use a spectral proactive defense approach to fundamentally address this issue.
2. In contrast to existing approaches that address the issue of noisy edges from the perspective of the edges themselves, our method approaches the problem from the node perspective. Recognizing the inherent challenge in discerning whether a specific edge in a graph is noise or necessary, we propose a solution that focuses on identifying nodes that are not vulnerable to noisy edges. By doing so, we aim to reconstruct a robust dataset that is resilient to the presence of noisy edges.
3. We show that by utilizing only a small number of robust nodes, significant improvements can be achieved in both the accuracy and efficiency for graph-based algorithms.

2 Preliminaries

2.1 Adversarial Attack and Defense

Adversarial attack aims to misled machine learning models by providing deceptive inputs, such as samples with intentional disturbances [7,8], which are commonly known as adversarial examples. [9–11] have demonstrated that machine learning models are often highly vulnerable to adversarial attacks.

To defend against adversarial attacks, many methods have been proposed. These methods can be categorized into two types: reactive defenses and proactive defenses. Reactive defenses concentrate on detecting adversarial examples within the model's inputs, as investigated by [12–14]. In contrast, proactive defenses seek to bolster the robustness of the models, making them less vulnerable to the influence of adversarial examples, such as the methods presented in [15,16].

2.2 Spectral Clustering

Spectral clustering is one of the most representative and widely applied graph-based algorithms. It can often outperform traditional clustering algorithms, such as k-means algorithms, due to its ability to extract structural features of the dataset from the graph representation [1]. There are three common spectral clustering algorithms used in practice, i.e., unnormalized spectral clustering [1] and two normalized spectral clustering methods [20,21]. These algorithms are quite similar, apart from using different graph Laplacians. As shown in Algorithm 1, typical spectral clustering algorithms can be divided into three steps: 1) construct a data graph according to the entire data set, 2) embed all data points into k-dimensional space using eigenvectors of k bottom nonzero eigenvalues of the graph Laplacian, and 3) perform k-means algorithm to partition the embedded data points into k clusters.

3 Method

3.1 Examining Graph-Based Machine Learning Models from the Perspective of Adversarial Attacks

Machine learning models are fundamentally mechanisms that map inputs to outputs via feature transformation. For instance, deep neural networks distill the original feature vectors of data through successive layers. Similarly, algorithms like Support Vector Machine (SVM) and Support Vector Clustering (SVC) utilize kernel functions to map the original feature space of data into a higher-dimensional feature space [22]. Adversarial attacks on machine learning models aim to mislead this mapping process.

In graph-based machine learning algorithms, the graph plays a pivotal role as the algorithm extracts structural information from the graph to transform input data, thereby mapping the inputs to outputs. Therefore, from the perspective

Algorithm 1. Unnormalized Spectral Clustering Algorithm

Input: A data set D with N samples $x_1, ..., x_N \in R^d$, number of clusters k.
Output: Clusters C_1,...,C_k.

1: Construct a graph G from the input data ;
2: Compute the adjacency matrix A_G, and diagonal matrix D_G of graph G;
3: Obtain the unnormalized Laplacian matrix L_G=D_G-A_G;
4: Compute the eigenvectors u_1,...u_k that correspond to the bottom k nonzero eigenvalues of L_G;
5: Construct $U \in \mathbb{R}^{n \times k}$, with k eigenvectors of L_G stored as columns;
6: Perform k-means algorithm to partition the rows of U into k clusters and return the result.

of adversarial attacks, perturbing the graph can disrupt the mapping process of the graph-based machine learning model.

Suppose there exists a ground-truth 'perfect' graph for a graph-based algorithm. In that case, any discrepancies between the actual graph constructed by our graph-building algorithm and the ground-truth graph could be considered as adversarial attacks applied to the ground-truth graph. Therefore, we propose to use adversarial defense methods to manage these noisy edges.

3.2 A Proactive Defense Strategy from the Node Perspective to Mitigate the Impact of Noisy Edges

Given that the ideal connectivity of edges within a graph is perpetually unknown, previous research focused on managing noisy edges directly from the edge perspective has yielded limited success [2,4]. In this paper, we propose to address the issue of noisy edges from the perspective of nodes.

Graph-based machine learning models, much like other machine learning models, predominantly focus on tasks associated with data points, such as classifying these points or predicting values associated with them. Graph-based algorithms distinguish themselves by utilizing the relationships signified by edges to assist in accomplishing tasks that are centered on the nodes. Edges in a graph play a twofold role. While they contain significant structural information that can aid in achieving more accurate data transformations-such as those utilized by graph neural networks to harness the associative information between nodes for improved performance-they can also be detrimental. A multitude of incorrect or superfluous edges can indeed degrade the performance of machine learning models. By enhancing the nodes' resilience to noisy edges, our method can exploit the beneficial edges to boost algorithmic performance without being adversely affected by the harmful ones. In this paper, we propose enhancing the nodes' resilience to noisy edges as a means to achieve improved algorithm performance by utilizing beneficial edges while concurrently mitigating the negative impact of harmful edges, when both coexist within the graph.

3.3 Adversarial Robustness Evaluation

In order to identify the nodes with strong resilience to noisy edges, we first evaluate the robustness of data points to noisy edges.

Inspired by [17,18] proposed that the adversarial robustness of a given machine learning model can be measured by examining the distortion between the manifolds of the input feature space and the output feature space, by leveraging the generalized Courant-Fischer theorem [19]. In this section, we employ this method to evaluate the robustness of each data point in the spectral clustering model. The specific steps are as follows:

- Given data set D with N samples $x_1, ..., x_N \in R^d$ and its number of clusters k, we first construct a k-NN graph G_{input} to capture the data manifold in the original d-dimensional feature space.
- We perform the spectral embedding step in the spectral clustering algorithm to map the data points from the original d-dimensional space into k-dimensional spectral space to obtain data set U with the points in the embedded feature space.
- We construct a k-NN graph G_{output} to capture the data manifold in the embedded k-dimensional feature space.
- Based on the generalized Courant-Fischer theorem [18,19] has further shown that the generalized eigenpairs of $L^{+}_{output}L_{input}$ can be used to estimate the robustness of each point, where L^{+}_{output} denotes the Moore-Penrose pseudoinverse of the graph Laplacian of G_{output} and L_{input} denotes the graph Laplacian matrix of G_{input}. To this end, we construct the following eigensubspace matrix $\mathbf{V}_k \in \mathbb{R}^{N\times k}$:

$$\mathbf{V}_k = \left[\mathbf{v}_1\sqrt{\lambda_1}, \ldots, \mathbf{v}_k\sqrt{\lambda_k}\right], \tag{1}$$

  where $\lambda_1, \lambda_2, \ldots, \lambda_k$ represent the first k largest eigenvalues of $L^{+}_{output}L_{input}$ and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ are the corresponding eigenvectors.
- Finally, a metric called spade score for evaluating the adversarial robustness of a specific node i can be calculated as follows [18]:

$$\text{spade}(i) = \frac{1}{|\mathcal{N}_(i)|}\sum_{j\in\mathcal{N}_(i)} \|\mathbf{V}_k^{\top}\mathbf{e}_{i,j}\|_2^2; \tag{2}$$

  where $j \in \mathcal{N}_(i)$ denotes the j-th neighbor of node i in graph G_{input}, and $\mathcal{N}_(i) \subseteq V$ denotes the node set including all the neighbors of node i, $\mathbf{e}_{i,j} = \mathbf{e}_i - \mathbf{e}_j$, and $\mathbf{e}_o \in \mathbb{R}^N$ denotes the standard basis vector with the i-th element being 1 and others being 0. A larger spade(i) implies that node i is likely more vulnerable to adversarial attacks.

The above procedures are efficient: constructing k-nearest graph can be done within $O(|n|log|n|)$ time [26]; The spade score can be computed in nearly-linear time leveraging recent fast Laplacian solvers [28].

3.4 A Multi-level Algorithm Framework Based on Robust Node Set

We calculate the spade score for all data points and sort them in ascending order. We then select a small number of data points with the lowest spade score, which correspond to the highest robustness, to form a robust subset. After obtaining the robust node set, we perform spectral clustering exclusively on this set to group the robust nodes into k clusters. For each cluster, we calculate its centroid. Subsequently, each non-robust data point is assigned to the cluster whose centroid is closest to it.

4 Experiment

In this section, we apply the proposed method to k-NN graph and use unnormalized spectral clustering to demonstrate its effectiveness. We assess the efficacy of our proposed approach by evaluating its ability to improve solution quality, as well as its capacity to increase the operational efficiency of the algorithm.

4.1 Data Sets

Experiments are performed using the following two real-world benchmark data sets:

- **USPS:** includes $9,298$ images of USPS hand written digits with 256 attributes.
- **MNIST:** the machine learning field's most recognized benchmark, features 60,000 training and 10,000 test images of handwritten digits, each with 784 attributes. We evaluate our methods using its test set.

4.2 Metric

To assess the solution quality of spectral clustering, we use the accuracy metric. It is defined as:

$$ACC = \frac{\sum_{j=1}^{n} \delta(y_i, map(c_i))}{n}, \tag{3}$$

pgwhere n represents the total count of data instances within the dataset, y_i denotes the ground-truth label as provided by the dataset, and C_i signifies the label ascribed by the clustering algorithm. The function $\delta(x, y)$ is a delta function, stipulated as: $\delta(x, y) = 1$ for $x = y$, and $\delta(x, y) = 0$, otherwise. The function $map(\bullet)$ serves as a permutation mapping that correlates each cluster

index c_i with an equivalent ground truth label, a process which can be efficiently accomplished utilizing the Hungarian algorithm [23]. An elevated ACC value is indicative of superior clustering performance.

4.3 Compared Algorithms

We compare our method against both the baseline and the state-of-the-art techniques for handling noisy edges in graphs. The specifics are as follows:

- **k-Nearest Neighbor Graph:** For the value of k in the k-NN graph for the USPS and the MNIST data sets, we use the setting in [4,25]: k is set to 10;
- **Consensus Method:** the state-of-the-art technique for selecting neighborhoods to construct affinity graphs. This method strengthens the graph's robustness by incorporating consensus information from various neighborhoods in a specified kNN graph [2];
- **Spectral Edge Sparsification Method:** The state-of-the-art method for detecting non-critical, misleading, and superfluous edges in the graph [4].

4.4 Results Of Solution Quality

Table 1 shows the solution quality of graph-based spectral clustering algorithm on the USPS and the MNIST data sets.

Table 1. Clustering Accuracy (%)

Data Set	k-NN	Consensus	Spectral Spar	Ours
USPS	64.31	68.54	70.74	78.87
MNIST	59.68	61.09	60.09	70.40

The clustering outcomes of our approach are derived by selecting the top 2,000 and 1,500 nodes with the highest robustness from the USPS and MNIST data sets, respectively. It is evident that our method surpasses the baseline kNN graph by over 14% and 10% in accuracy on the USPS and MNIST data sets, respectively, demonstrating the effectiveness of our approach in improving the solution quality. Furthermore, the results achieved by our approach, which surpasses the second-best denoising method by margins of 8% for the USPS data set and 9% for the MNIST data set, validate the advantage of employing the algorithm on nodes with reduced sensitivity to noisy edges over existing methods that concentrate on the elimination of such edges.

It can be seen that methods aimed at resolving noisy issues by removing noisy edges do manifest a clear improvement on the USPS data set, although they fall short of the enhancements our method provides when compared to the baseline. On the MNIST data set, however, their effects are marginal, with the consensus method and spectral sparsification method achieving only 1.5% and 0.5% increases in accuracy, respectively. We conjecture that this is attributable to the MNIST data set containing a greater number of features than the USPS data set, which may include more non-robust features that can induce noisy edges. Consequently, the task of excising noisy edges from MNIST is significantly more formidable. Existing denoising methods face difficulties in purging noisy edges without also eliminating beneficial edges. In contrast, our approach, which involves selecting nodes that are robust to noise and executing graph-based algorithms exclusively among these robust nodes, serves as an attack-agnostic method. This renders it highly effective for both the USPS and MNIST data sets.

4.5 Efficacy of Resolving Computational Bottleneck

In spectral clustering algorithm, the first step of constructing the kNN graph can be completed within a time complexity of $O(|n|log|n|)$ [26], where n is number of nodes in the graph. The third step, k-means, can also be performed in linear time [27]. However, the second step, eigen-decomposition, has a time complexity of $O(n^3)$, making it the computational bottleneck of the entire algorithm, as well as the dominant term in the time complexity analysis of the algorithm.

In our method, eigen-decomposition is only performed for a subset composed of a small number of robust nodes, thereby significantly reducing the computational cost.

Figure 1 and Table 2 show the eigen-decomposition time of the original full data set and our selected robust node set. It can be observed that by utilizing the selected robust node set, eigen-decomposition has been accelerated by a factor of 9 for the USPS data set and 90 for the MNIST data set, respectively. It is expected that the proposed method will be a key enabler for running computationally expensive graph-based algorithms in scenarios that require extremely fast response times and on devices with limited computational capabilities.

4.6 Parameter Discussion

In our method, the construction of G_{input} and G_{output} involves the selection of k in the k-NN algorithm. Our preliminary experiments indicate that when k is large enough to adequately capture the underlying structure of the data, the algorithm is not highly sensitive to the parameter setting. This means that the parameter choice is not overly strict. For example, when constructing G_{input}, we can choose k=50, and when constructing G_{output}, we can select k=10, achieving very goodresults. It is important to note that the k-NN graph is just one common

Table 2. Eigen-decomposition Time (s)

Data Set	Original Node Set	Robust Node Set
USPS	0.72	0.08
MNIST(test)	6.35	0.07

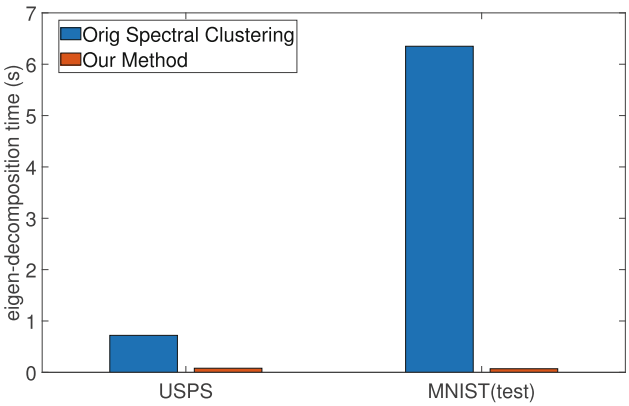


Fig. 1. Eigen-decomposition time of original node set and robust node set.

method to capture the underlying structure of the data; it is neither the only nor the necessary approach. Other more advanced graph learning methods can also be used.

5 Conclusion

In this paper, we view the noisy edges in graphs used from the perspective of adversarial attack. Building upon this viewpoint, we proposed a method based on robust nodes to mitigate the impact of noisy edges, grounded in adversarial robustness evaluation. Experimental results from real-world datasets show that our approach significantly boosts the performance of graph-based algorithms and outperforms the state-of-the-art methods in addressing noisy edges by a large margin.

References

1. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
2. Premachandran, V., Kakarala, R.: Consensus of k-NNS for robust neighborhood selection on graph-based manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1594–1601 (2013)
3. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(500), 2323–2326 (2000)

4. Wang, Y., Feng, Z.: Towards scalable spectral clustering via spectrum-preserving sparsification. In: 33rd British Machine Vision Conference 2022, BMVC 2022, London, 21–24 November 2022. BMVA Press (2022)
5. Yang, X., Prasad, L., Latecki, L.J.: Affinity learning with diffusion on tensor product graph. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 28–38 (2012)
6. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp. 285–295 (2001)
7. Goodfellow, I., McDaniel, P., Papernot, N.: Making machine learning robust against adversarial inputs. Commun. ACM **61**(7), 56–66 (2018)
8. Fawzi, A., Fawzi, O., Frossard, P.: Analysis of classifiers' robustness to adversarial perturbations. Mach. Learn. **107**(3), 481–508 (2018)
9. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
10. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 427–436 (2015)
11. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
12. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
13. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 (2017)
14. Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., Jordan, M.: Ml-loo: detecting adversarial examples with feature attribution. Proc. AAAI Conf. Artif. Intell. **34**(04), 6639–6647 (2020)
15. Agarwal, C., Nguyen, A., Schonfeld, D.: Improving robustness to adversarial examples by encouraging discriminative features. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 3801–3505. IEEE (2019)
16. Liu, X., Li, Y., Wu, C., Hsieh, C.-J.: Adv-bnn: improved adversarial defense through robust bayesian neural network. arXiv preprint arXiv:1810.01279 (2018)
17. Weng, T.-W., et al.: Evaluating the robustness of neural networks: an extreme value theory approach. arXiv preprint arXiv:1801.10578 (2018)
18. Cheng, W., Deng, C., Zhao, Z., Cai, Y., Zhang, Z., Feng, Z.: Spade: a spectral method for black-box adversarial robustness evaluation. In: International Conference on Machine Learning, pp. 1814–1824. PMLR (2021)
19. Spielman, D.: Spectral graph theory. Combinat. Sci. Comput. **18**, 18 (2012)
20. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
21. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14 (2001)
22. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support vector clustering. J. Mach. Learn. Res. **2**, 125–137 (2001)
23. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Courier Corporation (1998)
24. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
25. Szlam, A., Bresson, X.: A total variation-based graph clustering algorithm for cheeger ratio cuts. UCLA Cam Report, pp. 09–68 (2009)

26. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 824–836 (2018)
27. Pakhira, M.K.: A linear time-complexity k-means algorithm using cluster shifting. In: International Conference on Computational Intelligence and Communication Networks, pp. 1047–1051. IEEE (2014)
28. Kyng, R., Sachdeva, S.: Approximate gaussian elimination for Laplacians-fast, sparse, and simple. In: IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pp. 573–582. IEEE (2016)

Adaptive Graph-Based Manifold Learning for Gene Selection

Reshma Rastogi(✉) and Mamta Bhattarai Lamsal

Machine Learning and Statistical Inference (MLSI) Lab, Department of Computer Science, South Asian University, Maidan Garhi, India
reshma.khemchandani@sau.ac.in

Abstract. Microarray data, when coupled with advanced computational and statistical techniques, offers profound insights into cause of diseases and personalized therapy. However, the enormous genes present in microarray data poses challenges for identifying relevant gene selection, especially in limited labeled dataset. Conventional methods for graph construction suffer from empirical parameter selection, potentially failing to capture intrinsic data properties. To address these issues, we introduce the semi-supervised Adaptive Graph-based Manifold Learning Gene Selection (AGMLGS) approach. This unified framework integrates graph construction and projection matrix learning, preserving high-dimensional data structure (in particular for gene data) in a lower-dimensional space without losing their physical meaning. Our method surpasses seven state-of-the-art algorithms across nine datasets, proving its effectiveness in terms of average precision and exhibiting reasonable computational efficiency in the majority of cases. The MATLAB code employed in the proposed AGMLGS model are accessible in the following URL https://github.com/ml-lab-sau/AGMLGS.

Keywords: Adaptive graph · manifold learning · semi-supervised · gene selection · feature selection · microarray data

1 Introduction

Is well known fact that microarray data is correlated to diseases like cancer, therefore it becomes essential for biomedical applications including disease subtyping, biomarker discovery, and precision medicine. Microarray data are high dimensional, they contain thousands of genes but the samples available are limited resulting in the curse of dimensionality [1,2]. This results in significant challenges while dealing with these data and makes it computationally expensive and difficult to process for classification or clustering tasks [3,4]. The paper uses the terms "gene selection" and "feature selection" are used interchangeably because both refer to the process of selecting features from the data.

Machine Learning and Statistical Inference Lab, Department of Computer Science, South Asian University.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 453–468, 2025.
https://doi.org/10.1007/978-3-031-78107-0_29

To deal with the "curse of dimensionality", dimension reduction such as feature extraction or feature selection is performed on these data. Generally, all features present in the microarray data do not hold an equal contribution as some of them are redundant or irrelevant concerning the analysis. To choose a subset of the most pertinent gene while reducing data from a high dimension to its lower dimension space, retaining the physical meaning of the data, feature selection is therefore essential [5]. Multiple dimension reduction algorithms are available for diverse objectives, and the conceptual principles used to create the proposed algorithm are underlined.

Dimension reduction falls into two categories mainly linear and non-linear. Linear methods, such as PCA (Principle Component Analysis) and LDA (Linear Discriminant Analysis), are simple to implement but fail to capture the innate nonlinear structure of data. It has been demonstrated that high-dimensional data often possess non-linear structures, with microarray data being a prime example due to its high dimensionality and diverse, complex non-linear structure. Non-linear methods are often advantageous for handling such data, however, approaches like LLE, ISOMAP, and Laplacian Eigen Map, while performing well on linear or small size benchmark datasets, often struggle when applied to real-world non-linear datasets. Manifold-based methods are well-suited when embedded with appropriate graph based method for addressing these challenges inherent in nonlinear datasets [6,7].

Graph-based methods are gaining popularity due to their remarkable performance in handling high-dimensional data. Traditionally, graph construction methods rely on prior knowledge, and low-dimensional features are derived from this graph. Typically, graphs are constructed using either the KNN or ϵ-ball graph approaches, where the values of K and ϵ are chosen empirically. However, if these parameters are not selected correctly, it can affect the effectiveness of the algorithm [8]. An alternative approach to address this issue is through self-supervised learning, particularly beneficial for image data, as it eliminates the need for data augmentation [9]. This research paper proposes an adaptive method for constructing a graph, where the number of features to be selected does not need to be predetermined. Instead, features are chosen using a learned sparse projection matrix, effectively addressing this problem.

Gene data can be classified into three types: unlabeled, labeled, and partially labeled. Consequently, the problem of gene selection can be framed as an unsupervised, supervised, or semi-supervised classification problem [5]. Unlabeled data is typically used to understand the spatial arrangement or distribution of the data. In contrast, labeled data is used to improve the separation between different categories of data points. Semi-supervised learning combines the strengths of both labeled and unlabeled data [6]. By utilizing both types of data, semi-supervised learning aims to improve the performance and accuracy of learning algorithms [10].

Moreover, in semi-supervised feature selection, graph-based sparse feature selection is critical. However, the graph construction approaches make these semi-supervised graph-based methods sensitive to outliers and noise [11].

Our algorithm overcomes this shortcoming by introducing the adaptive graph approach in the manifold semi-supervised sparse learning. The main contributions of our work are given below:

1. The unified model that combines gene selection and classification tasks into a single optimization module.
2. Construction of adaptive graph which helps in identifying the global structure of genes data relationship.
3. Captures both the local and global intrinsic structure of genes in microarray data while performing dimension reduction.
4. Closed form solution which is easy to interpret and is efficient.

The rest of the paper progresses as outlined below: Sect. 2 delves into the relevant literature, Sect. 3 introduces the proposed model and discusses on optimization and solutions. Section 4 outlines the experimental results, and the paper concludes with a summary in the final section.

2 Related Works

Microarray data is often acknowledged for its complexity, presenting two significant challenges: accurately representing the data and maintaining its intrinsic structure while reducing it from high dimensions to low-dimensional space. Graph-based methods are increasingly favored for data representation due to their inherent nature. However, traditional graph learning methods construct the graph primarily based on prior knowledge, typically through K nearest neighbours (KNN) or the ϵ ball approach. Yet, empirically determining parameters such as K or ϵ may occasionally fail to capture the intrinsic properties of the original data, as the Laplacian information associated with the data remains fixed during the optimization process. Typically, in such approaches, graph construction and the learned projection matrix are treated as independent processes. Initially, the graph is constructed, and during the optimization process, the projection matrix is learned. If the graph fails to provide an accurate representation of the data, the issue persists throughout the learning process. To address this issue, adaptive graph learning has been introduced, where graph construction and projection matrix learning are unified into a single framework [8].

Addressing the challenge of reducing dimensionality while preserving the data's inherent structure is crucial. Manifold regularization tackles this by ensuring that similar samples in the original space remain close in the reduced-dimensional space as well [12]. Manifold regularization technique enhances smoothness on the constructed manifold graph, it remains separate from the classification aspect and may not be suitable with the supervised framework [11]. Moreover, manifold regularization provides a robust framework that is highly beneficial for semi-supervised learning. Importantly, most existing model uses a sparse model for preserving the geometric structure and a laplacian for graph construction. The sparse model especially lacks a theoretical explanation for

preserving the global structure. To overcome this shortcoming, paper [13] propose extending the sparse model by a series of scale factors (re-scale regression coefficient) to address limitations of both sparse and graph learning. Another approach, as demonstrated in [14], is in the semi-supervised framework of adaptive local embedding learning, where distinct graphs are generated for local and global structures. The local graph is crafted based on available label information, while the global graph is constructed independently of label information. However, this methodology proves effective only when ample sample size is provided.

Our approach integrates an adaptive graph generated using the radial basis function (RBF) into a semi-supervised manifold learning framework. While previous studies such as [8] highlight adaptive graph learning for dimension reduction in unsupervised scenarios, and [13] underscore the significance of manifold-based learning in semi-supervised contexts, our model uniquely utilizes adaptive graph learning within a manifold semi-supervised approach. It utilizes the sparsity of the learned projection matrix for feature selection.

3 Proposed Model

3.1 Mathematical Formulation

Matrices are defined using uppercase letters. The symbols employed in the paper are enumerated as: A represents an arbitrary matrix, A_i denotes the i^{th} row vector of matrix A, $A_{.j}$ refers to the j^{th} column vector of matrix A, A_{ij} represents the ij^{th} element of matrix A, X signifies the data matrix $[x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$, $X = [X_l, X_u]$ symbolizes that X comprises both labeled (l) and unlabeled (u) data, Y stands for the label information matrix with c number of classes, Y_l indicates the label information corresponding to labeled data, Y_u represents the label information corresponding to missing labels (unlabeled data), and $Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}$ denotes the combined label information matrix Y. Additionally, F signifies the global indicator matrix. The objective function of sparse semi-supervised feature selection consists of the loss and the regularization functions. The least square regression is widely used as a loss function and $l_{p(0<p\leq 1)}$-norm, or l_2-norm for regularization. The $l_{p(0<p\leq 1)}$-norm consistently provides a solution with sparsity but it tends to overfit. On the other hand, the l_2-norm mitigates the overfitting issue but sacrifices sparsity. The $l_{2,1}$-norm combines the benefits of both norms by preventing overfitting, and maintaining sparsity in the projection matrix, and is easily optimized. The objective function for sparse feature selection is given below:

$$\min_{W} \quad ||X^T W - Y||_F^2 + \alpha ||W||_{2,1}. \tag{1}$$

As W is a global projection matrix, α is a regularized parameter. As the Y matrix is not fully available in the semi-supervised approach [11], our model can be modified by replacing it with the predicted label matrix F [15,16] as follows:

$$\begin{aligned} &\min_{W} \quad ||X^T W - F||^2 + \alpha ||W||_{2,1} \\ &\text{s.t} \quad F^T F = I, F \geq 0. \end{aligned} \tag{2}$$

In order to use all the information present in data, i.e. to fully exploit unlabeled and labeled samples, a predicted label matrix F is define for all training data. As a result, the predicted label matrix F can be identified as:

$$\min_{F} \quad Tr(F^T LF) + Tr((F-Y)^T U(F-Y)). \tag{3}$$

In order to satisfy the smoothness on both manifold structure and the ground truth labels of samples, we define U as $U_{ij} = \infty$ if x_i is labeled, $U_{ij} = 0$ otherwise, let A_{ij} be the similarity matrix defined as $A_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$, D is diagonal matrix, $D_{ii} = \sum_{j=1}^{n} A_{ij}$, and Laplacian matrix is given as $L = D - A$ [6,7] .

In order to identify the relevant features through an adaptive learning approach the following terms are used:

$$\min_{W,S} \quad \sum_{i=1}^{n} ||W^T(X_i - XS_i)||^2 + \beta \sum_{i=1}^{n} ||R_i \otimes S_i||^2, \tag{4}$$

where $R_i = (r_{i1}, ...r_{ii-1}, \infty, ...r_{in})^T \in R^{n\times 1}$ is a locality adaptor vector, $r_{ij} = (\exp(||x_i - x_j||)^2/\sigma)$ and $\sigma = 1/n^2(\sum_{i,j}^{n} ||x_i - x_j||)^2$. A lower r_{ij} indicates that x_i is more similar to x_j, and vice versa. Minimizing $||R_i \otimes S_i||^2$ about S causes the algorithm to assign small or nearly zero reconstruction coefficients to samples that are far from x_i [8].

Unlike the static graph approach, the final model combines the sparse feature selection where the genes in the microarray data are selected and the adaptive graph enriches the model by giving the real representation data. Similarly, semi-supervised manifold learning retains the original intrinsic structure in the reduced space. The final model for semi-supervised adaptive graph-based manifold learning for gene selection can be explained through the following convex optimization problem.

$$\min_{F,W,S} \quad ||X^T W - F||^2 + \alpha||W||_{2,1} + Tr(F^T LF) + Tr((F-Y)^T U(F-Y)) +$$
$$\sum_{i=1}^{n} ||W^T(X_i - XS_i)||^2 + \beta \sum_{i=1}^{n} ||R_i \otimes S_i||^2 + Tr(W^T XX^T W - I). \tag{5}$$

$$\min_{F,W,S} \quad ||X^T W - F||^2 + \alpha Tr(W^T D_w W) + Tr(F^T LF) + Tr((F-Y)^T U(F-Y)) +$$
$$\sum_{i=1}^{n} ||W^T(X_i - XS_i)||^2 + \beta \sum_{i=1}^{n} ||R_i \otimes S_i||^2 + Tr(W^T XX^T W - I). \tag{6}$$

where,

$$D_W = \begin{bmatrix} 1/2||W_1||_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/2||W_d||_2 \end{bmatrix}$$

3.2 Optimization and Solution

This section provides an overview of how the proposed optimization problem is solved via alternating optimization where each decision variable is optimized keeping the others constant.

To update F, we fixed values for S and W, the process begins by initializing a matrix with random values. The expression for F is then derived from Eq. (6), considering Lagrangian function of (6) and differentiating it with respect to F, yielding the following equation:

$$F = (2I + L + U)^{-1}(X^T W + UY) \tag{7}$$

In a similar way, to update S while keeping F and W fixed, we substitute $y_i = W^T x_i$ and $Y = W^T X$. Deriving the column vector expression for S, differentiating Lagrangian function with respect to S and equating it to zero yields:

$$s_{.i} = (2YY^T + \beta r_{.i})^{-1}(Y y_i^T) \tag{8}$$

Finally, to update W while keeping F and S fixed, we introduce the matrix 'M' as $M = I - S$ and taking the derivative for expression W, differentiating Lagrangian function with respect to W by equating it to zero, we derive the expression for the projection matrix W [13,17] as:

$$W = (XX^T + \alpha D_W + XMM^T X^T)^{-1} XF, \tag{9}$$

3.3 Algorithm

The algorithm gives the step-wise solution for solving the projection matrix. The algorithm is simple and easy to implement and provides a closed-form solution.

Algorithm 1. Algorithm to find W

Initialize: W.
Input: $\{X, Y\}$, $X = [x_l, x_u]$, $Y = [y_l, y_u]$, α, β
Calculate D_W, L, R, σ.
Repeat:
- Compute F using updating equation (7).
- Compute S using updating equation (8).
- Compute W using updating equation (9).

Until convergence
Output: $F = XW$ Predicted Label, leading feature indices.

Once the projection matrix W is obtained, the predictor matrix for both the training and testing datasets is created through multiplication with W.

3.4 Convergence Proof

To prove the convergence of the sequence $\{J(F_t, W_t, S_t)\}$, we first express the objective function explicitly:

$$J(F,W,S) = \|X^TW - F\|^2 + \alpha \operatorname{Tr}(W^T D_w W) + \operatorname{Tr}(F^T L F) + \operatorname{Tr}((F-Y)^T U (F-Y)) + \sum_{i=1}^{n} \|W^T(X_i - XS_i)\|^2 + \beta \sum_{i=1}^{n} \|R_i \otimes S_i\|^2 + \operatorname{Tr}(W^T X X^T W - I).$$

The proof of convergence relies on showing that the sequence of objective function values $\{J(F_t, W_t, S_t)\}$ generated by the algorithm decreases monotonically and is bounded below.

1. **Convexity with respect to** F: For fixed W and S, consider the subproblem with respect to F:

$$\min_F \quad \|X^TW - F\|^2 + \operatorname{Tr}(F^T L F) + \operatorname{Tr}((F-Y)^T U (F-Y)).$$

   Each term in the subproblem is convex in F:
   - $\|X^TW - F\|^2$ is quadratic in F.
   - $\operatorname{Tr}(F^T L F)$ is a quadratic form in F with L positive semi-definite.
   - $\operatorname{Tr}((F-Y)^T U (F-Y))$ is a quadratic form in F with U positive semi-definite.

   Thus, the subproblem is convex in F. Minimizing this subproblem with respect to F yields F_{t+1} such that:

$$J(F_{t+1}, W_t, S_t) \leq J(F_t, W_t, S_t).$$

2. **Convexity with respect to** S: For fixed F and W, consider the subproblem with respect to S:

$$\min_S \quad \sum_{i=1}^{n} \|W^T(X_i - XS_i)\|^2 + \beta \sum_{i=1}^{n} \|R_i \otimes S_i\|^2.$$

   Each term in the subproblem is convex in S:
   - $\|W^T(X_i - XS_i)\|^2$ is quadratic in S.
   - $\|R_i \otimes S_i\|^2$ is quadratic in S.

   Thus, the subproblem is convex in S. Minimizing this subproblem with respect to S yields S_{t+1} such that:

$$J(F_{t+1}, W_t, S_{t+1}) \leq J(F_{t+1}, W_t, S_t).$$

3. **Convexity with respect to** W: For fixed F and S, consider the subproblem with respect to W:

$$\min_W \quad \|X^TW - F\|^2 + \alpha \operatorname{Tr}(W^T D_w W) + \sum_{i=1}^{n} \|W^T(X_i - XS_i)\|^2 + \operatorname{Tr}(W^T X X^T W - I).$$

   Each term in the subproblem is convex in W:

- $\|X^T W - F\|^2$ is quadratic in W.
- $\alpha \operatorname{Tr}(W^T D_w W)$ is a quadratic form in W with D_w positive semi-definite.
- $\|W^T(X_i - XS_i)\|^2$ is quadratic in W.
- $\operatorname{Tr}(W^T X X^T W - I)$ is quadratic in W.

Thus, the subproblem is convex in W. Minimizing this subproblem with respect to W yields W_{t+1} such that:

$$J(F_{t+1}, W_{t+1}, S_{t+1}) \leq J(F_{t+1}, W_t, S_{t+1}).$$

Combining the results from the above steps, we have:

$$J(F_{t+1}, W_{t+1}, S_{t+1}) \leq J(F_{t+1}, W_t, S_{t+1}) \leq J(F_{t+1}, W_t, S_t) \leq J(F_t, W_t, S_t).$$

This shows that the sequence $\{J(F_t, W_t, S_t)\}$ is monotonically decreasing. Additionally, since the objective function $J(F, W, S)$ is non-negative and bounded below, the sequence $\{J(F_t, W_t, S_t)\}$ converges.

3.5 Time Complexity Analysis

The algorithm time complexity analysis involves examining operations for updating each optimization variable. Updating F, which involves matrix multiplications and inversions with dimensions $n \times c$, $m \times n$, and $m \times c$ for F, X, and W, respectively, has a time complexity of $O(n^3)$. Updating S similarly, with dimensions $n \times n$ and $n \times c$ for S and Y, yields $O(n^3)$. Likewise, updating W requires operations with dimensions $m \times m$, resulting in $O(m^3)$. Considering iterative operations until convergence, denoted by k iterations, the total time complexity is $O(k \cdot (n^3 + m^3))$, with n, m, and c as the number of samples, feature, and class respectively.

4 Experiments

The effectiveness of the proposed AGMLGS algorithm is validated through comparisons with seven state-of-the-art algorithms. AGMLGS operates as a unified model, similar to FME, while the other comparison algorithms follow a two-phase process. Initially, these algorithms perform feature selection, choosing ten percent of the overall features. Classification is then conducted using Binary Relevance Support Vector Machine (BRSVM) [12] in a semi-supervised setting with 10%, 30%, and 50% labeled data. Unlike these methods, AGMLGS and FME do not predefine the percentage of top-ranked features. Instead, both unified algorithms employ sparse learning to inherently identify the top features, which can then be adjusted to the desired number. Additionally, since AGMLGS functions as a unified model, feature selection and classification are integrated within the same algorithmic framework.

4.1 Comparing Algorithms

Below is a detailed description of the state-of-the-art algorithms used for comparison.

1. **MDFS**[1] (Manifold regularized discriminative feature selection for multi-label learning) is a supervised embedded multi-label feature selection method with manifold regularization that incorporates $l_{2,1}$-norm regularization [17].
2. **JMI**[2] (Simple strategies for semi-supervised feature selection) is a semi-supervised approach that uses specific prior knowledge of the domain in situations where labels are not randomly missing. This simple approach accurately captures the dynamics of feature selection [18].
3. **SRCFS**[3] (Unsupervised feature selection with multi-subspace randomization and collaboration) is an unsupervised feature selection method that uses multi-subspace randomization and collaboration which demonstrates its efficiency and robustness on high-dimensional datasets. This approach integrates Laplacian scores from balanced random subspaces to enhance feature selection performance [19].
4. **EUFS**[4] (Embedded unsupervised feature selection) is an innovative unsupervised feature selection algorithm that seamlessly integrates feature selection into a clustering algorithm through sparse learning, avoiding the need for label information transformation [20].
5. **DUFS** (See Footnote 4)(Pairwise dependence-based unsupervised feature selection) uses mutual information to capture feature dependence, enhances regression-based feature selection and results in a compact feature set that effectively eliminates redundancy [21].
6. **FME**[5] pagination(Flexible Manifold Embedding: A Framework for Semi-Supervised and Unsupervised Dimension Reduction) a unified manifold learning framework that employs a linear regression function to map new data points while integrating label fitness, manifold smoothness, and a flexible penalty term on the regression residue. This framework effectively utilizes label information from labeled data and the manifold structure from both labeled and unlabeled data [22].
7. **LC-GODR**(see footnote 5) (The Locality Constrained Graph Optimization Dimensionality Reduction) algorithm addresses the limitations of traditional graph-based dimensionality reduction methods by combining graph optimization and projection matrix learning into a unified framework. Unlike conventional methods, LC-GODR adaptively updates the graph during the dimensionality reduction process, ensuring it accurately reflects the underlying data structure. Additionally, the inclusion of locality constraints helps preserve local information, distinguishing LC-GODR from other graph optimization-based dimensionality reduction techniques [8].

[1] https://github.com/jiazhang-ml/MDFS.
[2] https://github.com/sechidis/2018-MLJ-Semi-supervised-feature-selection?tab=readme-ov-file.
[3] https://github.com/huangdonghere/SRCFS.
[4] https://github.com/CAU-AIR-Lab/DUFS.
[5] https://github.com/ml-lab-sau/AGMLGS/tree/main/function.

4.2 Database Description

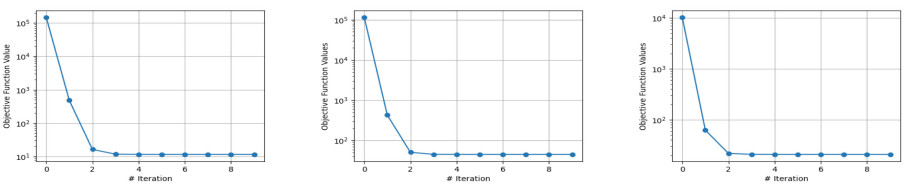
Table 1 discusses the microarray datasets where features are more than 1500 except Pro_can and DNA. Specifically, the datasets Alon, Gravier, Breast_can, Globun, Sing_pro, and Lung_can are sourced from [23–28], while the remaining datasets are obtained from the Kaggle database. Majority of the data are cancer related, except for one dataset, which is DNA data.

Table 1. Dataset Description.

Dataset	No of Instances	No of Features	No of Classes	Type
Endo_can	42	1771	7	Endocrine Cancer
Breast_can	49	7129	2	Breast Cancer
Alon	62	2000	2	Colon Cancer
Globun	72	7129	2	Lukemia
Pro_can	102	338	2	Prostrate Cancer
Sing_pro	102	12600	2	Prostrate Cancer
Gravier	168	2905	2	Breast Cancer
Lung_can	181	1626	2	Lungs Cancer
DNA	3186	180	3	DNA Data

4.3 Convergence Analysis

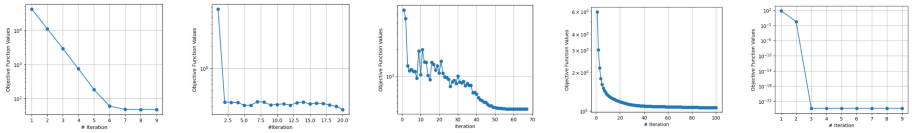
The convergence behavior of the proposed algorithm is evaluated through graphical representation, depicted in figures mentioned in Fig. 1. This evaluation focuses on three distinct datasets: Alon with 10% labeled data, Pro_can with 30% labeled data, and Lung_can with 50% labeled instances. The convergence graph is constructed with parameters are fixed at $\alpha = 0.9$ and $\beta = 0.3$. The graphs indicate that the proposed algorithm achieves convergence within a few iterations, typically fewer than five. This suggests the convex nature of the optimization framework, facilitating rapid convergence and hence efficient for real world datasets.



(a) Alon . (b) Lung_can . (c) Pro_can.

Fig. 1. Convergence plots

Similarly, convergence plots as shown in Fig. 2 are generated for comparison algorithms MDFS, SRCFS, EUFS, DUFS, and LC-GODR on the Pro_can dataset to showcase the effectiveness of the proposed AGMLGS algorithm. Convergence plots for FME and Semi_JMI are excluded as they follow single-step process. Comparing these plots, proposed AGMLGS and LC-GODR demonstrate rapid convergence, while EUFS and DUFS require more iterations to converge, and SRCFS converges after more than 20 iterations. MDFS achieves convergence in seven iterations. We can conclude that the proposed AGMLGS algorithm as well as LC-GODR converges faster compare to other algorithms.

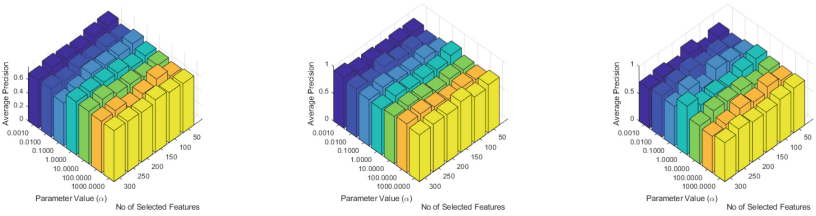


(a)MDFS (b)SRCFS (c)EUFS (d)DUFS (e)LC-GODR

Fig. 2. Convergence Plot of Comparing Algorithms on the Pro_can Dataset

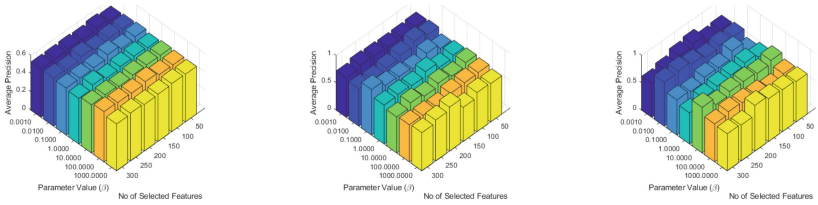
4.4 Parameter Sensitivity Analysis

Parameter sensitivity analysis was conducted on parameters, α and β, with respect to the average precision and the number of selected features. Figures in Fig. 3 and Fig. 4 illustrate the sensitivity analysis for α and β, respectively, across three datasets representing 10%, 30%, and 50% labeled samples. In the analysis, β was held constant at 1 while α was varies from $[10^{-3}, 10^{3}]$, and similar for the second analysis. The optimal choice of the parameters affects the average precision which is also dependent on the number of selected features.



(a) Pro_can (b) Lung_can (c) Alon

Fig. 3. Parameter Sensitivity Test for α with (a) 10% labeled Pro_can dataset, (b) 30% labeled Lung_can dataset, and (c) 50% labeled Alon dataset



(a) Breast_can (b) Alon (c) Endo_can

Fig. 4. Parameter Sensitivity Test for β with (a) 10% labeled Breast_can dataset, (b) 30% labeled Alon dataset, and (c) 50% labeled Endo_can dataset

4.5 Experimental Results

Table 2 presents the average precision achieved by various algorithms using different percentages (10%, 30%, and 50%) of labeled data. The experiment employs five-fold cross-validation and was conducted on a standalone system running Windows 11, equipped with an Intel Core i7-8700 CPU at 3.20GHz, and 16GB of RAM. The best results are indicated in bold. Table 3 displays the average execution time for each algorithm. Cells marked with "OM" indicate that the algorithm encountered "Out of Memory" issues on the specified computer configuration.

The Fig. 5 illustrates the change in Average Precision with respect to selected features for all algorithms, including the proposed AGMLGS algorithm. AGMLGS and FME perform feature selection based on the sparsity of the learned projection matrix W, while other algorithms, involve feature selection followed by the semi-supervised classifier BRSVM for average precision calculation.

4.6 Result Analysis and Discussions

Table 2 reports the average precision scores obtained from seven different algorithms and proposed algorithm (AGMLGS) across nine diverse datasets. With 10%, 30% and 50% labeled instances, AGMLGS exhibits superior performance in most of the datasets.

Table 3 presents the average execution times for each algorithm across nine datasets. AGMLGS demonstrates varying performance in terms of processing time compared to other methods. AGMLGS consistently proves to be a competitive or superior choice in terms of processing efficiency across a range of datasets.

In the plots depicted in Fig. 5 (5a and 5b), it becomes apparent that the number of relevant features contributing varies depending on the chosen dataset. Traditional methods of identifying the appropriate number of features can present challenges, often requiring extensive testing with different feature selections, which can be quite time-consuming. However, employing the proposed algorithm and maintaining a proper threshold for sparsity in the learned projection matrix

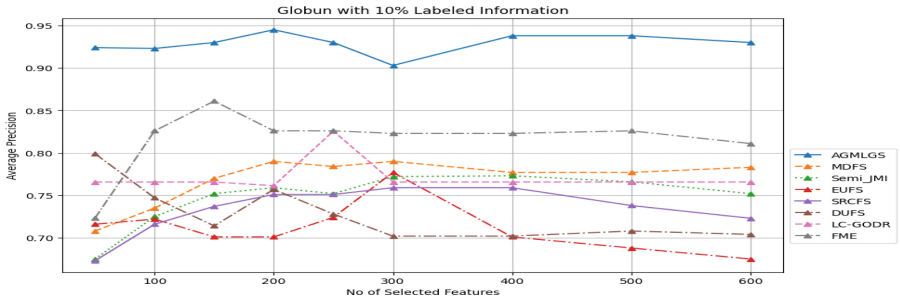
Table 2. Average Precision with Different Percentages of Labeled Data

Dataset		Proposed	Comparing Methods						
		AGMLGS	MDFS	SRCFS	Semi_JMI	EUFS	DUFS	FME	LC-GODR
	10%	**0.69 ± 0.01**	0.55 ± 0.05	0.62 ± 0.04	0.42 ± 0.04	0.64 ± 0.07	0.56± 0.04	0.56±0.05	0.62±0.05
	30%	**0.74 ± 0.03**	0.65 ± 0.04	0.57 ± 0.06	0.50 ± 0.02	0.61 ± 0.03	0.57 ± 0.06	0.64±0.05	0.54±0.07
Endo_can	50%	**0.75 ± 0.05**	0.65 ± 0.04	0.63 ± 0.08	0.51 ± 0.02	0.62 ± 0.02	0.67 ± 0.03	0.64±0.05	0.62±0.07
	10%	**0.81 ± 0.01**	0.73 ± 0.03	0.78 ± 0.03	0.74 ± 0.03	0.75 ± 0.04	0.78 ± 0.04	0.71±0.02	0.73±0.025
	30%	**0.82 ± 0.01**	0.79 ± 0.05	0.81 ± 0.04	0.74± 0.03	0.74 ± 0.03	0.75 ± 0.05	0.77±0.03	0.78±0.03
Breast_can	50%	**0.85 ± 0.01**	0.79 ± 0.04	0.82 ± 0.04	0.74 ± 0.03	0.73 ± 0.04	0.75 ± 0.04	0.76±0.03	0.76±0.03
	10%	0.83 ± 0.01	**0.86 ± 0.02**	0.84 ± 0.06	0.82 ± 0.03	0.85 ± 0.04	0.85 ± 0.01	0.82±0.03	0.69±0.04
	30%	0.85 ± 0.03	**0.86 ± 0.02**	0.81 ± 0.02	0.82 ± 0.04	0.84 ± 0.02	0.82 ± 0.01	0.82±0.03	0.82±0.03
Alon	50%	0.87 ± 0.01	**0.90± 0.02**	0.84 ± 0.01	0.82 ± 0.04	0.84 ± 0.03	0.85 ± 0.01	0.82±0.03	0.82±0.03
	10%	**0.93 ± 0.02**	0.77 ± 0.05	0.74 ± 0.04	0.69 ± 0.03	0.84 ± 0.03	0.87 ± 0.02	0.85±0.05	0.79±0.05
	30%	**0.91 ± 0.02**	0.81 ± 0.04	0.81 ± 0.03	0.70 ± 0.04	0.83 ± 0.04	0.90 ± 0.02	0.77±0.04	0.77±0.04
Globun	50%	0.92 ± 0.02	0.87 ± 0.04	0.83 ± 0.04	0.75 ± 0.04	0.87 ± 0.03	**0.93 ± 0.01**	0.77±0.04	0.77±0.04
	10%	**0.86 ± 0.01**	0.76 ± 0.02	0.77 ± 0.02	0.76 ± 0.03	0.80 ± 0.03	0.85 ± 0.03	0.73±0.01	0.76±0.01
	30%	**0.91 ± 0.01**	0.76 ± 0.02	0.77 ± 0.02	0.76 ± 0.03	0.79 ± 0.03	0.86 ± 0.03	0.73±0.03	0.73±0.03
Pro_can	50%	**0.90 ± 0.01**	0.84 ± 0.03	0.77 ± 0.01	0.78 ± 0.03	0.78 ± 0.02	0.85 ± 0.03	0.77±0.02	0.77±0.02
	10%	**0.90 ± 0.01**	0.83 ± 0.04	0.83 ± 0.02	0.76 ± 0.02	0.84 ± 0.02	0.86 ± 0.03	0.80±0.043	OM
	30%	**0.93 ± 0.02**	0.86 ± 0.01	0.81 ± 0.02	0.76 ± 0.02	0.89 ± 0.02	0.85 ± 0.03	0.73Â±0.03	OM
Sing_pro	50%	**0.93 ± 0.01**	0.86 ± 0.01	0.84 ± 0.01	0.77 ± 0.03	0.91 ± 0.03	0.89 ± 0.02	0.70±0.01	OM
	10%	**0.83 ± 0.01**	0.75 ± 0.03	0.71 ± 0.01	0.67 ± 0.02	0.67 ± 0.01	0.67 ± 0.01	0.83±0.018	0.77±0.04
	30%	**0.86 ± 0.01**	0.80 ± 0.02	0.75 ± 0.01	0.67 ± 0.02	0.69 ± 0.02	0.69 ± 0.02	0.83Â±0.03	0.81±0.03
Gravier	50%	**0.85 ± 0.02**	0.80 ± 0.03	0.81 ± 0.02	0.68 ± 0.02	0.78 ± 0.03	0.76 ± 0.02	0.83±0.02	0.81±0.03
	10%	0.92 ± 0.02	**0.96 ± 0.01**	0.94 ± 0.02	0.91 ± 0.01	.94 ± 0.01	0.95 ± 0.01	0.91±0.03	0.75±0.09
	30%	**0.97 ± 0.01**	0.96 ± 0.01	0.97 ± 0.01	0.93 ± 0.03	0.96 ± 0.01	0.96 ± 0.01	0.91±0.01	0.84±.07
Lung_can	50%	**0.98 ± 0.01**	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.01	0.91 ± 0.02	0.97 ± 0.01	0.91±0.01	0.91±0.01
	10%	**0.93 ± 0.01**	0.72 ± 0.01	0.72 ± 0.01	0.72 ± 0.01	0.72 ± 0.01	0.72 ± 0.00	0.72±0.01	0.56±0.01
	30%	**0.94 ± 0.01**	0.82 ± 0.03	0.74 ± 0.02	0.74 ± 0.03	0.72 ± 0.01	0.72 ± 0.00	0.72±0.002	0.59±0.04
DNA	50%	**0.95 ± 0.00**	0.93 ± 0.00	0.88 ± 0.02	0.85 ± 0.00	0.92 ± 0.00	0.72 ± 0.00	0.72±0.002	0.72±0.002

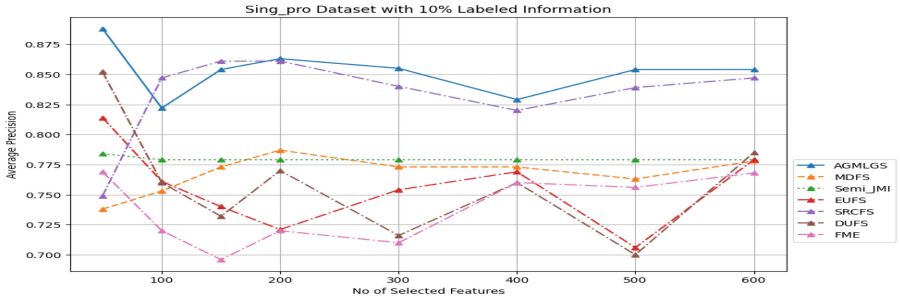
Table 3. Average Run Time in seconds

Dataset	AGMLGS	MDFS	SRCFS	Semi_JMI	EUFS	DUFS	FME	LC-GODR
Endo_can	0.70	0.65	2.34	4.32	1.37	136.45	**0.13**	65.87
Breast_can	1.93	19.21	35.17	16.66	**1.34**	2231.41	3.74	3.74
Alon	0.97	1.17	5.05	5.11	0.28	179.18	**0.16**	31.87
Globun	29.30	27.72	62.00	18.67	**1.76**	2289.01	3.72	0.77
Pro_can	**0.02**	**0.02**	0.97	0.94	1.55	30.98	0.09	0.39
Sing_pro	140.61	111.10	**4.93**	34.77	13.34	3753.96	15.97	OM
Gravier	2.58	2.15	28.74	9.47	1.57	395.50	**0.44**	114.48
Lung_can	0.64	0.57	1.30	5.14	0.97	124.03	**0.12**	166.03
DNA	2.91	20.52	36.94	7.23	7.81	164.56	**0.67**	0.71

can lead to optimal performance without the necessity for manual feature selection and testing. In the Globun dataset, the highest average precision is achieved with 200 feature selections, while in the Sing_pro dataset it is attained with 50



(a) Globun



(b) Sing_pro

Fig. 5. Average Precision achieved by various algorithms relative to the chosen features with 10% labeled Information.

feature selections. This performance is reported as the best average precision when compared to all other algorithms under consideration.

5 Conclusions and Prospective

The paper presents a novel approach to gene selection using semi-supervised adaptive graph manifold learning (AGMLGS). AGMLGS operates within a unified model, unlike most feature selection algorithms that typically involve separate phases for feature selection and classification. Unlike conventional algorithms, which rely on empirical feature selection, AGMLGS dynamically learns the graph and selects optimal features during operation via a feature weights matrix and considering its $L_{2,1}$ norm.

To explore data relationships, we utilize a similarity graph that is reconstructed in each iteration. Our evaluation encompasses multiple datasets and comparisons with seven state-of-the-art algorithms. Consistently, the results demonstrate that our proposed algorithm outperforms others in terms of average precision while maintaining reasonable execution speed across the majority of scenarios. These findings highlight the effectiveness and efficiency of our novel gene selection approach.

Although our current research focuses on one specific area, combining data from various domains, especially in multi-omics scenarios, poses a challenge that we can explore in future studies. Investigating this fusion could provide valuable insights and enhance the practical applications of our approach.

References

1. Govindarajan, R., Duraiyan, J., Kaliyappan, K., Palanisamy, M.: Microarray and its applications. J. Pharm. Bioall. Sci. **4**(Suppl 2), S310 (2012)
2. Tang, C., Cao, L., Zheng, X., Wang, M.: Gene selection for microarray data classification via subspace learning and manifold regularization. Med. Biol. Eng. Comput. **56**, 1271–1284 (2018)
3. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. **3**(02), 185–205 (2005)
4. Zheng, X., Zhang, C.: Gene selection for microarray data classification via dual latent representation learning. Neurocomputing **461**, 266–280 (2021)
5. Almutiri, T., Saeed, F.: Review on feature selection methods for gene expression data classification. In: Saeed, F., Mohammed, F., Gazem, N. (eds.) IRICT 2019. AISC, vol. 1073, pp. 24–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-33582-3_3
6. Wu, X., Chen, H., Li, T., Wan, J.: Semi-supervised feature selection with minimal redundancy based on local adaptive. Appl. Intell. **51**(11), 8542–8563 (2021). https://doi.org/10.1007/s10489-021-02288-4
7. Lai, J., Chen, H., Li, T., Yang, X.: Adaptive graph learning for semi-supervised feature selection with redundancy minimization. Inf. Sci. **609**, 465–488 (2022)
8. Wang, J., Zhao, R., Wang, Y., Zheng, C., Kong, J., Yi, Y.: Locality constrained graph optimization for dimensionality reduction. Neurocomputing **245**, 55–67 (2017)
9. Lee, N., Lee, J., Park, C.: Self-supervised graph representation learning via positive mining. Inf. Sci. **611**, 476–493 (2022)
10. Peng, Y., Lu, B.-L., Wang, S.: Enhanced low-rank representation via sparse manifold adaption for semi-supervised learning. Neural Netw. **65**, 1–17 (2015)
11. Wang, Y., Meng, Y., Li, Y., Chen, S., Fu, Z., Xue, H.: Semi-supervised manifold regularization with adaptive graph construction. Pattern Recogn. Lett. **98**, 90–95 (2017)
12. Wu, G., Zheng, R., Tian, Y., Liu, D.: Joint ranking SVM and binary relevance with robust low-rank learning for multi-label classification. Neural Netw. **122**, 24–39 (2020)
13. Lv, S., Shi, S., Wang, H., Li, F.: Semi-supervised multi-label feature selection with adaptive structure learning and manifold learning. Knowl.-Based Syst. **214**, 106757 (2021)
14. Nie, F., Wang, Z., Wang, R., Li, X.: Adaptive local embedding learning for semi-supervised dimensionality reduction. IEEE Trans. Knowl. Data Eng. **34**(10), 4609–4621 (2021)
15. Jiang, B., et al.: Semi-supervised multiview feature selection with adaptive graph learning. IEEE Trans. Neural Networks Learn. Syst. (2022)
16. Liu, Z., Shi, K., Zhang, K., Ou, W., Wang, L.: Discriminative sparse embedding based on adaptive graph for dimension reduction. Eng. Appl. Artif. Intell. **94**, 103758 (2020)

17. Zhang, J., Luo, Z., Li, C., Zhou, C., Li, S.: Manifold regularized discriminative feature selection for multi-label learning. Pattern Recogn. **95**, 136–150 (2019)
18. Sechidis, K., Brown, G.: Simple strategies for semi-supervised feature selection. Mach. Learn. **107**(2), 357–395 (2018)
19. Huang, D., Cai, X., Wang, C.-D.: Unsupervised feature selection with multi-subspace randomization and collaboration. Knowl.-Based Syst. **182**, 104856 (2019)
20. Wang, S., Tang, J., Liu, H.: Embedded unsupervised feature selection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
21. Lim, H., Kim, D.-W.: Pairwise dependence-based unsupervised feature selection. Pattern Recogn. **111**, 107663 (2021)
22. Nie, F., Xu, D., Tsang, I.W.-H., Zhang, C.: Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction. IEEE Trans. Image Process. **19**(7), 1921–1932 (2010)
23. Alon, U.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. **96**, 6745–6750 (1999)
24. Gravier, E., et al.: A prognostic DNA signature for T1T2 node-negative breast cancer patients. Genes Chromosom. Can. **49**, 1125–1125 (2010)
25. West, M.M., et al.: Predicting the clinical status of human breast cancer by using gene expression profiles. Proc. Natl. Acad. Sci. U.S.A. **98**(20), 11462–11467 (2001)
26. Golub, T.R., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**, 531–537 (1999)
27. Singh, D., et al.: Gene expression correlates of clinical prostate cancer behavior. Can. Cell **1**, 203–209 (2002)
28. Gordon, G.J.G., et al.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Can. Res. **62**, 4963–4967 (2002)

Author Index

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15301, pp. 469–470, 2025.
https://doi.org/10.1007/978-3-031-78107-0