

Mufti Mahmud · Maryam Daborjeh ·
Kevin Wong · Andrew Chi Sing Leung ·
Zohreh Daborjeh · M. Tanveer (Eds.)

LNCS 15296

Neural Information Processing

31st International Conference, ICONIP 2024
Auckland, New Zealand, December 2–6, 2024
Proceedings, Part XI

11
Part XI

ICONIP
2024



Springer

Founding Editors

Gerhard Goos

Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*

Wen Gao, *Peking University, Beijing, China*

Bernhard Steffen , *TU Dortmund University, Dortmund, Germany*

Moti Yung , *Columbia University, New York, NY, USA*

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.

Mufti Mahmud · Maryam Doborjeh ·
Kevin Wong · Andrew Chi Sing Leung ·
Zohreh Doborjeh · M. Tanveer
Editors

Neural Information Processing

31st International Conference, ICONIP 2024
Auckland, New Zealand, December 2–6, 2024
Proceedings, Part XI

Editors

Mufti Mahmud 
King Fahd University of Petroleum
and Minerals
Dhahran, Saudi Arabia

Kevin Wong 
Murdoch University
Murdoch, WA, Australia

Zohreh Doborjeh 
Auckland University of Technology
Auckland, New Zealand

Maryam Doborjeh 
Auckland University of Technology
Auckland, New Zealand

Andrew Chi Sing Leung 
City University of Hong Kong
Kowloon, Hong Kong

M. Tanveer 
Indian Institute of Technology Indore
Indore, Madhya Pradesh, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-981-96-6605-8

ISBN 978-981-96-6606-5 (eBook)

<https://doi.org/10.1007/978-981-96-6606-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Singapore Pte Ltd. 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Preface

Welcome to the 31st International Conference on Neural Information Processing (ICONIP 2024) of the Asia-Pacific Neural Network Society (APNNS), held in Auckland, New Zealand, December 2–6, 2024.

The mission of the Asia-Pacific Neural Network Society is to promote active interactions among researchers, scientists, and industry professionals who are working in neural networks and related fields in the Asia-Pacific region. APNNS has Governing Board Members from 13 countries/regions - Australia, China, Hong Kong, India, Japan, Malaysia, New Zealand, Singapore, South Korea, Qatar, Taiwan, Thailand, and Turkey. The society's flagship annual conference is the International Conference on Neural Information Processing (ICONIP). The ICONIP conference aims to provide a leading international forum for researchers, scientists, and industry professionals who are working in neuroscience, neural networks, deep learning, and related fields to share their new ideas, progress, and achievements.

ICONIP 2024 received 1301 papers, of which 318 papers were accepted for publication in Lecture Notes in Computer Science (LNCS), representing an acceptance rate of 24.44% and reflecting the increasingly high quality of research in neural networks and related areas. The conference focused on four main areas, i.e., “Theory and Algorithms”, “Cognitive Neuroscience”, “Human-Centered Computing”, and “Applications”. All the submissions were rigorously reviewed by the conference Programme Committee (PC), comprising 767 PC members, and they ensured that every paper had at least three high-quality single-blind reviews. Each paper was further reviewed by at least one chair.

We would like to take this opportunity to thank all the authors for submitting their papers to our conference, and our great appreciation goes to the PC members and the reviewers who devoted their time and effort to our rigorous peer-review process; their insightful reviews and timely feedback ensured the high quality of the papers accepted for publication. We hope you enjoyed the research program at the conference.

March 2025

Mufti Mahmud
Maryam Doborjeh
Kevin Wong
Andrew Chi Sing Leung
Zohreh Doborjeh
M. Tanveer

Organisation

Honorary Chairs

Nikola Kasabov

Derong Liu

Seiichi Ozawa

Jonathan Chan

Auckland University of Technology, New Zealand
Southern University of Science and Technology,
China

Kobe University, Japan

King Mongkut's University of Technology
Thonburi, Thailand

Conference Chairs

Maryam Daborjeh

Mufti Mahmud

Michael Witbrock

Auckland University of Technology, New Zealand
King Fahd University of Petroleum and Minerals,
Saudi Arabia

University of Auckland, New Zealand

Program Chairs

Kevin Wong

Andrew Chi Sing Leung

Zohreh Daborjeh

M. Tanveer

Murdoch University, Australia

City University of Hong Kong, China

University of Auckland, New Zealand

IIT Indore, India

Finance Chairs

Saori Tanaka

Terry Brydon

University of Electro-Communications, Japan

Auckland University of Technology, New Zealand

Competition Chairs

Paul S. Pang

Goh Wen Bin Wilson

Federation University Australia, Australia

Nanyang Technological University, Singapore

Special Session Chairs

Tingwen Huang
Wei-Neng Chen
Edmund Lai

Texas A&M University at Qatar, Qatar
South China University of Technology, China
Auckland University of Technology, New Zealand

Tutorial and Workshop Chairs

Alex Sumich
Kenneth Johnson
Marzia Hoque Tania
Akbar Ghobakhloo

Nottingham Trent University, UK
Auckland University of Technology, New Zealand
University of New South Wales, Australia
Auckland University of Technology, New Zealand

Award Chairs

Kenji Doya
Tanja Mitrovic

Okinawa Institute of Science and Technology,
Japan
University of Canterbury, New Zealand

Keynote Chairs

Zeng-Guang Hou
Amir H. Gandomi
Stephen Thorpe
Leanne Bint
Jing Ma
Mahsa Mohaghegh
Wei Qi Yan
Amir Hussain
Roopak Sinha

Chinese Academy of Sciences, China
University of Technology Sydney, Australia
Auckland University of Technology, New Zealand
Edinburgh Napier University, UK
Deakin University, Australia

Publicity Chairs

El-Sayed M. El-Alfy
David Brown
Siddhartha Bhattacharyya

King Fahd University of Petroleum and Minerals,
Saudi Arabia
Nottingham Trent University, UK
VSB Technical University of Ostrava, Czech
Republic

Hyeyoung Park	Kyungpook National University, South Korea
Qinmin Yang	Zhejiang University, China
Chu Kiong Loo	University of Malaya, Malaysia
Cosimo Ieracitano	Mediterranean University of Reggio Calabria, Italy
Bernhard Pfahringer	University of Waikato, New Zealand

Local Organising Chairs

Tek Tjing Lie	Auckland University of Technology, New Zealand
Felix Tan	Auckland University of Technology, New Zealand
Minh Nguyen	Auckland University of Technology, New Zealand
Yvonne Chan Cashmore	Auckland University of Technology, New Zealand
Peter Chong	Auckland University of Technology, New Zealand
Reza Enayatollahi	Toi Ohomai Institute of Technology, New Zealand
Selina Nihalani-Sharma	Auckland University of Technology, New Zealand
Janie van Woerden	Auckland University of Technology, New Zealand

International Advisory Committee

Giancarlo Fortino	University of Calabria, Italy
Hamido Fujita	Iwate Prefectural University, Japan
Shariful Islam	Deakin University, Australia
Tianzi Jiang	Institute of Automation, CAS, China
M. Shamim Kaiser	Jahangirnagar University, Bangladesh
Joarder Kamruzzaman	Federation University Australia, Australia
Francesco Carlo Morabito	University Mediterranea of Reggio Calabria, Italy
Stefano Panzeri	University Medical Center Hamburg-Eppendorf, Germany
Hanchuan Peng	SEU-Allen Institute for Brain & Intelligence, China
Nelishia Pillay	University of Pretoria, South Africa

Technical Program Committee

Hamid Abbasi	University of Auckland, New Zealand
Abdullah Abdullah	Universiti Teknologi PETRONAS, Malaysia
Hazel Abraham	Auckland University of Technology, New Zealand
Sonali Agarwal	IIIT Allahabad, India

Nehal Ahmad	Indian Institute of Technology Indore, India
Saad Bin Ahmed	RPTU, Germany
Toshiaki Aida	Okayama University, Japan
Aisha Ajmal	Open Polytechnic, New Zealand
Mushir Akhtar	IIT Indore, India
Zaid Al-Tameemi	Toi Ohomai Institute of Technology, New Zealand
Shafiq Alam	Massey University, New Zealand
Wajahat Ali	Aligarh Muslim University, India
Jumabek Alikhanov	HumbleBeeAI, South Korea
Fares Alkhawaja	Concordia University, Canada
Nathan Allen	Auckland University of Technology, New Zealand
Fahad Alvi	University of York, UK
Goran Andonovski	University of Ljubljana, Slovenia
Yuki Aoki	Osaka University, Japan
Sabri Arik	Istanbul University, Turkey
Tetsuya Asai	Hokkaido University, Japan
Mohammad Asif	IIIT Allahabad, India
G. Avinash	National Institute of Occupational Health, India
Helena Bahrami	Auckland University of Technology, New Zealand
Muhammad Zeeshan Baig	Macquarie University, Australia
Tao Ban	National Institute of Information and Communications Technology, Japan
Sourasekhar Banerjee	Umeå University, Sweden
Qiming Bao	University of Auckland, New Zealand
Soubhagya Barpanda	VIT-AP University, India
Arindam Basu	University of Canterbury, New Zealand
Boris Bačić	Auckland University of Technology, New Zealand
Aymen Ben Said	University of Regina, Canada
Krishan Berwal	Military College of Telecommunication Engineering, India
M. Agni Catur Bhakti	Sampoerna University, Indonesia
Raghavendra Bhalerao	IITRAM, India
Siddhartha Bhattacharyya	RCC Institute of Information Technology, India
Dasari Bhulakshmi	VIT, India
Ying Bi	Victoria University of Wellington, New Zealand
Jacques Blanc-Talon	DGA TA, France
Christian Bohn	University of Wuppertal, Germany
Phil Bones	University of Canterbury, New Zealand
Yee Ling Boo	RMIT University, Australia
Larbi Boubchir	University of Paris 8, France
Sally Britnell	Auckland University of Technology, New Zealand
Chenyang Bu	Hefei University of Technology, China

Sugam Budhraja	Auckland University of Technology, New Zealand
Michael Bunn	Auckland University of Technology, New Zealand
Jérémie Cabessa	Panthéon-Assas University Paris II, France
Wanqiang Cai	Nanjing University of Finance & Economics, China
Weidong Cai	University of Sydney, Australia
Cristian S. Calude	University of Auckland, New Zealand
Yuan Cao	Beijing University of Posts and Telecommunications, China
Elisa Capecci	Auckland University of Technology, New Zealand
Debasrita Chakraborty	Indian Statistical Institute, Kolkata, India
Srinivasa Chakravarthy	IIT Madras, India
Stephan Chalup	University of Newcastle, Australia
Victor Chan	Tsinghua University, China
Ritesh Chandra	IIT Allahabad, India
Siripinyo Chantamunee	Walailak University, Thailand
Gu Chaochen	Shanghai Jiao Tong University, China
Nisha Chauhan	Indian Institute of Technology Roorkee, India
Zaineb Chelly Dagdia	Université Paris-Saclay, France
Bang Chen	Ningbo University, China
Dongjie Chen	University of Chinese Academy of Sciences, China
Jianyong Chen	Shenzhen University, China
Jinpeng Chen	Beijing University of Posts & Telecommunications, China
Kecheng Chen	University of Science and Technology of China, China
Ling Chen	Southwest University, China
Lyu Chen	Shandong Normal University, China
Wei-Neng Chen	South China University of Technology, China
Xiaoqing Chen	University of Chinese Academy of Sciences, China
Xinrui Chen	Tsinghua University, China
Zhewei Chen	Australian National University, Australia
Zhi Chen	University of Electronic Science and Technology of China, China
Long Cheng	Institute of Automation, CAS, China
Xingfu Cheng	Qufu Normal University, China
Ziqi Cheng	Chongqing University, China
Sung-Bae Cho	Yonsei University, South Korea
Chak Fong Chong	Macao Polytechnic University, China
Peter Han Joo Chong	Auckland University of Technology, New Zealand
Eashita Chowdhury	Indian Institute of Technology Kharagpur, India

Mamadou B. H. Cissoko	University of Strasbourg, France
Raphael Couturier	Marie and Louis Pasteur University, France
Stephen Cox	Residio, New Zealand
Brian Cusack	Auckland University of Technology, New Zealand
Jianhua Dai	Hunan Normal University, China
Shaoxiang Dang	Nagoya University, Japan
Anik Das	St. Francis Xavier University, Canada
Santosh Das	OmDayal Group of Institutions, India
Subham Das	Indian Institute of Technology Madras, India
Sudhansu Bala Das	NIT Rourkela, India
Marcílio De Souto	University of Orléans, France
Neda Deljavan	Institute for Advanced Studies in Basic Sciences, Iran
Jeremiah D. Deng	University of Otago, New Zealand
Nagaraj V. Dharwadkar	National Institute of Technology, Warangal, India
Kumar Dheenadayalan	Qualcomm, USA
Johannes Dimyadi	University of Auckland, New Zealand
Yuxin Ding	Harbin Institute of Technology, China
Maryam Doborjeh	Auckland University of Technology, New Zealand
Zohreh Doborjeh	University of Auckland, New Zealand
Bin Dong	Ricoh Software Research Center, China
Ninghua Dong	Beijing Jiaotong University, China
Ruiqi Dong	South China Normal University, China
Jordan Douglas	University of Auckland, New Zealand
Kenji Doya	OIST, Japan
Haiwen Du	Harbin Institute of Technology, China
Fuqing Duan	Beijing Normal University, China
Piotr Duda	Częstochowa University of Technology, Poland
Thao Duong	Murdoch University, Australia
Varun Dutt	Indian Institute of Technology Mandi, India
Pratik Dutta	Indian Institute of Technology Patna, India
Mansoor Ebrahim	Iqra University, Pakistan
El-Sayed M. El-Alfy	King Fahd University of Petroleum and Minerals, Saudi Arabia
Reza Enayatollahi	Toi Ohomai Institute of Technology, New Zealand
Farnoush Falahatraftar	Polytechnique Montréal, Canada
Chenyou Fan	South China Normal University, China
Jiangtao Fan	Durham University, UK
Junyuan Fang	City University of Hong Kong, China
Sen Fang	University of Victoria, Australia
Yifei Fang	University of Chinese Academy of Sciences, China

Zhiyuan Fang	University of Waikato, New Zealand
Muhammad Farhan	Glocal University, India
Jiaxuan Feng	Southwest University of Science and Technology, China
Yong Feng	Chongqing University, China
Jaco Fourie	Lincoln Agritech Ltd, UK
Frieyadie Frieyadie	STMIK Nusa Mandiri, Indonesia
Junjie Fu	Southeast University, China
Taoran Fu	Hunan University, China
Xiping Fu	University of Otago, New Zealand
Yulin Fu	University of Auckland, New Zealand
Fumiyo Fukumoto	University of Yamanashi, Japan
Haitao Gan	Hubei University of Technology, China
Mudasir Ganaie	Indian Institute of Technology Indore, India
Varun Ganjigunte Prakash	CogniAble, India
Qian Gao	Qilu University of Technology, China
Terry Gao	Moreton Bay City Council, Australia
Xinyi Gao	Auckland University of Technology, New Zealand
Xizhan Gao	University of Jinan, China
Chandan Gautam	I2R, A*STAR, Singapore
Liang Ge	Chongqing University, China
Mengmeng Ge	University of Canterbury, New Zealand
Trevor Gee	University of Auckland, New Zealand
Mingyang Geng	National University of Defense Technology, China
Thanawit Gerdprasert	Yamaguchi University, Japan
Akbar Ghobakhloo	Auckland University of Technology, New Zealand
Hamid Gholamhosseini	Auckland University of Technology, New Zealand
Ashish Ghosh	Indian Statistical Institute, Kolkata, India
Tripti Goel	National Institute of Technology Silchar, India
Zbigniew Gomolka	University of Rzeszow, Poland
Jiaying Gong	Virginia Tech, USA
Rui Gong	Chinese Academy of Sciences, China
Yingjie Gong	Zhejiang University, China
Maanvik Gounder	University of the South Pacific, Fiji
Richard Green	University of Canterbury, New Zealand
Jessica Gu	University of Auckland, New Zealand
Tamil Selvan Gunasekaran	University of Auckland, New Zealand
Long Guo	Sichuan University, China
Ping Guo	Beijing Normal University, China
Qin Guo	Peking University, China
Deepak Gupta	MNNIT Allahabad, India

Harshit Gupta	IIIT Allahabad, India
Tomasz Hachaj	Pedagogical University of Krakow, Poland
Katsuyuki Hagiwara	Mie University, Japan
Masafumi Hagiwara	Keio University, Japan
Luke Hallum	University of Auckland, New Zealand
Chansu Han	National Institute of Information and Communications Technology, Japan
Gao Han	Xidian University, China
Zhang Haoyu	Chinese Academy of Sciences, China
Md. Tarek Hasan	United International University, Bangladesh
Tatsuhiro Hasegawa	University of Fukui, Japan
Hao He	Chinese Academy of Sciences, China
Xing He	Southwest University, China
Xinyang He	UCAS, China
Yangliu He	Beijing University of Posts and Telecommunications, China
Yuting He	University of Nottingham Ningbo, China
Mahshid Helali Moghadam	RISE Research Institutes of Sweden, Sweden
Hugo Hernault	Playtika, Japan
Akira Hirose	University of Tokyo, Japan
Yvonne Hong	Victoria University of Wellington, New Zealand
Yin Hongwei	Huzhou University, China
Adrian Horzyk	University of Krakow, Poland
Amanda Horzyk	University of Edinburgh, UK
Md Zakir Hossain	Australian National University, Australia
Zengguang Hou	Chinese Academy of Sciences, China
Menghao Hu	Pengcheng Laboratory, China
Yan Hu	University of New South Wales, Australia
Zhongyun Hua	Harbin Institute of Technology, China
He Huang	Soochow University, China
Kaizhu Huang	Duke Kunshan University, China
Sheng Huang	Chongqing University, China
Siyu Huang	Nanjing University of Aeronautics and Astronautics, China
Zhen Huang	Shenyang Institute of Computing Technology, China
Carolynne Hultquist	Pennsylvania State University, USA
Aarij Hussaan	Iqra University, Pakistan
David Iclanzan	Sapientia University, Romania
Cosimo Ieracitano	University “Mediterranea” of Reggio Calabria, Italy
Kazushi Ikeda	Nara Institute of Science and Technology, Japan

Radu Tudor Ionescu	University of Bucharest, Romania
Noriyuki Iwane	Hiroshima City University, Japan
Sana Jabbar	Lahore University of Management Sciences, Pakistan
Sapna Jaidka	University of Waikato, New Zealand
Debesh Jha	Northwestern University, USA
Ningning Jia	Beijing University of Posts and Telecommunications, China
Yan Jia	National University of Defense Technology, China
Peng Jiang	Wuhan University, China
Xianwei Jiang	Southwest University of Science and Technology, China
Xuesong Jiang	Qilu University of Technology, China
Chen Jiaqi	Dongguan University of Technology, China
Jin	Xi'an Jiaotong Liverpool University, China
Chunzhen Jin	Northeastern University, USA
Xiaozheng Jin	Qilu University of Technology, China
Vijay John	RIKEN, Japan
Kenneth Johnson	Auckland University of Technology, New Zealand
Seul Jung	Chungnam National University, China
Akbar K.	BITS Pilani, Goa Campus, India
Sweta Kaman	IIT Jodhpur, India
Keiji Kamei	Nishinippon Institute of Technology, Japan
Keisuke Kameyama	University of Tsukuba, Japan
Yoshimi Kamiyama	Aichi Prefectural University, Japan
Joarder Kamruzzaman	Federation University Australia, Australia
Bhavik Kanekar	IIT Mandi, India
Tomoyuki Kaneko	University of Tokyo, Japan
Jun-Su Kang	Kyungpook National University, South Korea
Shin'Ichiro Kanoh	Shibaura Institute of Technology, Japan
Nikola Kasabov	Auckland University of Technology, New Zealand
Arshpreet Kaur	NIT Jalandhar, India
Yoshinobu Kawahara	Osaka University/RIKEN, Japan
Hideaki Kawano	Kyushu Institute of Technology, Japan
Kostiantyn Khabarlak	Dnipro University of Technology, Ukraine
Mehshan Khan	Deakin University, Australia
Atikant Khanna	Auckland University of Technology, New Zealand
Daegyeom Kim	Korea University, South Korea
Jonghong Kim	Kyungpook National University, South Korea
Mutsumi Kimura	Ryukoku University, Japan
Irwin King	Chinese University of Hong Kong, China

Gisela Klette	Auckland University of Technology, New Zealand
Mallika Kliangkhlao	Walailak University, Thailand
Alistair Knott	Victoria University of Wellington, New Zealand
Kunikazu Kobayashi	Aichi Prefectural University, Japan
Rangachary Kommanduri	Indian Institute of Information Technology, Sri City, India
Aneesh Krishna	Curtin University, Australia
Rita Krishnamurthi	Auckland University of Technology, New Zealand
Adam Krzyzak	Concordia University, Canada
Sumant Kulkarni	Zenlabs, Zensar Technologies, India
Estine Kumar	University of the South Pacific, Fiji
Neetesh Kumar	IITR, India
Praveen Kumar	Banaras Hindu University, India
Rakesh Kumar	IIT BHU, India
Swaroop Kumar	L&T Technology Services, India
Chithrangi Kumarasinghe	University of Moratuwa, Sri Lanka
Anuradha Kumari	Indian Institute of Technology Indore, India
Naga Jyothi Kunchala	Massey University, New Zealand
Souraja Kundu	Indian Institute of Technology Guwahati, India
Hiroki Kurashige	Tokai University, Japan
Tomoki Kurikawa	Future University Hakodate, Japan
Kurnianingsih	Politeknik Negeri Semarang, Indonesia
Fatih Kurugollu	University of Sharjah, UAE
Thomas Lacombe	University of Auckland, New Zealand
Hamid Laga	Murdoch University, Australia
Edmund Lai	Auckland University of Technology, New Zealand
Leon Lange	University of California, San Diego, USA
Walter Langelaar	Victoria University of Wellington, New Zealand
Sang Hun Lee	Kookmin University, South Korea
Tet Chuan Lee	Auckland University of Technology, New Zealand
Yuan Lei	University of Chinese Academy of Sciences, China
Chi Sing Leung	City University of Hong Kong, China
Bingxian Li	Heilongjiang University, China
Bo Li	Baidu Inc, China
Fengyu Li	Beijing University of Posts and Telecommunication, China
Hongfei Li	Xinjiang University, China
Jiale Li	University of Auckland, New Zealand
Jun Li	Nanjing Normal University, China
Lining Li	CASIA, China
Maodong Li	Soochow University, China

Mengmeng Li	Zhengzhou University, China
Mengshu Li	University of Toronto, Canada
Mengting Li	Beijing University of Posts and Telecommunications, China
Ming Li	Wuhan University of Technology, China
Peifeng Li	Soochow University, China
Ruifan Li	Beijing University of Posts and Telecommunications, China
Sirui Li	Murdoch University, China
Tieshan Li	Dalian Maritime University, China
Weiwei Li	UESTC, China
Xiaohong Li	Northwest Normal University, China
Yantao Li	Chongqing University, China
Yi Li	Lancaster University, UK
Yinbao Li	Baidu Tech., China
Yun Li	Nanjing University of Posts and Telecommunications, China
Ziwei Li	UCAS, China
Jiawen Liang	City University of Hong Kong, China
Xiaokun Liang	Shenzhen Institute of Advanced Technology, China
Xiaoyi Liang	Tongji University, China
Xu Liang	Harbin Institute of Technology, China
Zhuonan Liang	University of Sydney, Australia
Junfeng Liao	Shanghai University of International Business and Economics, China
Xiao-Cheng Liao	South China University of Technology, China
Jianpeng Lin	Guangdong University of Technology, China
Qiuhua Lin	Dalian University of Technology, China
Yang Lin	University of Sydney, Australia
Baodi Liu	China University of Petroleum, China
Binghao Liu	Beihang University, China
Gangli Liu	Tsinghua University, China
Hanyuan Liu	City University of Hong Kong, China
Jian-Wei Liu	China University of Petroleum, China
Juan Liu	Wuhan University, China
Lei Liu	Northeastern University, USA
Renyang Liu	National University of Singapore, Singapore
Shang Liu	UCAS, China
Weifeng Liu	China University of Petroleum, China
Wen Liu	Chinese University of Hong Kong, China
Xiaoyang Liu	Huazhong University Science & Technology, China

Yang Liu	Fudan University, China
Yanming Liu	Georgia Institute of Technology, USA
Yaozhong Liu	Australian National University, Australia
Zhaoyi Liu	KU Leuven, Belgium
Zhe Liu	Jiangsu University, China
Han Long	National University of Defense Technology, China
Jieting Long	University of Sydney, Australia
Chu Kiong Loo	University of Malaya, Malaysia
Andrew Lowe	Auckland University of Technology, New Zealand
Gewei Lu	Shanghai Jiao Tong University, China
Heng-Yang Lu	Nanjing University, China
Hongtao Lu	Shanghai Jiao Tong University, China
Weihai Lu	Peking University, China
Wenhao Lu	Nanyang Technological University, Singapore
Yu Lu	Shenzhen Technology University, China
Shijie Luan	Chinese Academy of Sciences, China
Fangzhou Luo	McMaster University, Canada
Róisín Luo	University of Galway, Ireland
Siwen Luo	University of Western Australia, Australia
Yuchuan Luo	National University of Defense Technology, China
Raymond Lutui	Auckland University of Technology, New Zealand
Jiancheng Lv	Sichuan University, China
Yuezu Lv	Beijing Institute of Technology, China
Liangfu Lyu	Federation University Australia, Australia
Qingguo Lü	Southwest University, China
Jing Ma	Auckland University of Technology, New Zealand
Jinwen Ma	Peking University, China
Ming Ma	Yeshiva University, USA
Yan Ma	Fudan University, China
Yuqi Ma	Chinese University of Hong Kong, China
Alexei Machado	Pontifical Catholic University of Minas Gerais, Brazil
Jyoti Maggu	Thapar Institute of Engineering and Technology, India
Tariq Mahmood	COMSATS Institute of Information Technology, Pakistan
Mufti Mahmud	King Fahd University of Petroleum and Minerals, Saudi Arabia
Snehashis Majhi	Inria Sophia Antipolis, France
Mishaim Malik	University of Auckland, New Zealand

Mamta	IIT Patna, India
Raquel Marasigan	University of Asia and the Pacific, Philippines
Gerard M. Freixas	Fudan University, China
Stefan Marks	Auckland University of Technology, New Zealand
Archana Mathur	Nitte Meenakshi Institute of Technology, India
Yoshitatsu Matsuda	Seikei University, Japan
Tomas Maul	University of Nottingham Malaysia Campus, Malaysia
Jacek Mańdziuk	Warsaw University of Technology, Poland
Yogendra Meena	Delhi University, India
Yi Mei	Victoria University of Wellington, New Zealand
Erik Meijering	University of New South Wales, Australia
Qing-Xin Meng	China University of Petroleum, China
Alexander Merkin	AUT, New Zealand
Cheng Miao	Guangxi Normal University, China
Ashish Mishra	University of Nevada Las Vegas, USA
Sajib Mistry	Curtin University, Australia
Tanja Mitrovic	University of Canterbury, New Zealand
Seiji Miyoshi	Kansai University, Japan
Mohammadreza Montazerijouybari	Polytechnique Montréal, Canada
Francesco C. Morabito	University of Reggio Calabria, Italy
Satoru Morita	Yamaguchi University, Japan
Mpatisi Moyo	AiTonomy, UK
Taslim Murad	Georgia State University, USA
Shingo Murakami	Chuo University, Japan
Dharmalingam Muthusamy	Bharathiar University, India
Chitrakala Muthuveerappan	Victoria University of Wellington, New Zealand
Muhan Na	Inner Mongolia University, China
Isao Nambu	Nagaoka University of Technology, Japan
Parma Nand	Auckland University of Technology, New Zealand
Ajit Narayanan	Auckland University of Technology, New Zealand
Gokulmuthu Narayanaswamy	IIT Kharagpur, India
Kiyohisa Natsume	Kyushu Institute of Technology, Japan
Azadeh Nazemi	Norwood Systems, Australia
Usman Nazir	Lahore University of Management Sciences, Pakistan
Elena Nechita	Vasile Alecsandri University of Bacau, Romania
Chandra Mohan Singh Negi	Siemens, India
Mehdi Neshat	University of Adelaide, Australia
Kourosh Neshatian	University of Canterbury, New Zealand
Frank Neumann	University of Adelaide, Australia

Bach Nguyen	Victoria University of Wellington, New Zealand
Bao Sinh Nguyen	HUST, Vietnam
Binh P. Nguyen	Victoria University of Wellington, New Zealand
Minh Nguyen	Auckland University of Technology, New Zealand
Quang Vinh Nguyen	Western Sydney University, Australia
Mukku Nisanth Kartheek	National Institute of Technology Warangal, India
Sou Nobukawa	Chiba Institute of Technology, Japan
Anupiya Nugaliyadde	Murdoch University, Australia
Jeongbin Ok	Victoria University of Wellington, New Zealand
Diego Oliva	Universidad de Guadalajara, Spain
Toshiaki Omori	Kobe University, Japan
Sihem Omri	Higher School of Communication of Tunis, Tunisia
Hideaki Orii	Fukuoka University, Japan
Seiichi Ozawa	Kobe University, Japan
Achmad Pahlevi	Auckland University of Technology, New Zealand
Susmita Palmal	Ramgarh Engineering College, India
Guangyuan Pan	Linyi University, China
Wenkai Pan	Linyi University, China
Pankaj Pandey	Indian Institute of Technology, Delhi, India
Jason Pang	Lincoln University, UK
Paresh Kumar Panigrahi	VIT-AP University, India
Dipendra Pant	Norwegian University of Science and Technology, Norway
Hyeyoung Park	Kyungpook National University, South Korea
Dipanjyoti Paul	Indian Institute of Technology Patna, India
Mangor Pedersen	Auckland University of Technology, New Zealand
Siamak Pedrammehr	Deakin University, Australia
Anjie Peng	Southwest University of Science and Technology, China
Nasca Peng	Statistics New Zealand, New Zealand
Yong Peng	Hangzhou Dianzi University, China
Joao Pereira	Imperial College London, UK
Krassie Petrova	Auckland University of Technology, New Zealand
Somnuk Phon-Amnuaisuk	Universiti Teknologi Brunei, Brunei Darussalam
Pasu Poonpakdee	Walailak University, Thailand
Vijay Prakash	Thapar University, India
Narinder Singh Punn	Mayo Clinic, Arizona, USA
Junjie Qi	Guangxi Normal University, China
Weihua Qiang	Tianjin University, China
Yu Qiao	Shanghai Jiao Tong University, China
Sitian Qin	Harbin Institute of Technology at Weihai, China

Xiaoyang Qu	Huazhong University of Science and Technology, China
Abdul Quadir	IIT Indore, India
Uday Kiran Rage	University of Aizu, Japan
Krishna Raghuwaiya	University of the South Pacific, Fiji
Ibrahim Rahman	Open Polytechnic of New Zealand, New Zealand
Jessica Rahman	University of Dhaka, Bangladesh
Shri Rai	Murdoch University, Australia
Surbhi Raj	Indian Institute of Technology Patna, India
K. Ramakrishnan	Auckland University of Technology, New Zealand
R. Kanesaraj Ramasamy	Multimedia University, Malaysia
Deepak Ranjan Nayak	Malaviya National Institute of Technology, Australia
Rabia Naseer Rao	University of Auckland, New Zealand
Munish Rathee	Auckland University of Technology, New Zealand
Ramesh Rayudu	Victoria University of Wellington, New Zealand
Khalid Raza	Jamia Millia Islamia, India
Jianfeng Ren	University of Nottingham Ningbo, China
Minsi Ren	Chinese Academy of Sciences, China
Shao Renrong	East China Normal University, China
Tobias Rettenmeier	University of Applied Sciences Heilbronn, Germany
Young Ju Rho	Tech University of Korea, South Korea
Daniel Riccio	University of Naples Federico II, Italy
Rishabh	University of Delhi, India
Horacio Gonzalez	Universidad de Guanajuato, Mexico
Gargi Roy	Brunel University London, UK
Kaushik Roy	West Bengal State University, India
Muhammad Fakhru Rozi	National Institute of Information and Communications Technology, Japan
Ji Ruan	Auckland University of Technology, New Zealand
Khairun Saddami	Universitas Syiah Kuala, Indonesia
Michał Sadowski	Jagiellonian University, Poland
Amit Kumar Sah	South Asian University, India
Pranab Sahoo	Indian Institute of Technology Patna, India
Naveen Saini	Indian Institute of Information Technology Allahabad, India
Ken Saito	Nihon University, Japan
Toshimichi Saito	Hosei University, Japan
Md Sajid	IIT Indore, India
Ko Sakai	University of Tsukuba, Japan
Nazmus Sakib	Ahsanullah University of Science & Technology, Bangladesh

Rohit Salgotra	AGH University of Krakow, Poland
Michel Salomon	IUT Belfort-Montbéliard, France
Toshikazu Samura	Yamaguchi University, Japan
Xue Sang	Northwestern Polytechnical University, China
Takashi Sano	Tokyo University, Japan
Yassine Saoudi	Faculté des Sciences de Tunis, Tunisia
Naoyuki Sato	Future University Hakodate, Japan
Eri Sato-Shimokawara	Tokyo Metropolitan University, Japan
Seiya Satoh	Tokyo Institute of Technology, Japan
Wojciech Sałabun	West Pomeranian University of Technology, Poland
Erich Schikuta	University of Vienna, Austria
Eric Scott	MITRE Corporation, USA
Mahdi Setayesh	Microsoft Inc., USA
Noushath Shaffi	Sultan Qaboos University, Oman
Nida Shahab	Auckland University of Technology, New Zealand
Reza Shahamiri	University of Auckland, New Zealand
Jiaxing Shang	Chongqing University, China
Peng Shao	Jiangxi Agricultural University, China
Zhenzhou Shao	Capital Normal University, China
Sourabh Sharma	Avantika University, India
Megha Sharma	Indian Institute of Technology Mandi, India
Swakkhar Shatabda	BRAC University, Bangladesh
Hualei Shen	Henan Normal University, China
Zhixiang Shen	UESTC, China
Yin Sheng	Huazhong University of Science and Technology, China
Yongpan Sheng	Southwest University, China
Pumeng Shi	University of British Columbia, Canada
Qiushi Shi	NTU, Singapore
Xinxin Shi	Changchun University of Science and Technology, China
Hiroki Shibata	Tokyo Metropolitan University, Japan
Hayaru Shouno	University of Electro-Communications, Japan
Jiang Shuyu	Sichuan University, India
Shijing Si	Duke University, USA
Jiten Sidhpura	Sardar Patel Institute of Technology, India
Rangika Silva	Queensland University of Technology, Australia
Simeon Simoff	Western Sydney University, Australia
David Simpson	Massey University, New Zealand
Balkaran Singh	Auckland University of Technology, New Zealand
Om Singh	National Institute of Technology Patna, India

Shashank Singh	Kanpur Institute of Technology, India
Roopak Sinha	Deakin University, Australia
Soumen Sinha	Mahindra University, India
Stephen Skalicky	Victoria University of Wellington, New Zealand
Ferdous Sohel	Murdoch University, Australia
Upika Somaratne	Murdoch University, Australia
Aiguo Song	Southeast University, China
Chao Song	Zhejiang Gongshang University, China
Haohao Song	Xiamen University, China
Liang Song	Fudan University, China
Xianfeng Song	South China University of Technology, China
Laxmi Soni	AKS University, India
Paul Sowman	Auckland University of Technology, New Zealand
Aleksei Staroverov	MIPT, Russia
Amy Stewart	University of Waikato, New Zealand
Martin Stommel	Auckland University of Technology, New Zealand
Jianbo Su	Shanghai Jiao Tong University, China
Xinyan Su	University of Chinese Academy of Sciences, China
Yila Su	Inner Mongolia University of Technology, China
Zhixun Su	Dalian University of Technology, China
Badri Narayan Subudhi	Indian Institute of Technology Jammu, India
Toshiharu Sugawara	Waseda University, Japan
John Sum	National Chung Hsing University, China
Alexander Sumich	Nottingham Trent University, UK
Hao Sun	Nankai University, China
Shiwen Sun	Inner Mongolia University, China
Shuo Sun	Inner Mongolia University, China
Tao Sun	Beihang University, China
Zhanquan Sun	University of Shanghai for Science and Technology, China
Suryavardan Suresh	New York University, USA
Kanata Suzuki	Fujitsu Limited, Japan
Satoshi Suzuki	NTT Yokosuka, Japan
Yoshimi Suzuki	University of Yamanashi, Japan
Mikołaj Śląpiński	University of Wrocław, Poland
Izak Tait	Auckland University of Technology, New Zealand
Murtaza Taj	Lahore University of Management Sciences, Pakistan
Norikazu Takahashi	Okayama University, Japan
Hiroshige Takeichi	RIKEN, Japan
Takashi Takekawa	Kogakuin University, Japan

Hiroshi Tamura	Osaka University, Japan
Christine Nya-Ling Tan	Massey University, New Zealand
Chunyu Tan	Anhui University, China
Renzo Roel Tan	Nara Institute of Science and Technology, Japan
Ying Tan	Peking University, China
Zhengguang Tan	Guangdong University of Technology, China
Takuma Tanaka	Shiga University, China
Fengxiao Tang	Tohoku University, Japan
Haoran Tang	Beijing University of Posts and Telecommunications, China
Maolin Tang	Queensland University of Technology, Australia
Yang Tang	East China University of Science and Technology, China
Yihang Tang	Chongqing University of Posts and Telecommunications, China
Zerui Tang	Xiamen University, China
M. Tanveer	Indian Institute of Technology, Indore, India
Zerui Tao	Tokyo University of Agriculture and Technology, China
Jules-Raymond Tapamo	University of KwaZulu-Natal, South Africa
Prithvi Tarale	University of Massachusetts Amherst, USA
Shuhei Tarashima	NTT Communications Corporation, Japan
Yassin Terraf	Mohammed VI Polytechnic University, Morocco
Putthiporn Thanathamathee	Walailak University, Thailand
Veerakumar Thangaraj	National Institute of Technology Goa, India
Chuan Tian	University of Canterbury, New Zealand
Hao Tian	Zhejiang University of Technology, China
Aruna Tiwari	IIT Indore, India
Sadhana Tiwari	Galgotias College of Engineering & Technology, India
Ewaryst Tkacz	Silesian University of Technology, Poland
Kar-Ann Toh	Yonsei University, South Korea
Farank Tohidi	Charles Sturt University, Australia
Cong Tran	QUOC, Vietnam
Chidentree Treesatayapun	Walailak University, Thailand
Richa Tripathi	Washington University in St. Louis, USA
Enmei Tu	Shanghai Jiao Tong University, China
Nitin Tyagi	IIT Roorkee, India
Hamid Usefi	Memorial University of Newfoundland, Canada
Alireza Valizadeh	Universitat de les Illes Balears, Spain
Manju Vallayil	Auckland University of Technology, New Zealand
Maryam Var Naseri	Victoria University of Wellington, New Zealand

Matthieu Vignes	Massey University, New Zealand
Nobuhiko Wagatsuma	Toho University, Japan
Shanchuan Wan	University of Tokyo, Japan
Tao Wan	Beihang University, China
Wang	City University of Hong Kong, China
Bin Wang	Nanjing University of Finance & Economics, China
Binqiang Wang	Chinese Academy of Sciences, China
Chao Wang	China Academy of Railway Sciences Corporation Limited, China
Chen Wang	Chinese Academy of Sciences, China
Chengliang Wang	Chongqing University, China
Chunshi Wang	Guilin University of Electronic Technology, China
Guanjin Wang	Murdoch University, Australia
Haizhou Wang	Sichuan University, China
Han Wang	Xidian University, China
Hao Wang	Shenzhen University, China
Haowen Wang	Alipay, Ant Group, China
Jiale Wang	Zhejiang Sci-Tech University, China
Jianzong Wang	Ping An Technology Co., Ltd., China
Jun-Wei Wang	University of Science and Technology Beijing, China
Kaier Wang	Volpara Health Technologies Ltd., New Zealand
Liang Wang	Beijing University of Technology, China
Liang Wang	Huazhong University of Science and Technology, China
Liantao Wang	Hohai University, China
Ming Hui Wang	China University of Petroleum, China
Nana Wang	Jiangsu Normal University, China
Peijun Wang	Anhui Normal University, China
Rui Wang	Ningbo University, China
Ruiying Wang	Southwest University of Science and Technology, China
Xianzhi Wang	University of Technology Sydney, Australia
Xiao Wang	Chongqing Normal University, China
Xiulin Wang	Dalian University of Technology, China
Yadi Wang	Henan University, China
Ye Wang	National University of Defense Technology, China
Yong Wang	Southeast University, China
Yongyu Wang	JD Logistics, China
Zhenni Wang	City University of Hong Kong, China

Zhongsheng Wang	University of Auckland, New Zealand
Zi-Peng Wang	University of Jinan, New Zealand
Ziwei Wang	Huazhong University of Science and Technology, China
Yoshikazu Washizawa	University of Electro-Communications, China
Fengchen Wei	University of Sussex, UK
Hongxi Wei	Inner Mongolia University, China
Alastair Wells	Auckland University of Technology, New Zealand
Guanghui Wen	RMIT University, Australia
Junjie Wen	Chinese University of Hong Kong, China
Jinta Weng	Chinese Academy of Sciences, China
Lei Wenjie	City University of Hong Kong, China
David White	Auckland University of Technology, New Zealand
Tom White	Victoria University of Wellington, New Zealand
Savindi Wijenayaka	University of Auckland, New Zealand
Michael Witbrock	University of Auckland, New Zealand
Hiutung Wong	City University of Hong Kong, China
Ka-Chun Wong	City University of Hong Kong, China
Kevin Wong	Murdoch University, Australia
Boqi Wu	University of Wuppertal, Germany
Chao Wu	Zhejiang University, China
Chengkun Wu	National University of Defense Technology, China
Haha Wu	Xinjiang University, China
Song Wu	Hainan Tropical Ocean University, China
Xianze Wu	Shanghai Jiao Tong University, China
Zipeng Wu	University of Birmingham, UK
Zhiqiu Xia	Rutgers University, USA
Ziwei Xiang	Chinese Academy of Sciences, China
Jinying Xiao	Changsha University of Science & Technology, China
Qiang Xiao	Huazhong University of Science and Technology, China
Xi Xiao	University of Alabama at Birmingham, USA
Shiwen Xie	Central South University, China
Zaipeng Xie	Hohai University, China
Yucheng Xing	Stony Brook University, USA
Wang Xinshen	Guangdong University of Foreign Studies, China
Wenxin Xiong	City University of Hong Kong, China
Chao Xu	Changsha University of Science and Technology, China
Fanchao Xu	University of Science and Technology of China, China

Jianhua Xu	Nanjing Normal University, China
Jinhua Xu	East China Normal University, China
Lele Xu	Southeast University, China
Qing Xu	Tianjin University, China
Xinyue Xu	Hong Kong University of Science and Technology, China
Junyu Xuan	University of Technology Sydney, Australia
Felix Yan	Victoria University of Wellington, New Zealand
Shankai Yan	Hainan University, China
Teng Yan	Shenzhen University, China
Weiqi Yan	AUT, New Zealand
Da Yang	Beijing University of Aeronautics and Astronautics, China
Haitian Yang	Chinese Academy of Sciences, China
Jie Yang	Shanghai Jiao Tong University, China
Mengyu Yang	Beijing University of Posts and Telecommunications, China
Minghao Yang	Chinese Academy of Sciences, China
Peipei Yang	Chinese Academy of Science, China
Peng Yang	Chongqing Three Gorges University, China
Qinmin Yang	Zhejiang University, China
Wei Yang	University of Chinese Academy of Sciences, China
Yanfeng Yang	South China University of Technology, China
Zhan Yang	Central South University, China
Zhikai Yang	KTH Royal Institute of Technology, Sweden
Wangshu Yao	Soochow University, China
Yifeng Yao	University of South China, China
Yun Ye	Intel, USA
Wang Yingying	Shenyang Aerospace University, China
Yongmin Yoo	Macquarie University, Australia
Shuyuan You	Tianjin University, China
Dianzhi Yu	Chinese University of Hong Kong, China
Na Yu	Zhejiang University, China
Ping Yu	Nanjing University of Science and Technology, China
Wenxin Yu	Southwest University of Science and Technology, China
Zhibin Yu	Ocean University of China, China
Zhiwen Yu	South China University of Technology, China
Fangfang Yuan	Chinese Academy of Sciences, China
Xin Yuan	Southeast University, China

Dmitry Yudin	Moscow Institute of Physics and Technology, Russia
Chao Yue	University of Chinese Academy of Sciences, China
Xiaodong Yue	Shanghai University, China
Farzana Zahid	University of Waikato, New Zealand
Ruijie Zeng	Chinese Academy of Sciences, China
Weixin Zeng	National University of Defense Technology, China
Xinhua Zeng	Fudan University, China
Zekeng Zeng	Chinese Academy of Sciences, China
Ewa Zeslawska	University of Rzeszow, Poland
Zhiyuan Zha	Renmin University of China, China
Chao Zhang	Shanxi University, China
Chao Zhang	South China Normal University, China
Chenyi Zhang	University of Canterbury, New Zealand
Chong Zhang	Xi'an Jiaotong-Liverpool University, China
Chufan Zhang	Shanghai Jiao Tong University, China
Congwei Zhang	Southeast University, China
Fan Zhang	Shandong Technology and Business University, China
Francis X. Zhang	Durham University, UK
Gaoyan Zhang	Tianjin University, China
Han Zhang	Northwest University, China
Haoyang Zhang	Chongqing University of Posts and Telecommunications, China
Hongtao Zhang	Kochi University of Technology, Japan
Jiahui Zhang	Beijing University of Technology, China
Jinchuan Zhang	University of Electronic Science and Technology of China, China
Kai Zhang	East China Normal University, China
Kang Zhang	Kyushu University, Japan
Li Zhang	Soochow University, China
Qian Zhang	Jiangsu Open University, China
Ruixiao Zhang	University of Southampton, UK
Ting Zhang	Central China Normal University, China
Tong Zhang	China Mobile Research Institute, China
Weili Zhang	Xi'an Jiaotong University, China
Weilun Zhang	Tianjin University, China
Wendy Zhang	University of Canterbury, New Zealand
Xiaowei Zhang	Qingdao University, China
Xu Zhang	Jiangsu Normal University, China

Xulong Zhang	Ping An Technology (Shenzhen) Co., Ltd., China
Xunhui Zhang	National University of Defense Technology, China
Yang Zhang	City University of Hong Kong, China
Yangsong Zhang	Southwest University of Science and Technology, China
Zijing Zhang	University of Waikato, New Zealand
Bo Zhao	Beijing Normal University, China
Hui Zhao	University of Jinan, China
Jianhui Zhao	Wuhan University, China
Jigui Zhao	Xinjiang University, China
Jing Zhao	Qilu University of Technology, China
Keer Zhao	Zhejiang University of Technology, China
Ming Zhao	Central South University, China
Rui Zhao	University of Technology Sydney, Australia
Runjie Zhao	Zhejiang University, China
Xujian Zhao	Southwest University of Science and Technology, China
Yan Zheng	Kyushu University, Japan
Yingtao Zheng	University of Auckland, New Zealand
Yuchen Zheng	Shihezi University, China
Guoqiang Zhong	Ocean University of China, China
Jinghui Zhong	South China University of Technology, China
Ping Zhong	Central South University, China
Guangchong Zhou	Chinese Academy of Science, China
Jinjia Zhou	Hosei University, China
Shihua Zhou	Dalian University, China
Xiao-Hu Zhou	Chinese Academy of Sciences, China
Xinyu Zhou	Jiangxi Normal University, China
Yu Zhou	National University of Defense Technology, China
Zhenxiong Zhou	National University of Defense Technology, China
Leyi Zhu	University of Macau, China
Qiaoming Zhu	Soochow University, China
Qiyuan Zhu	Swinburne University of Technology, Australia
Xuanying Zhu	Australian National University, Australia
Yuesheng Zhu	Peking University, China
Wang Ziling	Sichuan University, China
Yuan Zong	Southeast University, China
Xu Zou	Zhejiang University, China

Contents – Part XI

Human Activity Recognition Model Capable of Handling Various Input Waveforms	1
<i>Tatsuhito Hasegawa</i>	
Enhance Radar Point Cloud with 2D Diffusion	17
<i>Yinbao Li, Yulei Zhang, Rui Zhu, and Chang Liu</i>	
LBRLF: Lightweight Privacy-Preserving Federated Learning with Byzantine-Robustness	32
<i>Pingzhang Shen, Shengnan Zhao, Jun Xu, Chuan Zhao, Zhenxiang Chen, and Shangqing Guo</i>	
Izhikevich Neurons in NeuCube for Longitudinal Data Classification	48
<i>Balkaran Singh, Sugam Budhraja, Maryam Doborjeh, Zohreh Doborjeh, Edmund Lai, and Nikola Kasabov</i>	
Calming the Mind: Spiking Neural Networks Reveal How Havening Touch to Reduce Persistent Distress Attenuates Left Temporal Electroencephalographic Connectivity	61
<i>Alexander Sumich, Zohreh Doborjeh, Nadja Heym, Aroha Scott, Kirsty Hunter, Tony Burgess, Julie French, Mustafa Sarkar, Maryam Doborjeh, and Nicola Kasabov</i>	
SCA-LSTM: A Deep Learning Approach to Golf Swing Analysis and Performance Enhancement	72
<i>Chengwei Feng, Boris Baćić, and Weihua Li</i>	
Unsupervised Document Image Tampering Localization via Anomaly Detection	87
<i>Yuan Li, Yan-Ming Zhang, Fei Yin, and Lin-Lin Huang</i>	
3VNet: Topological-Structure Driven Triple-V Network for Retinal Vessel Segmentation	102
<i>Wei Zhou, Xiaorui Wang, Bin Zhou, and Yugen Yi</i>	
Identification of Proton Exchange Membrane Fuel Cell Parameters Using a Parameterless Swarm Intelligent Algorithm	116
<i>Pankaj Sharma, Rohit Salgotra, Sarvanakumar Raju, Szymon Lukasik, and Amir H. Gandomi</i>	

Spatio-Temporal Graph Neural Networks for Infant Language Acquisition Prediction	133
<i>Andrew Roxburgh, Floriana Grasso, and Terry R. Payne</i>	
AUV Efficient Navigation Relying on Adaptive Proximal Policy Optimization	149
<i>Jingzehua Xu, Yongming Zeng, Jintao Zhang, Xuanchen Li, Lingru Meng, Haocai Huang, Jingjing Wang, and Yong Ren</i>	
MedSiML: A Multilingual Approach for Simplifying Medical Texts	165
<i>Hardik A. Jain, Chirayu Patel, Riyasatali Umatiya, Sajib Mistry, Aneesh Krishna, and Amin Beheshti</i>	
A Study on Time-Resilient Features for Detecting TLS Encrypted Malware Traffic	180
<i>Kaisei Fujiwara, Akira Yamada, Seiichi Ozawa, and Chanho Park</i>	
Enhancing Semantic Segmentation in Open Compound Domain Adaptation Through Mixed Image and Epistemic Uncertainty	196
<i>Yiqun Ma, Wenrui Wang, Siyuan Wang, Xi Yang, and Yuyao Yan</i>	
What Should Insect Brains Forget?	211
<i>Koichiro Yamauchi and Takahiro Hirate</i>	
Agent Clustering and Information Sharing Underlying MADRQN for Traffic Light Cooperative Control	227
<i>Haoran Cheng, Bo Wang, Jie Liu, and Tongchun Du</i>	
A Hybrid Contextual Deep Learning Model to Predict Renewable Energy Generation	243
<i>Deepak Kanneganti, Sajib Mistry, Sumedha Rajakaruna, Aneesh Krishna, and Amin Beheshti</i>	
Knowledge Tracing Method Based on Enhanced Global and Local Knowledge State Representation	258
<i>Jiagui Xiong, Hua Chen, Jiayu Hu, Xinyu Zhou, Wenlong Ni, and Hongwei Li</i>	
Tensor Mutual Information for Similarity Measurement of High-Dimensional Data: An Image Classification Perspective	271
<i>Joarder Kamruzzaman, Shaoning Pang, Liangfu Lu, and Jianwei Liu</i>	

Enforcing Specific Behaviours via Constrained DRL and Scenario-Based Programming	284
<i>Davide Corsi, Raz Yerushalmi, Guy Amir, Alessandro Farinelli, David Harel, and Guy Katz</i>	
Explainable AI in Feature Selection: Improving Classification Performance on Imbalanced Datasets	303
<i>Shahriar Siddique Ayon, Muhammad Ebrahim Hossain, Md Saef Ullah Miah, M. Mostafizur Rahman, and Mufti Mahmud</i>	
Enhancing Industrial Energy Efficiency with Predictive Analytics and Fuzzy Logic: A Case Study of Renewable Energy Management in the Meat Processing Industry	319
<i>Mostafa Pasandideh, Jason Kurz, Martin Atkins, and Mark Apperley</i>	
A Dual-Branch Riemannian Learning Network for EEG Speech Imagery Decoding	335
<i>Liying Zhang, Peiliang Gong, Qianru Sun, Yueying Zhou, Qi Zhu, and Daoqiang Zhang</i>	
ROSAL: Semi-supervised Active Learning with Representation Aggregation and Outlier for Endoscopy Image Classification	350
<i>Xiaocong Huang, Guoheng Huang, Guo Zhong, Xiaochen Yuan, Xuhang Chen, Chi-Man Pun, and Jianwu Chen</i>	
Adaptive Population-Based Incremental Learning for Feature Selection in Leukemia Gene Expression Data	365
<i>Eranga N. Fernando and Jeremiah D. Deng</i>	
Author Index	379



Human Activity Recognition Model Capable of Handling Various Input Waveforms

Tatsuhito Hasegawa^(✉)

Graduate School of Engineering, University of Fukui, Bunkyo 3-9-1,
Fukui 918-8105, Japan
t-hase@u-fukui.ac.jp

Abstract. In the fields of image recognition and natural language processing, foundation models built with extremely large datasets are being applied to various downstream tasks. However, in the field of sensor-based human activity recognition (HAR), constructing extremely large datasets is challenging, and the sensors used for each task differ, leading to significant variations in the input data formats. These unique challenges make it difficult to realize foundation models in HAR. In this study, I developed a model architecture that can uniformly handle diverse data formats, aiming to create a foundation for pre-training across multiple datasets in the future. I proposed an input combination strategy, a parameter sharing strategy, and an output combination strategy for HAR using multiple sensors, achieving a model capable of handling various inputs. Furthermore, I proposed dynamic grouping and evaluated its effectiveness using a public HAR dataset PAMAP2, revealing a model with improved estimation accuracy and appropriate parameter count due to dynamic grouping.

Keywords: Human activity recognition · neural networks · ensemble learning · data fusion

1 Introduction

Sensor-based human activity recognition (HAR) technology plays an important role in modern society and is utilized in various fields such as healthcare, security, and human-computer interaction (HCI) [9, 18, 20]. By detecting people's daily activities in real time, it enables applications such as health management for the elderly, detection of suspicious behavior, and personalized user support based on past behaviors. There are two main approaches to activity recognition: image-based methods and sensor-based methods. Image-based methods [26] estimate the activities of individuals appearing in images captured by cameras, making them effective for analyzing the movements of multiple people in a fixed location. Sensor-based methods [29], on the other hand, estimate the activities of individuals based on sensor data measured by wearable devices, making them

Table 1. Difference between sensor-based and image-based human activity recognition.

		Sensor based	Image based
(1)	Range	Differ for each sensor	Constant $\{x \in \mathbb{Z} \mid 0 \leq x \leq 255\}$
(2)	Characteristics	Nearest neighbor changes and periodicity need to be considered	Object features tend to cluster in the vicinity on the coordinates.
(3)	Measured environments	Characteristics vary from device to device and from individual to individual	Characteristics vary slightly with photographic equipment and environment
(4)	Public dataset	Many medium to small datasets	Many large datasets
(5)	Data shape	Undecided: window size \times # of sensors \times # of channels	Constant: height \times width \times RGB (3ch)

effective for tracking specific individuals and recognizing activities that cannot be identified through images.

There are various approaches to achieving HAR, but a common method involves extracting features designed based on human experience from measurement data and classifying activities using machine learning [2]. In recent years, methods that directly estimate activities from input data using deep learning models such as convolutional neural networks (CNNs) and Transformers have become prevalent [21]. It is important to note, as summarized in Table 1, that the characteristics of the data and models differ significantly between sensor-based and image-based methods. For example, sensor-based methods handle diverse sensors, resulting in undefined ranges for measurement values, and they also need to consider periodicity.

In this study, I focus on sensor-based HAR and address the issues (4) and (5) summarized in Table 1. Issue (4) pertains to the differences in existing datasets. In image-based methods, there are numerous large-scale datasets such as ImageNet [5] and the Open Images Dataset [13]. In contrast, while several datasets are available for sensor-based methods, such as HASC [10], PAMAP2 [19], OPPORTUNITY [3], and UniMiB SHAR [4], they often have a limited number of activity classes and smaller overall data volumes. Issue (5) concerns the differences in data formats. Image-based inputs are generally limited to images, resulting in a fixed format of height \times width \times RGB (3 channels). In contrast, sensor-based inputs vary depending on the type of sensor, its position, and the number of sensors used in each domain, leading to an undefined data format such as window length \times number of sensors \times number of channels.

In our previous research [34], I proposed domain robust pretraining (DRP) to address issue (4) by utilizing multiple datasets (OPPORTUNITY and PAMAP2) for pretraining a single model. However, this method is limited to handling 3-axis accelerometers and does not leverage the numerous sensors available in each domain.

Based on the above, this study aims to develop a robust HAR model that can handle the varying data formats characteristic of sensor-based inputs (issue (5)). The novelty of this study lies in decomposing the sensor data by channels and

then ensembling them. This approach enables the handling of various datasets with different formats using a single model. By combining this method with DRP [34] in the future, I aim to address issue (4) and realize a foundation model for HAR.

2 Related Studies

2.1 Sensor-Based HAR

Bao et al. [2] achieved HAR using machine learning based on measurements from accelerometers worn at multiple positions. The measurements were divided using a sliding window method, and they extracted features based on DC characteristics and energy characteristics from the FFT results. These features were then used to build classification models with machine learning algorithms such as decision trees. Additionally, various methods of measurement, preprocessing techniques, feature extraction methods, and the selection of learning algorithms have been explored in other studies [1, 14, 22, 27].

In recent years, there are some active studies on HAR using deep learning. Li et al. [15] demonstrated that a deep learning model, which stacks a long short-term memory (LSTM) network on top of a three-layer CNN, achieves higher accuracy compared to traditional methods that rely on feature extraction and machine learning as described above. Additionally, our previous research [35] comprehensively evaluated various CNN architectures proposed in the field of image recognition, showing that the Inception architecture is effective for HAR. Furthermore, other studies have explored optimal structures through network architecture search [12].

As such, many methods for achieving HAR have been extensively studied. However, the issue of unstructured formats mentioned in Table 1 (5) has often been overlooked. Methods based on feature extraction and machine learning require the implementation of appropriate feature extraction algorithms for each format. Although deep learning methods allow feature extraction to be data-driven, the input format remains fixed, which means the issue in (5) cannot be resolved.

2.2 Multimodal HAR

In multimodal HAR, where multiple sensor devices are used simultaneously to measure activities, it is essential to carefully design the input format for the model. Many HAR models [12, 15, 35] use a single 3-axis accelerometer as input, resulting in an input format of $\mathbf{x} \in \mathbb{R}^{B \times 3 \times L}$, where B is the batch size and L is the window size. Similarly, in cases of multimodal HAR [6, 28], the input format is treated as $\mathbf{x} \in \mathbb{R}^{B \times n \times L}$, where n is the total number of channels from all modalities. For instance, when dealing with two 3-axis accelerometers, n would be six. Essentially, this method involves simply concatenating the data along the channel dimension. Special cases include methods that concatenate data from all channels along the L dimension, resulting in $\mathbf{x} \in \mathbb{R}^{B \times 1 \times nL}$ [11], or models that

incorporate sensor fusion using complementary filters into the model architecture [31]. However, these methods cannot resolve the issue mentioned in (5) when the number of sensors increases or decreases.

Deep learning models that handle multimodal sensor data have also been proposed. Baloch et al. [1] proposed a method that extracts features using different CNNs for each sensor and then combines the feature maps. Since multiple CNNs are used with the input format $\mathbf{x} \in \mathbb{R}^{B \times 3 \times L}$, this method can accommodate various input formats by increasing or decreasing the number of CNNs. However, it assumes that each input has three channels. Similar approaches are employed in the literature [25, 30, 33]. As unique innovations, Yang et al. [30] introduced dynamic dropout and parameter quantization for the feature maps, while Tao et al. [25] and Zhao et al. [33] implemented dynamic weighting using self-attention on the feature maps. Additionally, Münzner et al. [16] and Gholamrezaei et al. [7] proposed methods that decompose sensor measurements by channel and extract features using different CNNs. Since multiple CNNs are used with the input format $\mathbf{x} \in \mathbb{R}^{B \times 1 \times L}$, this approach can accommodate various input formats, including sensors with more than three channels. However, experiments have shown that not adopting this method can lead to higher accuracy [7], and similar accuracy can be achieved by deepening the layers [16], indicating that the effective conditions are not yet clear.

2.3 Position of This Study

Based on the above, the novelty of our proposed method is the following two points.

- To organize existing multimodal HAR and propose new input combining strategies, parameter sharing strategies, and output combining strategies that comprehensively represent (Sect. 3.1).
- To propose new policies for grouping input combining and parameter sharing, such as per-axis, per-sensor type, and new dynamic grouping methods (Sect. 3.2).
- Train models end-to-end, rather than designing based on prior human knowledge.

Furthermore, through exhaustive validation experiments, this study contributes to the following findings.

- I demonstrated that the model may not be too wide.
- Finding a model that accounts for the trade-off between estimation accuracy and the number of parameters by input combining and parameter sharing grouping.

3 Proposed Method

3.1 Comprehensive Representation of Multimodal HAR

Input Combining Strategies. Based on related studies, multiple sensor input combining strategies in HAR can be classified into three methods: Fig. 1.

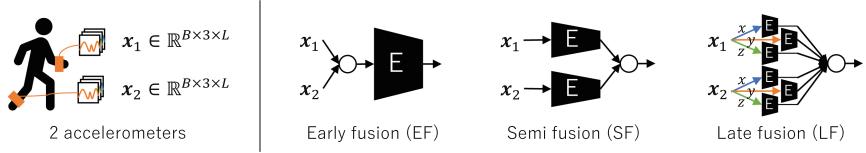


Fig. 1. Fusion strategies in multi-modal input. E stands for the encoder and \circ stands for concatenate processing in the channel direction.

- **Early fusion (EF):** After combining all raw measurements in the channel direction, feature maps are calculated by inputting them into a single Encoder (E).
- **Semi fusion (SF):** After combining raw measurements in the channel direction for each group, feature maps are calculated by inputting them into an E.
- **Late fusion (LF):** Feature maps are calculated by inputting each channel's measurement into an E, and merged.

EF processes all sensor data combined along the channel direction, which allows for the acquisition of cross-channel feature representations. However, because it needs to handle diverse modality information, it requires complex knowledge for E. This implies that a large amount of data might be necessary for sufficient training. On the other hand, LF independently computes feature maps for each channel before combining them. Although it cannot obtain cross-channel feature representations, each E can learn representations specialized for each channel. This method potentially allows for model training with limited data since it can restrict the domain of the input data. SF is an intermediate method between EF and LF. By appropriately defining groups, it can achieve the benefits of both EF and LF. Notably, SF is referred to as sensor-based LF in the literature [16].

Parameter Sharing Strategies. When using multiple Es as adopted in SF or LF, one can choose whether to share parameters between the Es. For example, in the literature [30, 33], parameters are shared, whereas in [7, 16], although it is not explicitly stated, it is assumed that independent Es without shared parameters are used. Considering grouping as in SF for parameter sharing, the following three methods can be classified as shown in Fig. 2.

- **Shared (Sh) :** Parameter sharing for each E.
- **Semi-shared (SSh) :** Parameter sharing for each group of E.
- **Independent (Id) :** Each parameter of E is independent.

According to the author's survey, SSh has not been proposed in the past. For example, strategies such as sharing parameters for each sensor or sharing them based on the sensor's attachment position can be adopted.

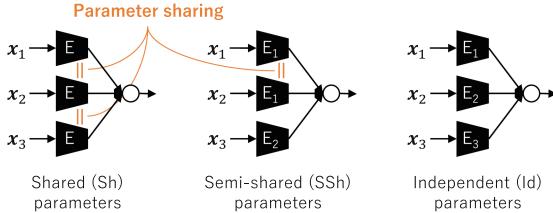


Fig. 2. Parameter-sharing strategies among encoders.

Since $\text{Sh} < \text{SSh} < \text{Id}$ in terms of the number of independent parameters for each E , the total number of model parameters increases in that order. While Sh can suppress the number of parameters, it complicates the domain handled within a single E , such as processing both accelerometer and geomagnetic data with the same E . This complexity may prevent the model from acquiring sufficient knowledge during training, potentially reducing estimation accuracy. On the other hand, inputting both accelerometer and geomagnetic data into a single E can be seen as an effective increase in training data. Therefore, if a good trade-off can be found, SSh can offer the advantages of both Sh and Id .

To share parameters, it is necessary to standardize the format of the input data. Therefore, during implementation, a conv1D layer and a batch normalization layer are inserted as a “Stem” layer immediately before each E . This process converts the data to three channels regardless of the original number of channels. This allows the parameters of each E to be shared, even if various channels are grouped by the input fusion strategy.

Output Combining Strategies. When using multiple E s, such as in SF or LF, one can choose the strategy for combining the feature maps output by each E . Traditional research has adopted concatenation (hereafter, cat), but since the format of the feature maps can be unified across each E , it is also possible to combine them using summation (hereafter, sum). In the study of multimodal data fusion [32], other weighted sum and outer product methods have also been proposed.

If the feature map output by each E is represented as $\mathbf{z}_i \in \mathbb{R}^{B \times d}$, then for cat, the overall feature vector becomes $\mathbf{z} \in \mathbb{R}^{B \times nd}$. Here, d is the dimension of the feature map output by each E . On the other hand, for sum, $\mathbf{z} \in \mathbb{R}^{B \times d}$, thus reducing the number of parameters in the classifier. While cat allows for independent weighting for each channel within the classifier, sum results in each channel being combined with equal weights, highlighting a key difference between the two approaches.

3.2 Grouping Strategies

Straight-Forward Grouping. SF in the input combining strategy and SSh in the parameter sharing strategy require the grouping of all input channels. For

instance, the method by Baloch et al. [1], which uses Es for each sensor, can be considered as an SF that groups by sensor. Based on prior knowledge, I define the following three straight-forward grouping methods:

- **Sensor-by-sensor (SbS):** Grouping by measurement sensor unit, such as 3-axis acceleration, 3-axis gyro, etc.
- **Axis-by-axis (AbA):** Grouping by axis, such as x-axis and y-axis for each sensor.
- **Type-by-Type (TbT):** Grouping by sensor type, such as acceleration and gyro, across multiple IMUs.

Dynamic Grouping. As SSh moves closer to Sh by advancing groupings, the training data given to each E can be effectively increased, but the domain handled by each E becomes more complex. Assuming that the complexity of the domain adversely affects training, it is desirable to group channels with similar domains. Therefore, this study proposes a new method for dynamically grouping channels based on the input distribution of each channel through clustering.

To group channels, it is necessary to define a distance measure between channels and perform clustering. While there are time series distance measures like dynamic time warping (DTW) that consider time distortions, calculating combinations of large amounts of data, such as training data in deep learning, requires significant computational cost. Therefore, this study defines the distance measure between channels based on the Jensen-Shannon Divergence (D_{JS}) calculated from the probability density estimation of data for each channel. D_{JS} is a symmetric version of the Kullback-Leibler Divergence (D_{KL}) and is defined by the following equation.

$$D_{JS}(P||Q) = \frac{1}{2}\{D_{KL}(P||M) + D_{KL}(Q||M)\}. \quad (1)$$

where P, Q are the probability distributions for which the distance is to be calculated, respectively, and the M denotes $M = (P + Q)/2$. D_{KL} is defined by the following equation.

$$D_{KL}(P||Q) = - \sum_{x \in X} P(x) \log \frac{Q(x)}{P(x)}. \quad (2)$$

Thus, D_{KL} means the expected value of the difference between $P(x)$ and $Q(x)$ in all stochastic variables $x \in X$.

Based on the above, the proposed method calculates the distance between channels P, Q by the following procedure.

$$D(P||Q) = \sum_{c \in C} D_{JS}(P_c||Q_c). \quad (3)$$

Here, P_c and Q_c represent the probability distributions of channels p and q for each activity class c , respectively. In other words, the sum of D_{JS} over all

classes is used as the distance measure. However, D_{JS} is clipped to a maximum value of 10. The probability distributions P_c and Q_c are estimated from the measured sensor data using kernel density estimation with a Gaussian kernel.

Finally, the distance matrix between channels is calculated, and the channels are grouped using agglomerative hierarchical clustering with the group average method. The number of clusters k is a hyper parameter; in the parameter-sharing strategy (SSh), the closer k is to the total number of channels, the closer it is to Id, and the closer k is to 1, the closer it is to Sh.

4 Experimental Settings

4.1 Dataset

I validate the effectiveness of the proposed method using the PAMAP2 benchmark dataset for multimodal HAR [19]. The PAMAP2 dataset includes nine subjects, though there is variability in the recorded activities. While there are 19 different activities including “others,” only the 12 activities with the highest number of recordings are used: lying, sitting, standing, walking, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning, ironing, and rope jumping.

PAMAP2 utilizes three IMUs and one heart rate monitor (HRM) for HAR. The IMUs are attached to the dominant wrist, chest, and dominant ankle, with a sampling rate 100 Hz. The HRM is belt-mounted on the chest and has a maximum sampling rate 9 Hz. The IMUs measure temperature (1 axis), accelerometer A (3 axes), accelerometer B (3 axes), gyroscope (3 axes), magnetometer (3 axes), and orientation (3 axes). Accelerometers A and B differ in their measurement scales. Since the official documentation indicates a failure in recording orientation data, the 3-axis orientation measurements are not used in this study. The HRM measures heart rate (1 axis). Including the 3 axes for temperature and the 1 axes for heart rate, the total number of axes is 40. Therefore, the input data format is $\mathbf{x} \in \mathbb{R}^{B \times 40 \times 256}$, where B is the batch size.

Since the measured data consists of variable-length time series, the data is segmented using a sliding window approach, which is commonly used in HAR. To exclude data before and after the start of the activity, two seconds were removed from both the beginning and the end of each recording file. The time series data was then segmented with a window size of 256 samples and a stride of 128 samples. Given the sampling frequency 100 Hz, each data segment corresponds to approximately 2.56 s. The HRM, which has a different sampling rate, was resampled 100 Hz using linear interpolation. Although each sensor has different scales, preliminary experiments indicated that standardization negatively affected estimation accuracy; therefore I did not apply the standardization for raw measurements. Ultimately, 14,689 data segments were extracted from the data of the nine subjects.

4.2 Evaluation Method

Since the characteristics of activities can vary significantly between subjects, HAR is generally evaluated by splitting the training and testing data on a per-subject basis. In this study, I use leave-one-subject-out cross-validation (LOSO-CV) to evaluate the performance on the nine subjects. Ideally, the number of data points per subject should be considered, but in this case, I calculate scores for each subject and discuss the results based on the average of the cross-validation scores for all nine subjects. Additionally, to mitigate the impact of randomness, I perform 10 trials with different random seeds and use the average accuracy from these trials for discussion.

I used accuracy as the evaluation metric, defined as the ratio of correctly predicted instances to the total instances for the 12 target activities. The accuracy of deep learning models varies with the training progress (epochs), and this variability is particularly pronounced in HAR due to the limited amount of data. Therefore, during evaluation, I use the median accuracy over the final 10 epochs.

In this study, I do not split the data into training, validation, and testing sets; instead, I evaluate using only the training and testing datasets with LOSO-CV. It is important to note that the tuning results discussed in this study are based on test accuracy. In the future, robustness verification through evaluation with a separate test dataset will be necessary.

4.3 Model Architecture and Training Settings

In the proposed method, the model architecture of E is flexible; however, in this study, I adopt a 10-layer VGG [23] as a simple architecture, as shown in Fig. 3. I adopted VGG because, despite its simplicity, its effectiveness had been demonstrated in previous study [35]. To avoid the issue described in (5), a Stem layer is applied to standardize the input channels, followed by E, an 8-layer CNN, and finally a classifier, which consists of a single fully connected layer for activity classification. By adjusting the number of filters f , which is a hyperparameter, I can control the width (number of parameters) of the E model. Note that the parameters shared in the parameter-sharing strategy are only those of E.

Based on the preliminary experimental results, the training of the model was adjusted to use 150 epochs, a batch size of 1000, the Adam optimizer, an initial learning rate of 1e-3, and a learning rate scheduler with cosine annealing ($T_{max} = 50$).

5 Experimental Results

5.1 Consideration of Model Size and Output Combining Strategies

In image recognition, VGG typically uses $f = 64$ filters, but in HAR, such a large model may not converge during training. Therefore, I conducted an investigation into the appropriate model size using sensor-specific SF (SbS) as the input combining strategy. I conducted experiments with all combinations of

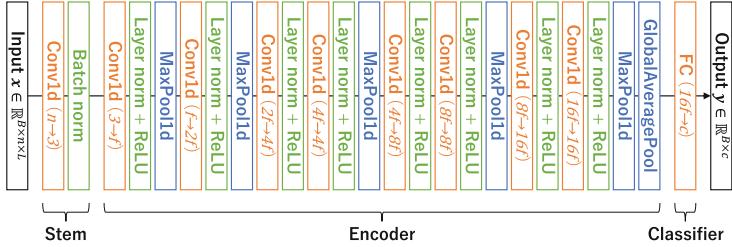


Fig. 3. Model architecture of VGG10 used in our experiments.

parameter-sharing strategies (Sh or Id) and output combining strategies (cat or sum). Figure 4 the experimental results. The vertical axis represents test accuracy, the horizontal axis represents the total number of model parameters, and each point corresponds to models with widths $f = 1, 2, 4, 8, 16, 32$, respectively. Note that the Stem was excluded in this experiment.

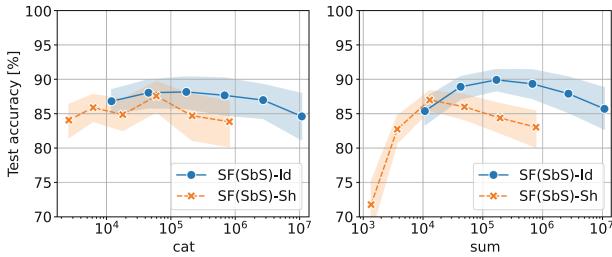


Fig. 4. Parameter-sharing strategies among encoders. The horizontal axis is the number of parameters in the entire model. The points are $f = 1, 2, 4, 8, 16, 32$ from left to right.

Several insights were obtained from Fig. 4. First, in all cases, $f = 4$ or 8 filters achieved the best accuracy. It was initially expected that Sh, which shares parameters and thus effectively increases the training data for E, would allow for training larger models compared to Id. However, contrary to expectations, relatively small models were found to be appropriate in all cases. Second, Id achieved higher accuracy than Sh. Although the difference in parameter-sharing strategies has not been mentioned in related research, I found that preferable in PAMAP2 to train Es individually rather than sharing them across sensors. Third, while cat was robust to changes in f , sum achieved slightly higher accuracy with an appropriate f . Therefore, if the goal is to reduce the number of parameters, cat should be used; otherwise, sum is preferable.

5.2 Input Combining Strategies and Parameter Sharing Strategies

I evaluated the effectiveness of the input combining strategy and parameter-sharing strategy, comprehensively defined in this study for multimodal HAR. Based on the results from the previous section, the common conditions are set with the width of each E fixed at $f = 8$, and the output combining strategy is set to cat.

Figure 5 shows the total number of model parameters for each combining of strategies. The vertical axis represents the input combining strategy, while the horizontal axis represents the parameter-sharing strategy. It is evident that the number of parameters can be reduced as grouping progresses in each strategy. In the case of Sh, the number of parameters for E remains the same; therefore, the slight differences in parameter numbers are due to the Stem's and classifier's parameters depending on the input combining strategy. On the other hand, in the case of Id, where multiple Es are used in parallel, the total number of parameters significantly differs depending on the input combining strategy, which determines the number of Es.

Figure 6 shows the test accuracy for each combining of strategies. It is evident that accuracy decreases as grouping progresses in any strategy. Looking at the differences in input combining strategies, EF, SF(SbS), and LF achieved 73.4%, 87.3%, and 90.9%, respectively. Although this result contradicts the conclusions of related research [7], it aligns with the suggestion in [16] that deeper layers could reverse this trend. By adopting VGG10, it appears that the inherent performance has been realized, demonstrating the effectiveness of LF. Moreover, grouping strategies based on prior knowledge (TbT, AbA) did not perform well, indicating that the traditional method of SbS is appropriate for the input combining strategy.

	Sh	SSh (TbT)	SSh (SbS)	SSh (AbA)	Id
EF		49k			
SF(TbT)	53k	-	-	-	293k
SF(SbS)	59k	300k	-	-	684k
SF(AbA)	59k	300k	-	-	684k
LF	80k	320k	704k	704k	1,954k

Fig. 5. Number of parameters in the overall model when each strategy is employed.

	Sh	SSh (TbT)	SSh (SbS)	SSh (AbA)	Id
EF		73.4			
SF(TbT)	78.2	-	-	-	81.3
SF(SbS)	87.3	86.4	-	-	88.3
SF(AbA)	83.8	85.8	-	-	86.1
LF	90.9	90.3	90.0	91.4	91.1

Fig. 6. Average of 10 trials of test accuracy [%] for each strategy employed.

Focusing on the parameter-sharing strategy, SF showed a notable difference in accuracy between Sh and Id, possibly due to the limited number of Es (5 for TbT, and 14 for both SbS and AbA). On the other hand, LF, with a total of 40 Es, showed a relatively small difference in accuracy between Sh and Id, likely due to the effective increase in training data resulting from the Sh and SSh strategies.

The method proposed in this study is applicable to various backbones, and Table 2 shows the results of experiments with different backbones. For comparison, we used ResNet18 [8] and Inception-v3 [24], which are commonly used in image recognition, as well as DeepConvLSTM [17], which is frequently adopted as a baseline for HAR. Based on the tuning results for VGG, the number of filters in the first layer of ResNet18 and Inception-v3 was set to 8. DeepConvLSTM remained in its original form. The table shows that, regardless of the backbone, the accuracy of SF(SbS) is higher than that of EF, although the number of parameters increases. It should be noted that VGG10 achieved the best performance as a result of parameter tuning.

Table 2. Impact of the backbone architecture (except for VGG10, the results are the average of three trials.)

	# of params		Test accuracy [%]	
	EF	SF(SbS)-Id	EF	SF(SbS)-Id
ResNet18 [8]	64k	867k	73.3	83.3
Inception-v3 [24]	910k	12,728k	73.6	83.9
DeepConvLSTM [17]	307k	4,137k	65.9	74.3
VGG10	49k	684k	73.4	88.3

5.3 Dynamic Grouping

I evaluated the impact of the proposed dynamic grouping method. Figure 7 shows the results of its application to the input combining strategy. In this evaluation, parameter sharing between Es is not performed (Id). The values in parentheses, C02 to C25, indicate the number of clusters. From the table, it is evident that while grouping progresses from LF to EF, resulting in reduced parameters, there is a significant decrease in accuracy.

Similarly, Fig. 8 shows the results of applying the proposed dynamic grouping method to the parameter-sharing strategy. In all cases, the input combining strategy is LF. From the table, the trend $\text{SSh}(\text{C02}) < \text{Sh} \leq \text{SSh}(\text{C03})$ is observed. Upon examining the clustering results, it was found that in C02, the data was split into temperature (3 axes) and everything else (37 axes). In C03, each IMU's gyroscope (9 axes) and some of the x-axes formed new clusters. The appropriate partitioning based on data distribution reveals that the proposed method can achieve equivalent or better accuracy while reducing the number of parameters compared to LF(SbS) and LF(AbA).

5.4 The Impact of Varying IMU Conditions Across Subjects

The impact of the initially anticipated issue of differing data formats across datasets was investigated by simulating this scenario. Table 3 shows the results of the evaluation, where the IMU (12 axes) from randomly determined positions for

	Test acc. [%]		# of params
	mean	std	
EF	73.4	19.7	49k
SF(C02)-Id	65.5	20.5	98k
SF(C03)-Id	74.7	18.6	147k
SF(C05)-Id	82.5	11.2	245k
SF(C10)-Id	87.5	8.2	489k
SF(C25)-Id	89.2	8.9	1,222k
LF-Id	91.1	6.6	1,954k

Fig. 7. Impact of dynamic grouping on input fusion strategies.

	Test acc. [%]		# of params
	mean	std	
LF-Sh	90.9	5.3	80k
LF-SSh(C02)	89.6	9.3	128k
LF-SSh(C03)	91.4	5.7	176k
LF-SSh(C05)	91.3	5.2	272k
LF-SSh(C10)	91.1	5.8	512k
LF-SSh(C25)	91.4	6.1	1,233k
LF-Id	91.1	6.6	1,954k

Fig. 8. Impact of dynamic grouping on parameter sharing strategies.

each subject were omitted. Due to implementation constraints, the experiments were conducted with zero-padding instead of actual data omission.

From the table, it can be seen that the overall trends are similar to those in normal conditions. LF-Id achieved the highest accuracy at 71.7%, while EF performed the worst at 54.7%. Notably, the gap between Sh and Id has widened. This indicates that in environments where sensors or channels may be missing, as in this experiment, the importance of grouping strategies increases.

Table 3. Test accuracy [%] of each method in situations where IMU varies from subject to subject.

	EF	SF(SbS)-Sh	SF(SbS)-Id	LF-Sh	LF-Id
Accuracy	54.7	65.8	70.5	68.2	71.7

6 Conclusion

This study addresses the challenge of variable input formats in sensor-based HAR tasks by developing an activity recognition model capable of handling various formats. After reviewing related studies on multimodal HAR using deep learning, I proposed input combining strategies, parameter-sharing strategies, and output combining strategies to comprehensively represent these methods. Furthermore, I introduced a method for dynamic groups within each strategy.

Validation experiments using the PAMAP2 dataset demonstrated the effectiveness of the proposed method, yielding several insights. First, regarding input combining strategies, it was found that it is preferable to perform feature extraction using independent CNNs for each channel (sensor axis) without input fusion. When parameters are not shared, performance can be improved with only a slight increase in the number of parameters. Second, for parameter-sharing strategies, although not sharing parameters results in higher accuracy, the decrease in accuracy is minimal even when parameters are shared. Third, dynamic grouping for

parameter sharing identified clusters that consider the trade-off between estimation accuracy and the number of parameters. Additionally, in experiments where one IMU was randomly omitted for each subject–LF-Id achieved the highest accuracy. However, there was a 20% drop in accuracy compared to the scenario without omissions, indicating that improving training methods to mitigate accuracy loss is a future challenge. While dynamic grouping formed clusters of temperature and other data, clustering temperature and heart rate might be more appropriate from the perspective of waveform changes. Therefore, there is also room for improvement in the dynamic grouping method. Furthermore, since the robustness across various HAR datasets has not yet been fully demonstrated, we plan to investigate in future work whether similar trends can be observed in different multimodal HAR tasks.

Acknowledgments. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant-in-Aid for Scientific Research (C) under Grant 23K11164. This work was also supported by several competitive funds within the University of Fukui.

References

1. Baloch, Z., Shaikh, F.K., Unar, M.A.: CNN-LSTM-Based late sensor fusion for human activity recognition in big data networks. *Wireless Communications and Mobile Computing* **2022** (Aug 2022)
2. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. *Pervasive Comput.*, 1–17 (2004)
3. Chavarriaga, R., et al.: The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. *Pattern Recogn. Lett.* **34**(15), 2033–2042 (2013)
4. D. Micucci, M.M., Napoletano, P.: Unimib shar: a dataset for human activity recognition using acceleration data from smartphones. *Appli. Sci.* **7**(10) (2017). <https://doi.org/10.3390/app7101101>
5. Deng, J., , et al.: Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
6. Dua, N., Singh, S.N., Semwal, V.B.: Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing* **103**(7), 1461–1478 (2021)
7. Gholamrezaei, M., AlModarresi, S.: A time-efficient convolutional neural network model in human activity recognition. *Multimedia Tools Appl.* **80**(13), 19361–19376 (2021)
8. 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 (CVPR), pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
9. Iqbal, A., et al.: Wearable internet-of-things platform for human activity recognition and health care. *Inter. J. Distributed Sensor Netw.* **16**(6) (2020)
10. Kawaguchi, N., et al.: Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings. In: Proceedings of the Augmented Human International Conference (AH) (Mar 2011)

11. Kaya, Y., Topuz, E.K.: Human activity recognition from multiple sensors data using deep CNNs. *Multimedia Tools Appl.* **83**(4), 10815–10838 (2024)
12. Kobayashi, S., et al.: MarNASNets: toward CNN model architectures specific to sensor-based human activity recognition. *IEEE Sens. J.* **23**, 18708–18717 (2023)
13. Krasin, I., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from (2016). <https://github.com/openimages>
14. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsl* **12**(2), 74–82 (2011). <https://doi.org/10.1145/1964897.1964918>
15. Li, F., et al.: Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors* **18**(679), 1–22 (2018). <https://doi.org/10.3390/s18020679>
16. Münzner, S., et al.: CNN-based sensor fusion techniques for multimodal human activity recognition. In: Proc. of the ACM International Symposium on Wearable Computers (ISWC), pp. 158–165 (Sep 2017)
17. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **16**(1) (2016) <https://doi.org/10.3390/s16010115>
18. Qingxin, X., et al.: Unsupervised factory activity recognition with wearable sensors using process instruction information. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), vol. 3(2), pp. 1–23 (2019)
19. Reiss, A., Stricker, D.: Creating and benchmarking a new dataset for physical activity monitoring. In: In Proc. of the International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), pp. 40:1–40:8 (2012). <https://doi.org/10.1145/2413097.2413148>
20. Sayem, F.R., et al.: Feature-based method for nurse care complex activity recognition from accelerometer sensor. In: Adjunct Proceedings of the UbiComp 2021 and ISWC 2021, pp. 446–451 (2021)
21. Shiranthika, C., et al.: Human activity recognition using cnn & lstm. In: Proceedings of the International Conference on Information Technology Research (ICITR), pp. 1–6 (2020)
22. Shoaiib, M., et al.: Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors* **16**(4) (2016). <https://doi.org/10.3390/s16040426>
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (ICLR), pp. 1–14 (May 2015)
24. 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 (CVPR), pp. 2818–2826 (2016). <https://doi.org/10.1109/CVPR.2016.308>
25. Tao, W., et al.: Attention-Based sensor fusion for human activity recognition using IMU signals (Dec 2021)
26. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **29**(10), 983–1009 (2013). <https://doi.org/10.1007/s00371-012-0752-6>
27. Voicu, R.A., et al.: Human physical activity recognition using smartphone sensors. *Sensors* **19**(3) (2019). <https://doi.org/10.3390/s19030458>

28. Wan, S., et al.: Deep learning models for real-time human activity recognition with smartphones. *Mobile Netw. Appli.* **25**(2), 743–755 (2020)
29. Wang, J., et al.: Deep learning for sensor-based activity recognition: a survey. *Pattern Recogn. Lett.* **119**, 3–11 (2019). <https://doi.org/10.1016/j.patrec.2018.02.010>
30. Yang, Z., et al.: DFTerNet: towards 2-bit dynamic fusion networks for accurate human activity recognition. *IEEE Access* **6**, 56750–56764 (2018)
31. Zhang, Y., et al.: IF-ConvTransformer: a framework for human activity recognition using IMU fusion and ConvTransformer. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), vol. 6(2), pp. 1–26 (2022)
32. Zhao, F., Zhang, C., Geng, B.: Deep multimodal data fusion. *ACM Compu. Surv.* **56**(9) (2024). <https://doi.org/10.1145/3649447>
33. Zhao, Y., et al.: Attention-based sensor fusion for emotion recognition from human motion by combining convolutional neural network and weighted kernel support vector machine and using inertial measurement unit signals. *IET Signal Process.* **17**(4) (2023). <https://doi.org/10.1049/spl2.12201>
34. Zhao, Z.K., Hasegawa, T.: Domain-robust pre-training method for the sensor-based human activity recognition. In: Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), pp. 67–71 (2022). <https://doi.org/10.1109/ICMLC56445.2022.9941291>
35. Zhongkai, Z., et al.: A comparative study: toward an effective convolutional neural network architecture for sensor-based human activity recognition. *IEEE Access* **10**, 20547–20558 (2022). <https://doi.org/10.1109/ACCESS.2022.3152530>



Enhance Radar Point Cloud with 2D Diffusion

Yinbao Li^{1(✉)}, Yulei Zhang², Rui Zhu³, and Chang Liu⁴

¹ Baidu Technology (Beijing), Beijing 100080, Haidian, China
zhousidadi@126.com

² Alibaba Technology (Beijing), Beijing 100015, Chaoyang, China
zhangyulei.zyl@alibaba-inc.com

³ Huawei Technologies, Shenzhen 518129, Longgang, China

⁴ Hong Kong Lingnan University, New Territories 999077, Tuen Mun, China

Abstract. We propose a novel local diffusion aided single stage detector for radar to tackle the noise and sparsity issues of its point cloud in detection tasks. We utilized space occupancy maps to represent the downsampled point cloud and applied a diffusion module to denoise them. We also introduced instance-aware downsampling strategies and 3D upsampling module to enhance the model’s perception ability in 3D space. Experiments show that LDRadSSD outperformed those SOTA approaches in predicting bounding boxes for road users such as cyclists and pedestrians and figuring out the drivable space in bird’s eye view (BEV) of the scene. In particular, with IoU thresholds of 0.5/0.25/0.25, the average prediction precision (AP) of main type of road users (cars, pedestrians and cyclists) reached at competitive 58.5%, 63.1%, and 82.1%, respectively, while mean IoU of free space was 92.6%. Moreover, the prediction precision of object orientation dramatically raised to averaged 71.1%. We also demonstrate in this paper that LDRadSSD can satisfy real-time requirements in autonomous driving as its running speed reach at 18.2 to 41.7 milliseconds per frame.

Keywords: Radar point cloud · Object detection · Diffusion model

1 Introduction

As one of the most important sensors in autonomous driving, 3+1D millimeter-wave (mmw) radar has salient strengths when compared with LiDAR and camera. Specifically, the ability for space penetration of mmw radar facilitates wider dynamic landscapes in free space detection [36], and key information such as Doppler and height measurements carried by its point cloud enables a better understanding of its surroundings. However, radar point clouds have two fatal flaws, namely high noise and sparsity. The existence of these defects significantly affects the calculation accuracy of target detection, making it a far-fetched dream for radar to replace LiDAR in practice. Fortunately, the emergence of generative models has brought hope for addressing these problems simultaneously.

For example, [20] effectively removes noise points and generates point clouds with density comparable to LiDAR by utilizing Unet on radar Range-Angle-Map (RAM).

As for object detection tasks based on point cloud data, current deep learning approaches can be classified into three categories according to how they encode features of the data: voxel-based, point-based, and point-voxel-based. Generally speaking, voxel-based methods have intrinsic deficiency such as quantization loss [35] and the point-voxel-based methods are hard to satisfy real-time requirement in engineering owing to their large calculations. In this paper, we develop our model with point-based methods as they work with only the raw point cloud data and the number of points in radar data is so small that it is promising to achieve real-time performance. Afterwards, we introduce our LDRadSSD model involving efficient data encoding strategies and point cloud enhancement module. Finally, we test our LDRadSSD on public radar dataset View-Of-Delft (VOD, [18]) and nuScenes, and an example scene from VOD is shown in Fig. 1. In summary, our contributions are listed as follows:

- We have adapted the instance-aware sampling strategy designed for LiDAR point clouds to radar ones by introducing upsampling steps.
- We designed a point cloud enhancement module based on stable diffusion techniques, which denoises the point cloud and improves its density effectively.
- We created 2D images in bird’s eye view (BEV) based on occupancy evidence, which assists in detecting 3D objects and drivable area.

2 Related Work

2.1 Point-Based Object Detection

Point-based neural networks [21, 25, 34] directly extract and aggregate point-wise features via PointNet [22] and its variants [11, 23, 24, 29, 30]. Two-stage 3D object detection methods such as PointRCNN [25] first identify foreground points and then encode point-wise features to regress 3D bounding boxes with rich semantics. One-stage 3D detectors such as VoteNet [21] and 3DSSD [34] are based on point selection. Specifically, VoteNet applies a voting mechanism (i.e., Hough Voting) to predict instance centroids, and 3DSSD adopts a sampling strategy considering the Farthest Point Sampling (FPS) in both feature and Euclidean space. IA-SSD [35] is a mixture of the mentioned two types of network proposing an instance-aware downsampling strategy to select a fixed number of representative points. Without complex processing such as voxelization of data, point-based encoding methods, especially the one-stage ones, are straightforward but somewhat limited in efficiency.

On the other hand, convolutional backbones and transformer are the two most popular choices for the core structure of detection networks. In point-based networks, convolutional backbones are usually implemented with a series of convolutional layers to extract multi-scale features from the data and a couple

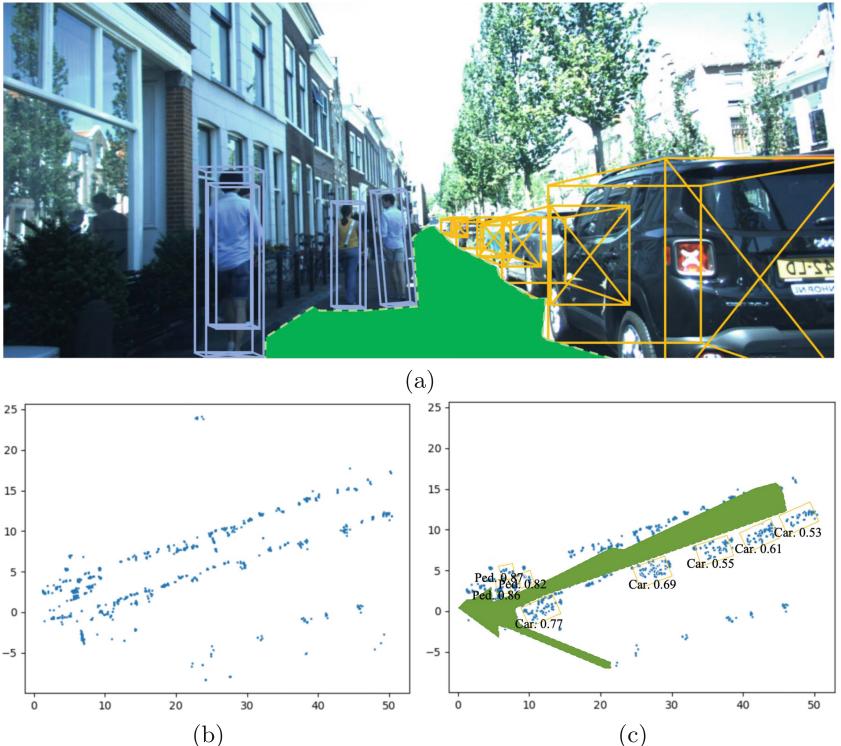


Fig. 1. Example scenario within VOD dataset. (a) shows the camera data of the example scene with labeled targets. (b) is the raw radar point cloud in BEV. (c) depicts the detection results of our model for objects and free space within the example scene (in BEV), where raw point cloud was denoised and locally densified.

of deconvolutional layers serves in model heads to detect objects. In this case, Fully Convolutional Networks (FCN) and Unet [19] are ideal representatives of this encoder-decoder structure. Specifically, Feature Pyramid Network (FPN) [10] was designed to integrate features from all layers of such backbones and non-max suppression (NMS) allows better detection performance by combining multi-scale detection. In contrast, self-attention mechanism in transformer backbones [6] enables the network to learn object-level features, and the following feed-forward network (FFN) are used to produce predictions. To enhance detection performance, voxelization encoding approach is used prior to transformer architecture [6, 12]. Making full use of semantic information is another essential issue in improving learning performance. For example, [36] realized multi-task learning by leveraging the semantic information in their model’s convolutional backbone. An inspiring attention-augmented network [27] was proposed for scene parsing via bilinear upsampling in feature map, bridging the semantic and resolution gap between multi-level features.

2.2 Diffusion Models

Diffusion model is a type of generative model, and the publication of the Denoising Diffusion Probabilistic Model (DDPM) in 2020 has led to a shift of focus in many works in the field of image generation towards diffusion models [5]. Fundamentally, the working principle of diffusion model involves perturbing the training data by continuously adding Gaussian noise and then using a learned denoising process to recover the data [33]. After training, the diffusion model can be used to input randomly sampled noise into the model and generate data through the learned denoising process. Specifically, the diffusion model is a type of latent variable model that uses a Markov Chain to map to a latent space. Although the diffusion model has shown excellent performance in various tasks, it still has its own limitations, such as slow sampling speed and difficulty in maximizing likelihood [7, 17, 28, 31]. Non-Markovian processes or partial sampling were adopted to accelerate diffusion models and objectives designing and noise schedule optimization were used to make it easier for maximizing likelihood. There is also research applying diffusion model in high quality point cloud generation. For instance, [13, 15] proposed certain shape latent to affect reverse diffusion process and thus generate point clouds with smooth surfaces and sharp details. However, it is not necessary to enhance radar point cloud to perfection as there is a risk of being unable to meet the real-time requirements in autonomous driving.

3 The Proposed LDRadSSD

As shown in Fig. 2, our LDRadSSD contains a **main pipeline** and a **branch**. The **branch of LDRadSSD** takes the occupancy evidence map designed in main pipeline as input and ultimately outputs a reference map to guide the generation of high-resolution 3D point clouds for object detection. In the **detection module of LDRadSSD**, we apply 3D non-maximum-suppression (NMS) to determine bounding boxes of main road users in the point cloud and use its 2D image in BEV to implement free space detection.

3.1 Main Pipeline of LDRadSSD

Class-Aware Sampling. The sampling process brings the hazard of losing part of foreground points so a variety of sampling strategies have been proposed. It has been demonstrated that under commonly-used encoding architecture Point-Net++ with 4 encoding layers, the instance recall rate (i.e., the ratio of instance retained after sampling) of farthest point sampling (FPS)-based on Euclidean distance (D-FPS) [23], or feature distance (Feat-FPS) [34], or both (FS) [34] is higher than that of random sampling [35]. Density-based sampling method such as Poisson Disk [4] is effective in recalling instances but time-consuming with small-size data. To strike a balance between the recall rate and time consumption, we adopt D-FPS as the first encoding layer, followed by class-aware

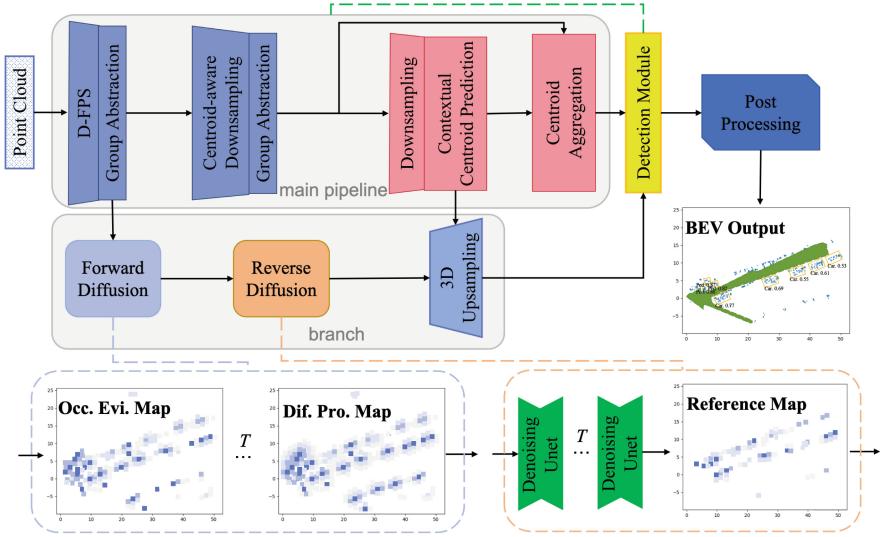


Fig. 2. Overview of the structure of LDRadSSD.

sampling layer, as many object classification models [35] do. The vanilla cross-entropy loss is as follows,

$$L_{cls} = - \sum_{c=1}^C (s_i \log(\hat{s}_i) + (1 - s_i) \log(1 - \hat{s}_i)), \quad (1)$$

where C represents number of object categories, s_i is the one-hot labels and \hat{s}_i is the predicted logits. In LDRadSSD, we use a sampling strategy to learn point-wise semantics and MLP layers are always adopted following it to further estimate the semantic categories of each point.

Centroid-Aware Sampling. For instance-aware tasks, determining an object's location is as significant as identifying its category. To this end, a soft point mask of instances as follows is usually calculated to assign higher weight to points that are nearer to the instance center [34, 35],

$$Mask = \sqrt[3]{\frac{\min(f^*, b^*)}{\max(f^*, b^*)} \times \frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(u^*, d^*)}{\max(u^*, d^*)}}, \quad (2)$$

where $f^*, b^*, l^*, r^*, u^*, d^*$ represent the distance of a point to the 6 surfaces of the 3D bounding box, respectively. It is easy to notice that the value of the mask falls in $[0, 1]$ as once a point locates on those surfaces the numerator of the radicand will be 0. In this way, a weighted centroid-aware sampling strategy can be written as follows.

$$L_{ctr} = - \sum_{c=1}^C (Mask_i \cdot s_i \log(\hat{s}_i) + (1 - s_i) \log(1 - \hat{s}_i)). \quad (3)$$

Contextual Centroid Prediction. Inspired by score-based methods such as Hough Voting [14, 35], we introduce the following loss term to optimize centroid prediction by aggregating more contextual information.

$$L_{cent} = \frac{1}{|N_{ins}|} \frac{1}{|N_g|} \sum_i \sum_j (|\Delta \hat{d}_{ij} - \Delta d_{ij}| + |\hat{d}_{ij} - \bar{d}_i|) \cdot I_{ij}, \quad (4)$$

where

$$\bar{d}_i = \frac{1}{|N_g|} \sum_j \hat{d}_{ij}, I_{ij} \in \{0, 1\}, \quad (5)$$

where $|N_{ins}|$ is the number of candidate instances, $|N_g|$ is the number of points used to predict the instance center, Δd_{ij} is the ground truth offset of the j th point to the i th instance, $\Delta \hat{d}_{ij}$ is the predicted value of this offset, and \bar{d}_i is the mean destination of i th instance. I_{ij} is the indicator for whether to count this term into the loss (1) or not (0). Subsequent to centroid prediction, we aggregate the selected point features to extract centroids of instance via shared MLPs and symmetric functions.

3.2 Diffusion Branch of LDRadSSD

Occupancy Evidence Map. Occupancy evidence is usually used to indicate the probability of object existence. Our following diffusion and sampling modules will benefit from the occupancy evidence map we develop in this part as the computational workload on 2D data is much smaller than that on 3D data. Intuitively, point density can directly represent object existence and for point cloud, point density has been demonstrated to be beneficial to object detection [6]. Inspiring works such as [19, 26] offered another access to compute the evidence based on the signal-noise-ratio (SNR) of point cloud. In the present research, we only use density-based occupancy evidence for further calculation. Specifically, we obtain density-based occupancy evidence by defining each point's local density as the number of its neighbouring points within a given radius, and the larger the density is, the more possible the local space is occupied.

We further compress all points' occupancy evidence into a specific range such as $[0, 1]$. Then we adopt Kernel Density Estimation (KDE) to approximate the distribution of occupancy evidence [4, 6] in BEV by supposing all points' occupancy evidence follow Gaussian distribution in their surrounding environment [19]. In specific, for the point set $\{\mathbf{p}_i = \{\mathbf{x}_{\mathbf{p}_i}, \mathbf{f}_{\mathbf{p}_i}\} | i = 1, \dots, N\}$, where $\mathbf{x}_{\mathbf{p}_i}$ are spatial measurements of these points, $\mathbf{f}_{\mathbf{p}_i}$ are other measurements such as Doppler and SNR, and N is the number of points, we have

$$\hat{\mathcal{P}}(x, y) = \frac{1}{Nh^2} \sum_{i=1}^N \pi_i \cdot \mathcal{N}\left(\frac{(x - x_i)(y - y_i)}{h^2}\right), \quad (6)$$

where $\mathcal{N}(\cdot)$ is the chosen Gaussian kernel, h is bandwidth and the additive weight is defined in terms of point-wise occupancy evidence p_i as follows.

$$\pi_i = \frac{p_i}{\sum_{j=1}^N p_j}. \quad (7)$$

We use Eq. (6) to obtain continuous occupancy likelihood function. In this case, we grid the whole scene into identical cells along x- and y-axis in BEV and let the cell center represent each cell. As a result, an occupancy evidence map can be calculated by plugging the coordinate matrix of cell centers \mathbf{M}_{coor} in function $\hat{\mathcal{P}}$ as follows

$$\mathbf{M}_{occ} = \hat{\mathcal{P}}(\mathbf{M}_{coor}). \quad (8)$$

Forward Diffusion. In model training phase, different from sequentially adding Gaussian noises to the whole occupancy evidence map, we introduce noises into the map according to local point density considering the poor semantic information carried by the map. Conventionally, the forward diffusion process is modeled as a Markov chain [15]:

$$q(x_i^{(1:T)} | x_i^{(0)}) = \prod_{t=1}^T q(x_i^{(t)} | x_i^{(t-1)}), \quad (9)$$

where $q(x_i^{(t)} | x_i^{(t-1)})$ is the Markov diffusion kernel. The kernel adds noise to the image at the previous time step and models the distribution of pixel values at the next time step. However, as we aim to outline the real scene reflected by point cloud, it is not necessary to keep adding noises to the image to reach a standard Gaussian noise. Following the convention of [1], we define the diffusion kernel according to our detection task as follows:

$$x^{(t)} = D(x^{(0)}, t), \quad (10)$$

with $D(x^{(0)}, 0) = x^{(0)}$, where $D(\cdot)$ is the degradation operator of $x^{(0)}$ with severity t . To take advantage of the occupancy evidence map, we elaborate the degradation operator as follows.

There is a fact of mmw radar point cloud that local point clouds of instances are much denser than background points and how point clouds' spatially distribute varies a lot across the types of road users. For example, point clouds of pedestrian and cyclist are more likely to fall in their ground truth 3D bounding boxes while that of car tends to fall on edges of their bounding boxes. This fact will result in severe imbalance in detection accuracy for different type of objects. Therefore, within the occupancy evidence map, we first grade each pixel value S in \mathbf{M}_{occ} into 5 levels as follows

$$S = \begin{cases} 0.0, & \text{if } S < 0.15, \\ 0.3, & \text{if } 0.15 \leq S < 0.35, \\ 0.5, & \text{if } 0.35 \leq S < 0.55, \\ 0.7, & \text{if } 0.55 \leq S < 0.75, \\ 0.9, & \text{if } S \geq 0.75. \end{cases} \quad (11)$$

We secondly use Monte Carlo method to sample fix-number elements in \mathbf{M}_{occ} with the aid of Eq.(6) and update their values by adding 0.1 if their values $S \in [0.1, 0.8]$, or reset their values to 0.1 if their initial values are 0 and the distance between them and their nearest non-zero neighbour is within a specific threshold. Specifically, if an element's value reaches 0.9, we stop updating its value. Eventually, after T times update, we obtain diffusion processed maps.

Reverse Diffusion. We regard the generation process as the reverse of the diffusion process, where image sampled from Gaussian noise $p(x_i^{(T)})$ that approximates $q(x_i^{(T)})$ is given as the input. Then, the image is passed through the reverse Markov chain and finally form the desired output image. Unlike the forward diffusion process that simply adds noise to the points, the reverse process aims to recover the desired image from the input noise, which requires training from data. The reverse diffusion process for generation can be formulated as:

$$p_{\theta}(x^{(0:T)}|z) = p(x^{(T)}) \prod_{t=1}^T p_{\theta}(x^{(t-1)}|x^{(t)}, z), \quad (12)$$

where z is the latent encoding the diffusion processed map. In canonical diffusion methods, z is supposed to follow a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Essentially, in LDRadSSD, the diffusion processed map can be regarded as a mask of the Gaussian distribution, which can accelerate the reverse Markov process by pruning the image of pure Gaussian noise. In other words, the reverse Markov process can directly reach at the key intermediate state of it.

Subsequently, we design a restoration operator R that approximately inverts D . This operator has the property that

$$R(x^{(t)}, t) \approx x^{(0)}. \quad (13)$$

In the present research, we use ground truth bounding boxes to generate ground truth point clouds and project them onto a BEV image. Hence, we use the following expression to govern the model training process:

$$\min_{\theta} \mathbb{E}_{x \sim \mathcal{X}} \|R_{\theta}(D(x, t), t) - x_g\|, \quad (14)$$

where x_g is the generated ground truth image, x denotes a random image sampled from distribution \mathcal{X} and $\|\cdot\|$ denotes the l_1 norm.

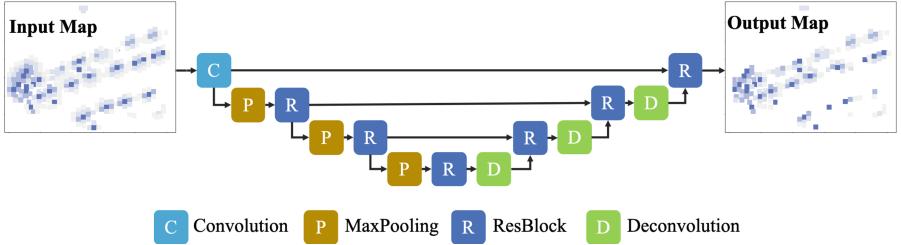


Fig. 3. The ResUnet structure used for reverse diffusion.

As for the restoration operator R , as many researchers focusing on image denoising do, we apply denoising Unet structure, as shown in Fig. 3. Repeating this structure for T times, we can obtain a reference map for 3D upsampling module of LDRadSSD.

3D Upsampling. The 3D upsampling module in LDRadSSD takes the reference map produced by the reverse diffusion process and contextual centroids predicted by the downsampling process as its input. These predicted contextual centroids are a sparse collection of 3D location information of potential objects. For each non-zero pixel in the reference map, we adopt randomly upsampling fix-number of point clouds according to the value of the pixel, i.e., the larger the value is, the more points will be generated proportionally. Specifically, in the pitch direction, we use Monte Carlo sampling to determine height dimensions of generated points according to the Gaussian distribution $\mathcal{N}(h, 1)$, where h is the height dimension of the nearest contextual centroid to the pixel. For other measurements, such as SNR and Doppler, of newly generated points, we also follow this fashion.

3.3 Other Details of LDRadSSD

Road User Detection. In object detection module, we first aggregate the point cloud selected by the main pipeline of LDRadSSD and that generated by the diffusion branch of LDRadSSD. The detection module replicates all layers contained in main pipeline of LDRadSSD because more objects and sharper details are expected to be detected in the densified point cloud. Then, by applying 3D non-maximum-suppression (NMS), we can determine bounding boxes of road users.

Free Space Detection. We adopt a rule-based method to implement free space detection in the post processing stage. Inspired by the work in [9], in the model preparation stage, we use M_{coor} to compute the distance and orientation of all cells recorded by the matrix relative to the origin. We store them as dictionaries to facilitate the construction of the free space polygon. Then, by Eq. (8), we can obtain the updated M_{occ} of the newly generated point cloud and we group

Table 1. Network settings of main pipeline of LDRadSSD.

Layer	Sampling Method	Grouping Radius	Points	Features
1	D-FPS	[0.2, 0.8]	1024	64
2	Ctr	[0.8, 3.2]	512	128
3	Ctr	—	384	256
4	Vote	—	384	—
5	—	[3.2, 6.4]	384	512

non-zero elements in it with respect to their orientations degree by degree and within each group we find the nearest element to the origin. Lastly, we connect those selected elements to form the free space polygon. With those mentioned dictionaries, the free space detection can be accelerated by parallel computation.

Loss Functions. Besides the aforementioned loss functions, LDRadSSD also takes prediction loss into consideration. As most 3D detection approaches do, we count in box prediction loss, which consists of losses from predictions of box center, size and orientation. We also use mean square error (MSE) to evaluate the prediction of free space polygon. So the prediction loss function of LDRadSSD is

$$L_{pre} = L_{loc} + L_{size} + L_{ori} + L_{fs}. \quad (15)$$

4 Experiments

4.1 Implementation Details

Baselines and Model Settings. Since we intend to develop a real-time detector for objects and free space, we take relevant those state-of-the-art as baselines. PointPillars [8], was selected as representatives of voxel-based models because they showed excellent speed in other studies. PDV [6] adopted an architecture including transformer and it outperformed PointPillars by considering point density. IA-SSD [35] and 3DSSD [34] are considered as a baseline because they are famous for their rapidity and they also proposed highly effective point sampling strategies. RadarNet [32] was specifically designed for object detection with radar point cloud. For free space detection, we choose Deformable Polygon [3], NVRadarNet [19] and Occupancy-Net [16] as references because they are known for real-time drivable area calculation. Table 1 illustrates settings of group abstraction layers in main pipeline of LDRadSSD. Specifically, we use two layers employing the mentioned centroid-aware sampling ('Ctr') to select 256 point features, followed by voting-based centroid prediction layer ('Vote'). Instance centroids are predicted by three MLP layers and the terminal outputs for object detection with instance classification and regression are produced by another 3 MLP layers. The KDE of ball queries is calculated with bandwidth $h=0.25$. All our experiments are executed on a single RTX 3080 Ti GPU.

Table 2. Quantitative object detection performance of different methods on the VOD 5-frame dataset.

Method	Car	Pedestrian	Cyclist	AP Std.	mAOS	Speed
PointPillars	40.4	36.5	53.0	8.5	38.7	11.2
IA-SSD	35.9	50.2	75.5	17.3	49.9	18.9
3DSSD	35.1	54.1	77.1	18.2	50.6	41.7
PDV	38.2	38.5	73.0	17.1	46.8	76.9
RadarNet	36.7	<u>57.1</u>	78.7	17.5	53.2	40
LDRadSSD*	<u>48.4</u>	54.9	<u>80.1</u>	14.4	<u>64.0</u>	<u>18.2</u>
LDRadSSD	58.5	63.1	82.1	<u>11.9</u>	71.1	41.7

Datasets and Evaluation Metrics. We validate LDRadSSD on View-of-Delft (VOD, [18]) dataset and the nuScenes dataset [2]. NuScenes contains sensor data from 1 LiDAR and 5 radars, including 1 front radar. In contrast, VOD dataset only contains front radar data and it has the advantage that the proportion of annotation for objects such as cars, pedestrians, and cyclists are more balanced than nuScenes. Since we focus on main road users, we evaluate on three challenging road user classes: cars, pedestrians and cyclists. For nuScenes, we follow the official training/validation split with 700/150 logs each. We report the model performance on object detection. Mean Average Precision (mAP) is used as the detection metric, which is defined on center distance in BEV between the detection and the label. The final AP is averaged over four different distance thresholds (0.5 m, 1 m, 2 m and 4 m). In VOD dataset, we use 5-frame accumulated data of it in our experiments as it is of higher resolution and we split the dataset into a training, validation, and testing set in a ratio of 59%/15%/26%. For object detection in VOD, besides mAP, we also adopt mean Angle of Similarity (mAOS), which indicates prediction accuracy of object orientation. For free space, we employ intersection over union metrics (IoU-smooth, IoU-gt) [3] to evaluate the smoothness and precision of predicted free space. The detection domain of LDRadSSD is set as 35 m in the lateral direction and 50 m in forward direction.

4.2 Experimental Evaluation

In this section, we test the object detection performance of LDRadSSD on the VOD and nuScenes datasets but more detailed test will be conducted on VOD, as it contains more balanced annotation information compared to nuScenes.

Test on VOD. Road User Detection. As shown in Table 2, we employ particular indicators, mAOS and Speed, to compare model performance. LDRadSSD* is the variant of LDRadSSD with hyperparameter $T = 1$ yet the same amount of upsampled points in diffusion branch. In the table, we learn that PointPillars is fastest but not the most efficient method owing to its lackluster detection

Table 3. Comparison of free space detection algorithms.

Method	IoU-gt	IoU-smooth
Deformable Polygon	<u>0.73</u>	<u>0.90</u>
NVRadarNet	0.59	—
Occupancy-Net	0.44	—
LDRadSSD	0.88	0.91

Table 4. Comparison of object detection methods on nuScenes.

Method	Car	Pedestrian	Cyclist	AP Std.
PointPillars	<u>33.4</u>	29.5	46.2	5.04
IA-SSD	29.2	<u>43.0</u>	71.1	12.3
3DSSD	28.8	49.2	73.3	12.9
PDV	31.7	32.2	65.7	11.3
RadarNet	30.5	<u>53.6</u>	79.3	14.1
LDRadSSD	51.3	57.1	<u>77.0</u>	<u>7.8</u>

performance. For Car detection, LDRadSSD and its variant display prominent advantages with 10% to 20% increase in AP. Meanwhile, for Pedestrian and Cyclist detection, LDRadSSD improved detection AP of them in general. The decrease in AP Std. implies that LDRadSSD can detect different types of road users more evenly than other methods. In addition, mAOS of LDRadSSD also dramatically surpasses those of other approaches by 10% to almost 20%, indicating that LDRadSSD can more accurately determine the orientation of objects. However, to obtain such improvement in detection performance, we paid the price of more than a half reduction in model running speed, from 18.2 to 41.7 ms per frame, which barely meets the real-time requirements in driving.

Free Space Detection. When compared with other models, as shown in Table 3, our LDRadSSD outperforms those baseline models such as Deformable Polygon in terms of IoU-gt and IoU-smooth. In particular, IoU-smooth is not applicable for NVRadarNet and OccupancyNet.

Test on NuScenes. As shown in Table 4, all results are evaluated by mAP with 40 recall positions via the VOD evaluation server with 3D IoU (0.5 for car, 0.25 for pedestrian and cyclist). The best result of each column is shown in bold, and the suboptimal results are underlined. As we can see, LDRadSSD shows significant strength in identifying Car, with almost 18% higher than the suboptimal method. Meanwhile, the detection ability for Pedestrian and Cyclist of LDRadSSD also outperforms most baselines. By introducing the standard deviation of mAP (AP Std.), we can indicate more meaningful detail of model performance, that is, the smaller AP Std. is, the more even the detection ability of the model on different types of objects is. In this way, although LDRadSSD

Table 5. Results of ablation study on T .

T	AP	AP Std.	Spe.	T	AP	AP Std.	Spe.
1	61.1	14.4	18.2	6	67.2	11.6	62.5
2	61.9	15.1	21.3	7	65.4	12.5	76.9
3	63.8	13.9	24.4	8	66.2	11.0	111.1
4	66.2	12.2	30.3	9	64.3	13.3	142.9
5	67.9	11.9	41.7	10	63.8	13.7	200.4

achieved the second place in terms of AP Std., it is proven to have the best comprehensive performance.

4.3 Ablation Studies

Ablation on T . The number of points within one 5-frame accumulated VOD point cloud is around 1500, so if we generate 1500 new points using an upsampling tool, then the input data for detection module would be equivalent to the accumulated data of 10 frames, or even more. In this ablation study, T is scheduled to be 1,...,10 and as shown in Table 5, we learn from this table that as the diffusion process getting more and more complex, the average AP of object detection task increases firstly but starting from $T=5$ the model performance shrinks steadily in all aspects, especially the running speed (Spe. in milli-second per frame).

Ablation on Key Modules. The occupancy evidence map (OEM) contributes in contextual centroid prediction and also makes a difference in detection head. As we are mainly interested in its impact on obstacle detection, we remove this tool and replace the detection head with that used in IA-SSD [35]. Consequently, the prediction precision of all objects drops down and that of Car and mAOS again falls severely by 8.7% and 14.4%, respectively. In contrast to the proposed upsampling strategy, the occupancy evidence map plays greater impact on the model. Specifically, this again proves that the map tool improved object orientation prediction.

5 Conclusion

Our LDRadSSD improved detection unbalance among car, pedestrian, and cyclist with its upsampling strategies, and the occupancy evidence map-based centroid prediction module works superbly in promoting the detection accuracy of Car and object orientation. In experiments, our method outperformed other SOTA approaches especially with respect to mAOS. The performance of LDRadSSD definitely indicates that point cloud of millimeter-wave radar tends to produce reliable detection for vulnerable road users, i.e., pedestrians and cyclists.

References

1. Bansal, A., et al.: Cold diffusion: inverting arbitrary image transforms without noise. arXiv preprint [arXiv:2208.09392](https://arxiv.org/abs/2208.09392) (2022)
2. Caesar, H., et al.: nuscenes: a multimodal dataset for autonomous driving. In: CVPR, pp. 11621–11631 (2020)
3. Gao, X., Ding, S., Vanas, K., Dasari, H.R., Soderlund, H.: Deformable radar polygon: A lightweight and predictable occupancy representation for short-range collision avoidance. arXiv preprint [arXiv:2203.01442](https://arxiv.org/abs/2203.01442) (2022)
4. Hermosilla, P., Ritschel, T., pau Vazquez, P., Àlvar Vinacua, Ropinski, T.: Monte carlo convolution for learning on non-uniformly sampled point clouds. In: ACM TOG, vol. 37, pp. 1–12 (2018)
5. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS, pp. 6840–6851 (2020)
6. Hu, J.S.K., Kuai, T., Waslander, S.L.: Point density-aware voxels for lidar 3d object detection. In: CVPR, pp. 8469–8478 (2022)
7. Kingma, D.P., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. In: NeurIPS, pp. 21696–21707 (2021)
8. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: fast encoders for object detection from point clouds. In: CVPR, pp. 12697–12705 (2019)
9. Li, Y., Yu, S., Li, M., Jia, Z., Song, Y.: Qdtree: quasi-density-tree accelerates free space detection of mmw radar point cloud. In: Int. Conf. on Intel. Traffic and Transpor. (ICITT) (2023)
10. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR, pp. 2117–2125 (2017)
11. Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: CVPR, pp. 8895–8904 (2019)
12. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV, pp. 10012–10022 (2021)
13. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: CVPR, pp. 2837–2845 (2021)
14. Luo, S., Hu, W.: Score-based point cloud denoising. In: ICCV, pp. 4583–4592 (2021)
15. Lyu, Z., Kong, Z., Xu, X., Pan, L., Lin, D.: A conditional point diffusion-refinement paradigm for 3d point cloud completion. arXiv preprint [arXiv:2112.03530](https://arxiv.org/abs/2112.03530) (2021)
16. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR, pp. 4460–4470 (2019)
17. Nichol, A., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML, pp. 8162–8171 (2021)
18. Palffy, A., Pool, E., Baratam, S., Kooij, J.F.P., Gavrila, D.M.: Multi-class road user detection with 3+1d radar in the view-of-delft dataset (2022)
19. Popov, A., et al.: Nvradarnet: real-time radar obstacle and free space detection for autonomous driving. In: International Conference on Robotics and Automation (ICRA), pp. 6958–6964 (2023)
20. Prabhakara, A., et al.: Radarhd: Demonstrating lidar-like point clouds from mmwave radar. In: International Conference on Mobile Computing and Networking, pp. 1–3 (2023)
21. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV, pp. 9277–9286 (2019)

22. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: CVPR, pp. 652–660 (2017)
23. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
24. Qian, R., Garg, D., Wang, Y., You, Y.: End-to-end pseudo-lidar for image-based 3d object detection. In: CVPR, pp. 5881–5890 (2020)
25. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR, pp. 770–779 (2019)
26. Sless, L., Cohen, G., Shlomo, B.E., Oron, S.: Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation. In: International Conference on Computer Vision Workshops (ICCVW) (2019)
27. Song, Q., Mei, K., Huang, R.: Attanet: attention-augmented network for fast and accurate scene parsing. In: AAAI, pp. 2567–2575. No. 3 (2021)
28. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint [arXiv:2011.13456](https://arxiv.org/abs/2011.13456) (2020)
29. Thakur, S., Peethambaran, J.: Dynamic edge weights in graph neural networks for 3d object detection. arXiv preprint [arXiv:2009.08253](https://arxiv.org/abs/2009.08253) (2020)
30. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. In: ACM TOG, vol. 38, pp. 1–12 (2019)
31. Watson, D., Chan, W., Ho, J., Norouzi, M.: Learning fast samplers for diffusion models by differentiating through sample quality. In: ICLR (2021)
32. Yang, B., Guo, R., Liang, M., Casas, S., Urtasun, R. (eds.): Radarnet: Exploiting radar for robust perception of dynamic objects. Springer (2022)
33. Yang, L., et al.: Diffusion models: a comprehensive survey of methods and applications. ACM Comput. Surv. (2022)
34. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: point-based 3d single stage object detector. In: CVPR, pp. 11040–11048 (2020)
35. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In: CVPR, pp. 18953–18962 (2022)
36. Zhou, F., Chaib-draa, B., Wang, B.: Multi-task learning by leveraging the semantic information. In: AAAI, pp. 11088–11096. No. 12 (2021)



LBRFL: Lightweight Privacy-Preserving Federated Learning with Byzantine-Robustness

Pingzhang Shen^{1,2}, Shengnan Zhao^{2(✉)}, Jun Xu², Chuan Zhao^{2(✉)},
Zhenxiang Chen³, and Shanqing Guo^{2,4}

¹ School of Information Science and Engineering, University of Jinan,
Jinan 250022, China

² Quan Cheng Laboratory, Jinan 250103, China
zsn.sdu@gmail.com, ise_zhaoc@ujn.edu.cn

³ Shandong Provincial Key Laboratory of Ubiquitous Intelligent Computing,
University of Jinan, Jinan 250022, China

⁴ School of Cyber Science and Technology, Shandong University, Qingdao, China

Abstract. Privacy-Preserving Federated Learning (PPFL) enables efficient model training and prediction while safeguarding clients' data privacy. However, malicious clients can launch poisoning attacks by manipulating local training data or model updates sent to the server which may lead to an incorrect global model. Existing defense methods for detecting malicious clients typically fail to balance privacy, overhead, and accuracy. To address this, there is a pressing need for an efficient PPFL framework that can resist poisoning attacks. In this paper, we propose a lightweight PPFL framework LBRFL that simultaneously achieves privacy, efficiency, and accuracy. Specifically, we introduce a lightweight privacy-preserving clustering method that leverages secure Euclidean distance computation to process masked gradients uploaded by clients efficiently. To minimize the impact of malicious clients and maintain the accuracy of the global model, we propose an efficient Byzantine-Robustness secure aggregation method by introducing removable masks. In addition, experiments on the MNIST and CIFAR-10 datasets demonstrate that, compared to existing methods, LBRFL effectively defends against common poisoning attacks in both independent and identically distributed (IID) and non-IID data settings without heavy encryption costs.

Keywords: Privacy-Preserving · Federated Learning · Poisoning Attack · Byzantine-Robustness

1 Introduction

Federated Learning (FL) [19] has emerged as a promising distributed machine learning paradigm to address the issue of data silos in model training. In the classical FL framework, each client trains the model locally and uploads the model

weights to a central server, rather than transmitting the raw data. The server then utilizes the aggregation algorithm to aggregate these weights, forming an updated global model. This process inherently provides a level of data privacy since clients only share model updates instead of raw data. However, FL faces significant security and privacy challenges. Recent studies [9, 11, 31] have demonstrated that even when clients only upload gradient information, their privacy may still be at risk. Adversaries could compromise the central server, potentially inferring properties of the training data or even reconstructing the original data from the gradients.

Privacy-Preserving Federated Learning (PPFL) schemes have been proposed to mitigate privacy concerns. The PPFL schemes leverage advanced cryptographic techniques such as Secure Multi-Party Computation (SMPC) [2, 7], Homomorphic Encryption (HE) [8, 30], and Differential Privacy (DP) [26, 27] to protect the privacy of data during the FL process. By using these cryptographic methods, PPFL aims to enhance the privacy and security of FL, ensuring that sensitive information remains confidential even in the presence of adversaries. However, despite these additional security measures, PPFL frameworks are not immune to poisoning attacks. Recent studies [1, 6, 22] have shown that PPFL remains vulnerable to poisoning attacks. These attacks can be broadly categorized into untargeted and targeted poisoning attacks. Untargeted poisoning attacks aim to degrade the overall performance of the global model, thereby increasing the test error rate indiscriminately. Targeted poisoning attacks aim to manipulate the global model to produce specific incorrect outputs for certain test inputs without affecting its performance on other inputs.

Due to the inherent trade-off between privacy and robustness in FL, defending against poisoning attacks by malicious clients in PPFL remains a challenging issue. Most existing studies focus on either privacy preservation or robustness independently. Current solutions that try to address both issues at the same time often use HE [16, 17] or Local Differential Privacy (LDP) [14, 25]. HE-based methods allow calculations on encrypted data, protecting privacy while still enabling the detection of malicious activities. However, these methods come with significant computational and communication overhead, making the FL process slow and resource-heavy. On the other hand, LDP-based approaches add noise to the data at the client side to protect privacy and then attempt to detect malicious behavior under the influence of this noise. Although LDP-based methods are less computationally demanding than HE-based schemes, the introduction of noise can significantly reduce the model's accuracy.

To ensure the robustness of the entire FL while protecting privacy and avoiding the high computational cost of HE and the accuracy loss of LDP, we propose a lightweight malicious detection PPFL scheme (LBRFL). The main contributions of this paper can be summarized as follows.

- We propose a lightweight PPFL scheme LBRFL with removable masks as the underlying technology. The LBRFL designed to resist poisoning attacks from malicious clients and protect clients' privacy from honest-but-curious servers. The use of removable masks avoids complex computational overhead and prevents accuracy loss.

- We propose a novel lightweight privacy-preserving Euclidean distance clustering method that enables malicious detection without compromising privacy. Based on this, we further introduce a new Byzantine-Robustness secure aggregation technique to minimize the impact of malicious gradients during global aggregation.
- We provide a comprehensive security analysis demonstrating that the scheme ensures client privacy while achieving Byzantine-Robustness. Besides, the experiments on MNIST and CIFAR10 datasets indicate the proposed LBRFL has robust accuracy and lightweight performance against common poisoning attacks in both IID and non-IID.

The remainder of this paper is organized as follows. In the Sect. 2, some related works are reviewed. Then, In Sect. 3, we overview the problem statement. Next, we propose the lightweight anomaly detection privacy preserving federated learning scheme LBRFL in Sect. 4, followed by the analysis of the privacy and the robustness in Sect. 5. Finally, we evaluate the performance of the proposed FL scheme in Sect. 6, and draw the conclusion in Sect. 7.

2 Related Works

Currently, numerous defenses focus on reducing the influence of statistical outliers induced by poisoning attacks during the aggregation of local model updates. For instance, Krum [3] selects a single model update with the smallest squared Euclidean distance to serve as the new global model update. The Trimmed-mean approach [29] computes the coordinate-wise mean of model update parameters after discarding extreme values, subsequently using these means to update the global model. Similarly, the Median method [29] updates the global model by computing the coordinate-wise median of local model updates. FLTrust [5] employs a small, clean dataset to compute a server-side update, which acts as a baseline to establish trust in local model updates. However, these methods do not simultaneously address privacy concerns, making them unsuitable for PPFL scenarios.

To defend against poisoning attacks in PPFL, researchers have explored various strategies. For honest-but-curious servers, Homomorphic Encryption (HE)-based and Local Differential Privacy (LDP)-based solutions are popular. Ma et al. [17] propose a privacy-preserving defense strategy based on two-trapdoor HE to resist model poisoning, which can identify encrypted malicious gradients based on cosine similarity. Liu et al. [16] evaluate user reliability by using the median coordinate in the ciphertext state to identify malicious gradients that significantly deviate from the baseline. Le et al. [14] propose an Ada-PPFL scheme based on LDP that resists poisoning attacks from malicious clients by applying Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to noisy gradients to filter out malicious ones. Additionally, the inclusion of noise ensures user privacy. Wang et al. [25] design a FL scheme that integrates distributed differential privacy (DDP), gradient encryption, and gradient range proofing. The DDP mechanism adds noise to local model gradients on the user side, while

range proofing techniques on the server side filter out gradients from clients that fall outside a reasonable range. This approach effectively mitigates the impact of malicious user data on the model. However, HE-based solutions incur significant communication and computational overhead, whereas LDP-based solutions can degrade model accuracy due to noise accumulation.

Consequently, further research is warranted to devise defenses against poisoning attacks in PPFL that are both cost-effective and maintain high accuracy. Our proposed solution employs a lightweight client-side malicious detection framework, which protects user privacy while defending against malicious client poisoning attacks. Based on the previous discussion, we summarize the comparison between different schemes in Table 1. In the second column, “Privacy” indicates whether the scheme addresses privacy concerns. The third column, “Robustness,” denotes whether the scheme is robust against poisoning attacks by malicious clients. Additionally, the fourth and fifth columns consider whether the scheme incurs accuracy loss and whether it involves huge cost, respectively.

Table 1. Comparison of Existing Schemes

Approach	Privacy	Robustness	No Accuracy Loss	No Huge Cost
Krum [3]	✗	✓	✓	✓
[29]	✗	✓	✓	✓
FLTrust [5]	✗	✓	✓	✓
The HE-based Approach [16] [17]	✓	✓	✓	✗
The LDP-based Approach [14] [25]	✓	✓	✗	✓
Ours	✓	✓	✓	✓

3 Problem Statement

3.1 System Model

As depicted in Fig. 1, our system model adopts a dual server architecture, and the specific structure is as follows:

- The sever S_1 : Firstly, S_1 is used to negotiate masks with clients during the offline phase and has the full share of the masks. Secondly, S_1 is used to decrypt the aggregated results and perform global model updates.
- The sever S_2 : S_2 is used to perform Byzantine-Robustness secure aggregation and malicious detection of gradients, thereby filtering out malicious gradients uploaded by malicious clients.
- Data Owners: All data owners, also called clients, collaboratively train a uniform model with the coordination of the sever. For privacy reasons, each client trains the model locally over the private data on device, then uploads the masked gradients to the S_2 .

3.2 Threat Model

In this paper, we focus on the privacy and robustness vulnerabilities of FL that could be exploited by malicious participants to leak users' privacy or disrupt the entire federated learning process. Therefore, we consider the following threat models:

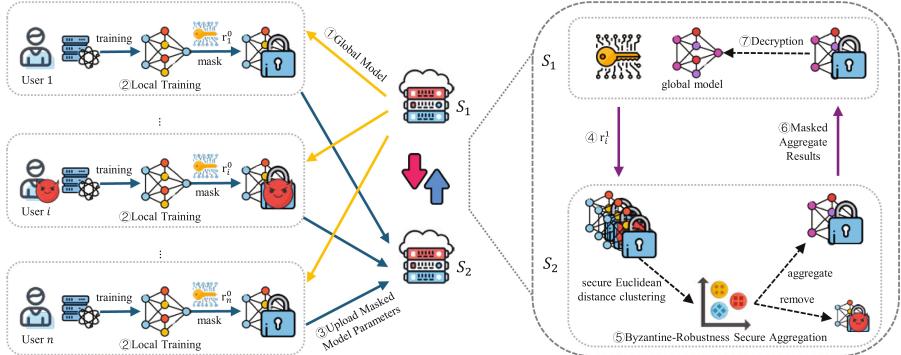


Fig. 1. System model.

- Honest-but-curious sever: while S_1 and S_2 perform all operations honestly according to the protocol, they may still attempt to infer additional information from the gradients they access, potentially compromising clients' data privacy. Referring previous works, we assume there is no collusion between the two servers [20, 23].
- Malicious participant: We assume that each malicious participant possesses a local dataset and a local model for training. Malicious users can manipulate the private data stored on their devices or the information they upload locally to perform various poisoning attacks. Similar to prior works [10, 29], to ensure the practicality and availability of the model, we assume an upper limit on the number of malicious users, specifically, $|M| \leq \frac{|C|-1}{2}$, where M and C denote the number of malicious users and the total number of users, respectively, and $|\cdot|$ represents the cardinality of the set.

4 Proposed Method

4.1 Overview

The proposed scheme is a 4-step protocol. The first step is the offline phase, each client generates a private key to generate a mask, and sends the mask to S_1 . The S_1 generates a mask that matches the mask generated by each client. In the second step, the client uses the input of its local dataset to perform local training, then masks the local gradient with its own mask and uploads it to the

S_2 . In the third step, the S_2 performs Byzantine-Robustness secure aggregation with malicious detection capabilities and provides the aggregation results to S_1 for decryption. Finally, the S_1 decrypts aggregation results and performs global updates. We assume that each client communicates with the server through a private and authenticated channel.

4.2 Offline Phase

In the offline phase, each of the n clients $\{C_1, C_2, \dots, C_n\}$ independently generates a private key k_i . This key is used to create a round-specific mask share $r_{i,t}^0$ for masking the upload gradient, where $r_{i,t}^0 = PRC(k_i, t)$. The $r_{i,t}^0$ is then sent to S_1 , which generates the corresponding mask share $r_{i,t}^1$ for each round. And ensure that in each round of global training $r_t = r_{i,t}^0 + r_{i,t}^1$.

Since the servers do not collude, the use of masks prevents S_2 from eavesdropping on the local private information uploaded by users. Additionally, S_1 only receives aggregated information throughout the federated learning process. Consequently, even if S_1 has knowledge of all the mask information, it cannot compromise the privacy of individual users.

Furthermore, to reduce the communication overhead between clients and S_1 during each round of mask negotiation, k_i can be treated as a long-term key for generating the mask $r_{i,t}^0$. It suffices for S_1 to generate a new mask $r_{i,t}^1$ in each round of global training to ensure the privacy of the entire scheme.

4.3 Local Training

Before entering the local training phase, a global model with random weights w_0 is initialized at S_1 and distributed to each user. In the t -th training iteration, each user C_i independently performs local training using SGD to compute the local gradient $g_{i,t}$. Since each local gradient $g_{i,t}$ contains a large number of elements, using the SGD algorithm effectively reduces the communication overhead between users and the server. After completing local training, users mask their local gradients $h_{i,t}$ with the masks $r_{i,t}^0$ generated during the offline phase and send the masked gradients to S_2 . The use of masks protects local gradients from privacy leakage, and unlike noise, these masks can be fully removed during the decryption phase, ensuring that model training accuracy is not affected. Additionally, the use of removable masks avoids high encryption costs, resulting in a more lightweight performance for our scheme. The detailed process is presented in Algorithm 1.

Algorithm 1. Local Training

Require: n clients $\{C_1, C_2, \dots, C_n\}$ with local training datasets $D = \{D_1, D_2, \dots, D_n\}$, batch size b , local training rate η , number of local training iterations E , number of global training iterations T , sever S_1 and S_2 , initial global weigh w_0 .

Ensure: Masked local gradients $h_{i,t}$.

- 1: **for** each global round $t = 1, 2, \dots, T$ **do**
- 2: **for** each $i \in \{C_1, C_2, \dots, C_n\}$ **do**
- 3: Get global model $w_{t-1}(t \in T)$ from S_1 ;
- 4: Local training: $g_{i,t} = LocalUpdate(w_{t-1}, b, \eta, E)$;
- 5: Masking: $h_{i,t} = g_{i,t} + r_{i,t}^0 = g_{i,t} + PRG(k_i, t)$;
- 6: **end for**
- 7: **end for**
- 8: **return** $h_{i,t}$.

Algorithm 2. Byzantine-Robustness Secure Aggregation

Require: U_1 local model updates $\{h_{1,t}, h_{2,t}, \dots, h_{m,t}\}_{m \in U_1}$, corresponding mask share $r_{i,t}^1$, global training rate ξ .

Ensure: The security aggregation result $y_{sum,t}$, the number of clients included in the aggregation result $|U_3|$.

- 1: **for** each global round $t = 1, 2, \dots, T$ **do**
- 2: S_2 received $h_{i,t}$ from C_i (Denote the set of clients as U_1);
- 3: S_2 request $r_{i,t}^1$ from S_1 ;
- 4: S_1 sends $r_{i,t}^1$ to S_2 ($i \in U_1$);
- 5: S_2 received $r_{i,t}^1$ from S_1 (Denote the set of clients as U_2);
- 6: S_2 calculates $y_{i,t} = h_{i,t} + r_{i,t}^1$ ($i \in U_2$);
- 7: S_2 executes Algorithm 3;
- 8: S_2 calculates $y_{sum,t} = \sum_{i \in U_2} p_{i,t} y_{i,t}$ ($i \in U_3$);
- 9: **end for**
- 10: **return** $y_{sum,t}$ and $|U_3|$.

Algorithm 3. Malicious Detection

Require: $|U_2|$ local model updates $y_{i,t}$.

Ensure: A set of honest clients U_3 , $p_{i,t}$.

- 1: The calculation is performed on S_2 :
- 2: S_2 calculates $|U_2|$ $y_{i,t}$ to get their median value $y_{med,t}$.
- 3: **for** $i \in U_2$ **do**
- 4: Calculate Euclidean distance $d_{i,t} = \sqrt{(y_{i,t} - y_{med,t})^2}$;
- 5: **end for**
- 6: Take the Top-50% local updates of the smallest $d_{i,t}$ client set, denoted as U_3 for aggregation.
- 7: Calculate the proportion of each client in aggregation in U_3 : $p_{i,t} = \frac{1}{\sum_{i \in U_3} \frac{1}{d_{i,t}}}$;
- 8: **return** U_3 and $p_{i,t}$.

4.4 Byzantine-Robustness Secure Aggregation and Malicious Detection

When S_2 receives all the masked gradients sent by the selected clients, it begins Byzantine-Robustness secure aggregation. S_2 requests the corresponding part of the mask $r_{i,t}^1$ for each client from S_1 . Then, S_2 can calculate the gradient $y_{i,t} = h_{i,t} + r_{i,t}^1 = g_{i,t} + r_t$. The value of r_t can be adjusted by S_1 in each round to ensure higher security. And the mask r_t can be removed during the decryption phase, unlike differential privacy noise which is non-removable. Moreover, our aggregation method can handle user dropouts effectively, if users go offline while uploading local parameters, it does not affect the aggregation and decryption process. Therefore, our approach can naturally scale to FL scenarios involving a large number of clients.

During aggregation, we also execute a malicious detection algorithm based on lightweight privacy-preserving clustering to identify and filter out malicious gradients uploaded by compromised clients. The detailed steps of this algorithm are outlined in Algorithm 3. Since we limit the number of malicious clients to $|M| \leq \frac{|C|-1}{2}$, we first sort the received local gradients and select the median $y_{med,t}$ as the reference. Based on $y_{med,t}$, we perform secure clustering based on Euclidean distance. Subsequently, we use the Top-k algorithm to select the top 50% of local gradients with the smallest clustering distances for aggregation. The detailed Byzantine-Robustness secure aggregation process is shown in Algorithm 2. The use of removable masks maintains privacy during malicious detection with lightweight overhead, while minimizing the impact of malicious clients on the aggregation results.

4.5 Decryption and Global Model Update

After S_2 completes the secure aggregation, it sends the aggregation result $y_{sum,t}$ and the number of participating clients $|U_3|$ to S_1 . S_1 decrypts the aggregation results by removing the masks based on r_t for each round and then updates the global model accordingly. S_1 only obtains the aggregated result of the local gradients and cannot analyze each user's individual local information. Therefore, it does not compromise users' privacy.

The overall protocol design of the scheme can be found in Algorithm 4.

5 Security Analysis

*Proposition 1 (Honest But Curious Security, With Curious Servers): There exists a PPT simulator **SIM** such that for the given security parameter k , the sever $S = \{S_1, S_2\}$, and the participating client set $C = \{1, \dots, C\}$, the output of $\text{SIM}_S^{C,k}$ is computationally indistinguishable from the output of $\text{REAL}_S^{C,k}$*

$$\text{SIM}_S^{C,k} \approx \text{REAL}_S^{C,k}. \quad (1)$$

Proof: According to the definition of $\text{REAL}_S^{C,k}$, it consists of all internal state and messages received by the parties in S during the execution of the protocols. We adopt the standard hybrid argument used in [4, 28] to prove this proposition, i.e., given the security parameters k , we define a PPT simulator **SIM** through a series of subsequent modifications to the random variables in $\text{REAL}_S^{C,k}$, so that the output of **SIM** is computationally indistinguishable from $\text{REAL}_S^{C,k}$. The detailed proof is described below.

Hyb₁: We initialize a random variable whose distribution is indistinguishable from $\text{REAL}_S^{C,k}$, the joint views of parties in S in the real protocol execution.

Hyb₂: In this hybrid, we change the behavior of simulated honest users $C_x \in C$, so that each user C_x masks a randomly selected vector β_x with r_x^0 , instead of the original gradient vector g_x . Due to only changing the masked content, the PRG mechanism and the two non-collusive S_1 and S_2 setting guarantees that this hybrid is indistinguishable from the previous one.

Hyb₃: In this hybrid, we simulate the S_1 and each user to generate mask r , which is sampled uniformly at random, rather than through PRG. Due to

Algorithm 4. LBRFL

- 1: **Offline Phase:**
 - 2: Each client executes the Negotiate Mask Share Algorithm during the offline phase.
 - 3: C_i generates a k_i is used to generate the mask $r_{i,t}^0$ generated by itself for each round, and then sends k_i to S_1 . ($i \in \{C_1, C_2, \dots, C_n\}$)
 - 4: **Round 1: (Local Training)**
 - 5: C_i downloads the global model w_t from S_1 for local training. ($i \in \{C_1, C_2, \dots, C_n\}$)
 - 6: C_i executes Algorithm 1. ($i \in \{C_1, C_2, \dots, C_n\}$)
 - 7: C_i sends $h_{i,t}$ to S_2 . ($i \in \{C_1, C_2, \dots, C_n\}$)
 - 8: S_2 receives $h_{i,t}$ from C_i . (The set of clients that S_2 received $h_{i,t}$ from clients is defined as U_1)
 - 9: **Round 2: (Byzantine-Robustness Secure Aggregation and Malicious Detection)**
 - 10: S_2 requests mask share $r_{i,t}^1$ from S_1 based on U_1 . ($i \in U_1$)
 - 11: S_1 calculates $r_{i,t}^1$ based on the k_i uploaded by C_i in the offline stage and the global training round T , so that $r_t = r_{i,t}^0 + r_{i,t}^1$.
 - 12: S_1 send $r_{i,t}^1$ to S_2 .
 - 13: S_2 receives $r_{i,t}^1$ form S_1 , and calculates $y_{i,t} = h_{i,t} + r_{i,t}^1$. (The set of clients that S_2 received $r_{i,t}^1$ from S_2 is defined as U_2)
 - 14: S_2 calculates the median value $y_{med,t}$ in $y_{med,t}$.
 - 15: S_2 executes Algorithm 2 to detect malicious clients.
 - 16: S_2 executes Algorithm 3 for secure aggregation.
 - 17: S_2 send $|U_3|$ and $y_{sum,t}$ to S_1 .
 - 18: **Round 3: (Decryption and Global Model Update)**
 - 19: S_1 receives $y_{sum,t}$ from S_2 and calculates $y_{global,t} = y_{sum,t} - |U_3| \cdot r$.
 - 20: S_1 calculates global update: $w_t = \text{GlobalUpdate}(y_{global,t}, \xi, T)$.
 - 21: Move on to the next round of training, S_1 send w_t to C_i . ($i \in U_1$)
-

the fact that parameters generated by uniform randomness are also uniform randomness, they are indistinguishable from masks generated through PRG. Therefore, ensure that this hybrid is no different from the previous one.

Hyb₄: In this hybrid, we change the model parameters uploaded by users from the S_2 to $\beta_x + r_x^0$ instead of $g_x + r_x^0$. The PRG mechanism and the two non-collusive S_1 and S_2 setting guarantees that this hybrid is indistinguishable from the previous one.

Hyb₅: In this hybrid, we simulate the S_1 to calculate the parameter information uploaded by each client, as the S_1 has the mask r_x^0 . However, since S_1 and S_2 do not collude, S_2 only provides aggregation results to S_1 , therefore this hybrid is indistinguishable from the previous one.

The argument proves that there is a simulator **SIM** sampled from the distribution described above so that its output is computationally indistinguishable from the output of **REAL**. Hence, our proposal holds the security property that the curious S_1 and S_2 learn nothing about users' private data.

Proposition 2: Our proposal ensures the security property that malicious users cannot compromise the privacy of other users.

Proof: In our threat model, malicious clients act as active adversaries, launching poisoning attacks and compromising the privacy of other clients. As shown in [21], they infer the presence of target samples in the training dataset by exploiting malicious local gradients. This type of attack heavily relies on the average aggregation rule of federated learning. Our approach mitigates this attack by performing malicious detection on each local gradient and identifying and discarding malicious gradients before aggregation. Consequently, malicious clients cannot infer whether target samples are used during training, thereby preventing client privacy leakage. Additionally, all communication between clients and the server is conducted in an encrypted state, effectively countering message interception attacks and other direct attempts to obtain client privacy.

6 Experimental Analysis

In this section, we first introduce the experiment setup. Then we evaluate the performance of our scheme.

6.1 Experiment Setup

Datasets: We conduct experiments on MNIST handwritten digits dataset and CIFAR-10 dataset [12]. The training datasets are IID and non-IID.

Training Settings: In each round of global training, 30 out of 100 clients are selected to participate, with the proportion of malicious clients set at 10%, 20%, 30%, and 40%, respectively.

Metrics: To evaluate the performance of our proposed scheme, we compared its accuracy and overall runtime against FedAVG [15], Krum [3], HE-based approach [16], and LDP-based approach [14] under four types of poisoning attacks.

The overall runtime was used to represents both computation and communication overhead. These four types of attacks are parameter tampering [13], data poisoning [24], FGSM attacks [18], and backdoor attacks [1].

Implementations: The implementations of our scheme are achieved by Python 3.9 and Pytorch 1.10.1 on PC with 2.20 GHz Intel Core i7, 16 GB RAM, and Windows11 operating system.

6.2 Experiment Results

Accuracy Evaluation. We evaluate the accuracy performance of our scheme against several other approaches under four different poisoning attacks, varying the proportion of malicious clients in each round across two data distribution scenarios (IID and non-IID).

Accuracy in Parameter Tampering Attack: In parameter tampering attack, we configure malicious clients to upload fake local model parameters to disrupt the aggregation of the global model. Our experiments were conducted under both IID (see Fig. 2) and non-IID (see Figs. 3 and 4) data scenarios. We observed that our scheme and other schemes achieve high accuracy on the MNIST dataset, with no significant differences. This is attributed to the simplicity and ease of learning of the MNIST dataset. However, on the CIFAR-10 dataset, our scheme demonstrates visibly higher accuracy compared to other schemes under varying proportions of malicious clients. We attribute this performance to our Byzantine-Robustness aggregation based on secure Euclidean distance clustering, where we aggregate the top 50% of local parameters with the smallest Euclidean distances from the median. In contrast, Krum only selects a single local parameter for aggregation, and the LDP-based scheme introduces additional noise, negatively impacting their accuracy. Additionally, as shown in Fig. 4, our scheme, similar to Krum and the LDP-based scheme, achieves faster training and convergence compared to the non-attacking FedAVG. This is because these schemes begin processing and filtering the parameters uploaded by clients from the start of training, aggregating more similar parameters, thereby accelerating the entire training process.

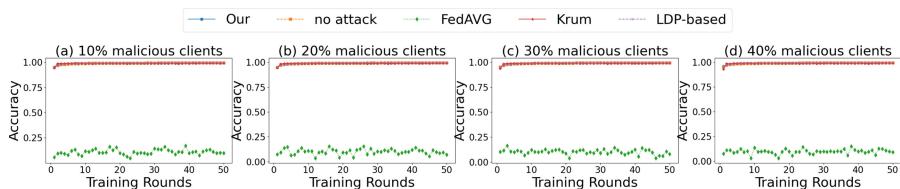


Fig. 2. Parameter tampering attack on MNIST IID

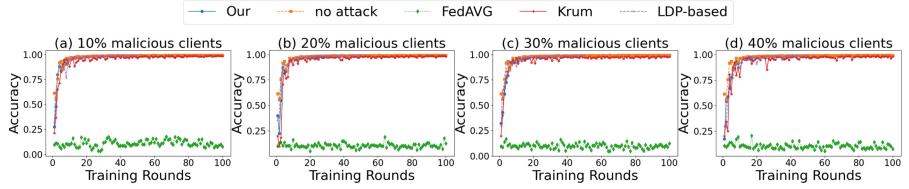


Fig. 3. Parameter tampering attack on MNIST non-IID

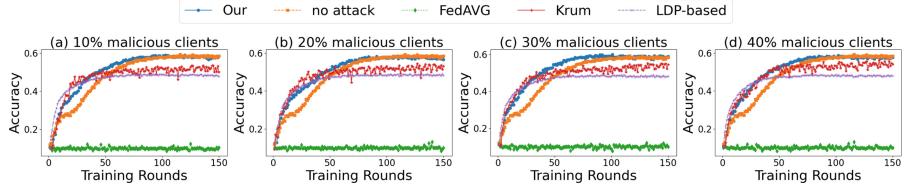


Fig. 4. Parameter tampering attack on CIFAR10 non-IID

Accuracy in Data Poisoning Attack: As shown in Fig. 5, in data poisoning attack, we configure malicious clients to train on poisoned local data and test our approach using the CIFAR-10 dataset with a non-IID distribution. We can see our scheme demonstrates a notable degree of robustness against data poisoning attacks, maintaining accuracy closest to that of training in an attack-free state. Figure 5 also indicates that the Krum algorithm exhibits low robustness against data poisoning attacks, becoming increasingly ineffective as the proportion of malicious clients increases. While the LDP-based scheme shows some resilience to data poisoning attacks, the introduction of excessive noise results in a noticeable decrease in model accuracy.

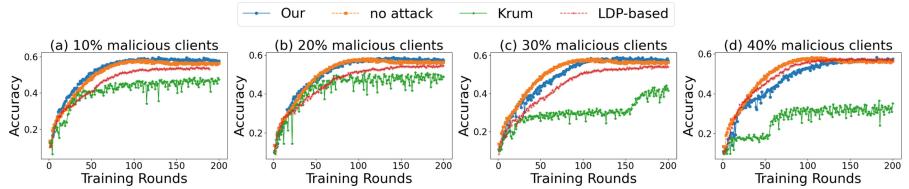


Fig. 5. Data poisoning attack on CIFAR10 non-IID

Accuracy in FGSM Attack: As shown in Fig. 6, in FGSM attack we use the MNIST dataset with a non-IID distribution for testing. Due to the subtle and hard-to-detect perturbations introduced by FGSM attack, the oscillation amplitude of the accuracy curve in Fig. 6 is more pronounced compared to the first two types of attacks. Nonetheless, our approach remains closest to the accuracy of

model training without attacks. Figure 6 also shows that FedAVG and Krum are ineffective against FGSM attacks. The LDP-based scheme, despite its resilience, still exhibits lower accuracy than our scheme due to the impact of noise.

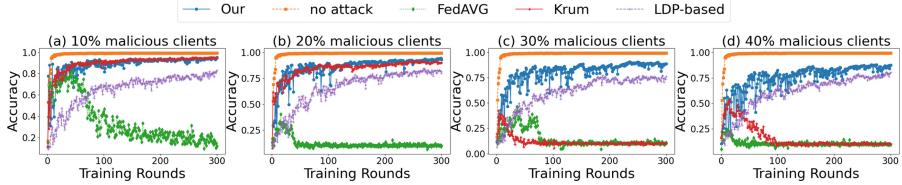


Fig. 6. FGSM attack on MNIST non-IID

Accuracy in Backdoor Attack: As shown in Fig. 7, in backdoor attack, malicious clients inject backdoors by modifying certain pixels in the training dataset image and setting corresponding backdoor labels to carry out backdoor attacks. At the same time, we inject the CIFAR10 non-IID dataset of the backdoor for testing. From the Fig. 7, we can see that both our scheme and LDP-based scheme have certain robustness against backdoor attacks, while Krum lacks robustness against backdoor attacks due to only aggregating one client in each round of global aggregation. For the high success rate of the first few rounds of backdoor attacks in the Fig. 7, we believe that it is due to insufficient communication and aggregation between clients in the early stages of training, which is a normal phenomenon. As the training discussion increases, both our scheme and LDP based scheme show a certain degree of robustness.

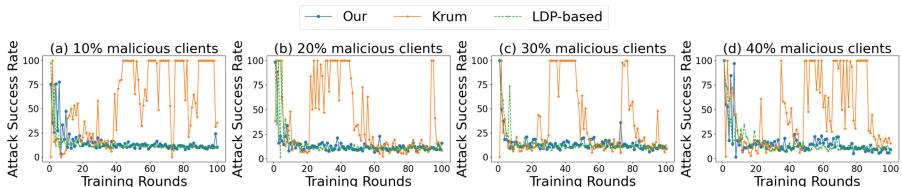


Fig. 7. Backdoor attack on CIFAR10 non-IID

Cost Evaluation. We represent the total computational and communication costs of different schemes using time cost. We train and test on the MNIST non-IID dataset, randomly selecting ten rounds from the global training process with varying proportions of malicious clients to calculate the time costs for each round. Due to the significantly higher time costs of the HE-based scheme compared to

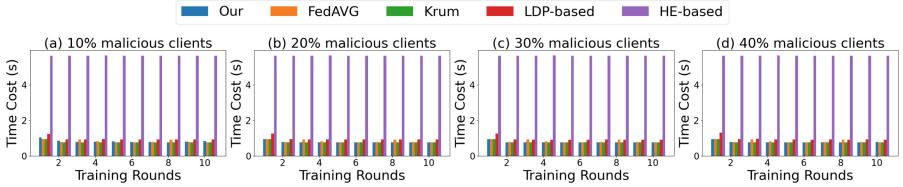


Fig. 8. Time Cost on MNIST non-IID

other approaches, we scale down its experimental data by a factor of 1000. As shown in Fig. 8, our scheme achieves privacy and robustness with lightweight cost. The time cost of our scheme is comparable to FedAVG and Krum, but our scheme offers stronger privacy and robustness. Compared to the HE-based scheme, our method achieves privacy and robustness with significantly lower costs. Additionally, compared to the LDP-based scheme, our method not only provides higher accuracy but also offers advantages in terms of cost.

7 Conclusion

In this paper, we propose a novel framework named LBRFL, which achieves both privacy and robustness in a lightweight manner. In LBRFL, we employ removable masks to ensure privacy throughout the scheme and design a Byzantine-Robustness secure aggregation method based on secure Euclidean distance clustering. This method provides a feasible and lightweight solution for detecting malicious gradients in their encrypted state, thereby preventing poisoning attacks from malicious users. Experimental results demonstrate that LBRFL performs well in terms of accuracy and robustness against various poisoning attacks while maintaining a lightweight performance.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (62472252, 62172258), TaiShan Scholars Program (tsqn202211280, tstp20240828), Shandong Provincial Natural Science Foundation (ZR2024QF131, ZR2023LZH014, ZR2022ZD01, ZR2022MF264, ZR2021LZH007), Shandong Provincial Key R&D Program of China (2021SFJC0401, 2021CXGC010103), Department of Science & Technology of Shandong Province (SYS202201), and Quan Cheng Laboratory (QCLZD202302).

References

1. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics, pp. 2938–2948. PMLR (2020)
2. Bell, J.H., Bonawitz, K.A., Gascón, A., Lepoint, T., Raykova, M.: Secure single-server aggregation with (poly) logarithmic overhead. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1253–1269 (2020)

3. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. *Adv. Neural Inform. Process. Syst.* **30** (2017)
4. Bonawitz, K., et al.: Practical secure aggregation for privacy-preserving machine learning. In: proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191 (2017)
5. Cao, X., Fang, M., Liu, J., Gong, N.Z.: Fltrust: byzantine-robust federated learning via trust bootstrapping. arXiv preprint [arXiv:2012.13995](https://arxiv.org/abs/2012.13995) (2020)
6. Cao, X., Gong, N.Z.: Mpaf: model poisoning attacks to federated learning based on fake clients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3396–3404 (2022)
7. Duan, J., Zhou, J., Li, Y.: Privacy-preserving distributed deep learning based on secret sharing. *Inf. Sci.* **527**, 108–127 (2020)
8. Fang, H., Qian, Q.: Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet* **13**(4), 94 (2021)
9. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients—how easy is it to break privacy in federated learning? *Adv. Neural. Inf. Process. Syst.* **33**, 16937–16947 (2020)
10. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning, pp. 3521–3530. PMLR (2018)
11. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 603–618 (2017)
12. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
13. Kumar, A., Khimani, V., Chatzopoulos, D., Hui, P.: Fedclean: a defense mechanism against parameter poisoning attacks in federated learning. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4333–4337. IEEE (2022)
14. Le, J., Zhang, D., Lei, X., Jiao, L., Zeng, K., Liao, X.: Privacy-preserving federated learning with malicious clients and honest-but-curious servers. *IEEE Trans. Inform. Forensics and Secur.* (2023)
15. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint [arXiv:1907.02189](https://arxiv.org/abs/1907.02189) (2019)
16. Liu, X., Li, H., Guowen, X., Chen, Z., Huang, X., Rongxing, L.: Privacy-enhanced federated learning against poisoning adversaries. *IEEE Trans. Inf. Forensics Secur.* **16**, 4574–4588 (2021)
17. Ma, Z., Ma, J., Miao, Y., Li, Y., Deng, R.H. .: ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning. *IEEE Trans. Inf. Forensics Secur.* **17**, 1639–1654 (2022)
18. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint [arXiv:1706.06083](https://arxiv.org/abs/1706.06083) (2017)
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. *Artif. Intell. Statist.*, 1273–1282 (2017)
20. Mohassel, P., Zhang, Y.: Secureml: a system for scalable privacy-preserving machine learning. In: 2017 IEEE symposium on security and privacy (SP), pp. 19–38. IEEE (2017)

21. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 739–753. IEEE (2019)
22. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
23. So, J., Güler, B., Avestimehr, A.S.: Byzantine-resilient secure federated learning. *IEEE J. Selected Areas Commun.* **39**(7), 2168–2181 (2020)
24. Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L.: Data poisoning attacks against federated learning systems. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12308, pp. 480–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58951-6_24
25. Wang, F., He, Y., Guo, Y., Li, P., Wei, X.: Privacy-preserving robust federated learning with distributed differential privacy. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 598–605. IEEE (2022)
26. Wei, K., et al.: Federated learning with differential privacy: algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **15**, 3454–3469 (2020)
27. Wu, N., Farokhi, F., Smith, D., Kaafar, M.A.: The value of collaboration in convex machine learning with differential privacy. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 304–317. IEEE (2020)
28. Guowen, X., Li, H., Zhang, Y., Shengmin, X., Ning, J., Deng, R.H.: Privacy-preserving federated deep learning with irregular users. *IEEE Trans. Dependable Sec. Comput.* **19**(2), 1364–1381 (2020)
29. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659. Pmlr (2018)
30. Zhang, Q., Xin, C., Wu, H.: Gala: Greedy computation for linear algebra in privacy-preserved neural networks. arXiv preprint [arXiv:2105.01827](https://arxiv.org/abs/2105.01827) (2021)
31. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. *Adv. Neural Inform. Process. Syst.* **32** (2019)



Izhikevich Neurons in NeuCube for Longitudinal Data Classification

Balkaran Singh¹(✉), Sugam Budhraja¹, Maryam Dotorjeh¹, Zohreh Dotorjeh², Edmund Lai¹, and Nikola Kasabov¹

¹ KEDRI, Auckland University of Technology, Auckland, New Zealand
balkaran.singh@aut.ac.nz

² School of Population Health and Center for Brain Research,
University of Auckland, Auckland, New Zealand

Abstract. This study focuses on the integration of the Izhikevich neuron model into the NeuCube framework to improve classification accuracy in longitudinal data. NeuCube is a reservoir spiking neural network architecture and offers advanced features such as spatial neuron connectivity and biologically plausible learning, making it suitable for complex machine learning tasks. For extracting meaningful features from the spiking activity of NeuCube, we evaluated various sampling methods on the classification performance of control and UHR samples from the LYRIKS RNA sequencing dataset. Our findings demonstrate that feature extraction using temporal binning, particularly with 10 bins, yields the highest accuracy across neuron types by effectively capturing fine-grained temporal dynamics. Intrinsic bursting neurons showed superior accuracy across most sampling methods, underscoring their versatility in information transmission.

Keywords: Spiking Neural Networks · Reservoir Computing · Liquid State Machines

1 Introduction

Liquid State Machines (LSMs) are a class of recurrent neural networks known for their ability to process time-varying inputs by transforming them into high-dimensional temporal patterns, making them effective for tasks involving prediction and classification in machine learning [1]. NeuCube enhances LSMs by introducing range of features, including connectivity based on spatial positions of neurons, biologically plausible learning and interpretable insights [2]. Traditionally, LSMs and NeuCube often employ Leaky Integrate-and-Fire (LIF) neurons due to their simplicity and computational efficiency. However, LIF neurons can be limited in their ability to simulate more complex neuronal dynamics [3]. The simple model of spiking neurons, also known as the Izhikevich neuron model, proposed by Eugene Izhikevich, is a compelling alternative. This model combines the biological plausibility of Hodgkin-Huxley-type models with the computational efficiency of LIF neurons and can reproduce a wide variety of spiking

and bursting patterns observed in cortical neurons [4]. This versatility makes the Izhikevich model particularly useful for simulating the diverse behaviors of neurons within a spiking neural network framework. The diversity of spiking and bursting patterns allows for a more comprehensive representation of input signals, which may also help in improving the network's ability to distinguish between subtle differences in input patterns.

When spike trains from different classes are presented to the SNN reservoir, they are expected to elicit distinct dynamic responses. This is because each input generates a unique pattern of neuronal activity, leading to different high-dimensional temporal states in the reservoir. By analysing these states, it becomes possible to infer which class is being presented. Tuning the dynamics of the reservoirs for particular tasks by adjusting the parameters that govern reservoir behaviour can help optimize performance and enhance classification accuracy. This typically involves tuning the synaptic connection weights within the reservoir using STDP or other mechanisms [5–7]. In the simple Izhikevich neuron model, the parameters a , b , c , and d determine the neuron's firing patterns and can be tuned to mimic the dynamic behaviour of many types of cortical neurons [4]. By carefully adjusting these parameters, it is possible to simulate various spiking patterns such as regular spiking, fast spiking, chattering and intrinsic bursting. NeuCube and most LSM models typically utilize LIF neurons capable of simulating regular spiking or fast spiking. However, advancements in neuroscience indicate that single spiking dynamics can lead to transmission failures, whereas burst spikes are transmitted more reliably and play a crucial role in information processing [8]. In this paper, we aim to integrate these findings into the computational modelling of SNNs, potentially boosting performance in machine learning tasks.

Neural activity within reservoirs is decoded into state vectors by various sampling methods (also known as feature extraction), which serve as features for machine learning classifiers. The classifier can then map these features back to their respective classes. These methods can be broadly categorized into rate coding, temporal coding or hybrids between the two. Rate coding methods, such as spike count and mean firing rate, quantify the frequency of neurons firing over a specified period, offering a straightforward measure of neuronal activity. Temporal coding methods, such as inter spike interval (ISI), include spike timing information. The applicability of these methods may depend on the specific neuron behaviours being modelled. Neurons exhibiting regular, high-frequency firing may be better represented by rate coding. Temporally diverse firing patterns, such as bursting or chattering, may benefit more from temporal coding to preserve the temporal dynamics. As such, the choice of sampling method may significantly impact the accuracy and robustness of the classification tasks, making it essential to align the sampling strategy with the neuronal behaviours being simulated.

This paper aims to explore the integration of Izhikevich neurons into the NeuCube framework using the NeuCube-Py library and compare the effects of different sampling methods on the classification accuracy of control and UHR samples

using the LYRIKS RNA sequencing dataset. Additionally, we will investigate the impact of diverse neuron dynamics on the performance of the model. By evaluating these aspects, we seek to enhance the understanding of how complex neuron models, neuron dynamics and diverse sampling techniques can improve the performance of NeuCube in longitudinal data classification tasks.

2 Related Works

Research has explored the application of Izhikevich neurons as input neurons in structured SNN reservoirs, particularly focusing on MRI data to predict EEG signals [9]. Another framework extended this application to predict epilepsy and migraine using EEG data [10]. Experimental studies have been conducted on the Leaky Integrate-and-Fire (LIF) model against the more complex Quadratic Integrate-and-Fire (QIF) and Exponential Integrate-and-Fire (EIF) models [11]. Findings reveal that, for datasets with intricate spatio-temporal features, the QIF and EIF models outperform the simpler LIF model. This emphasizes the necessity of selecting neuron models that match the complexity of the dataset to optimize performance.

Another study experimented with the LIF model and introduced three probabilistic extensions. The results demonstrated that these probabilistic models provided greater flexibility and improved reservoir separability when distinguishing between different classes [12]. Additionally, the performance differences between Izhikevich neurons with bursting dynamics and those with single spiking dynamics in liquid state machines (LSMs) have been examined [13]. Research showed that bursting dynamics significantly improve performance, particularly for complex tasks. Further analysis suggested that bursting dynamics enhance the spiking activity's complexity within the network. This finding is supported by neuroscience research, indicating that burst spiking provides more precise information, is an optimal stimulus for exciting neural cells, and can mitigate synaptic transmission failure. This highlights the potential of using bursting dynamics to increase the efficacy of SNNs.

Traditionally, neuron-related parameters remain constant, with only synaptic weights being optimized. However, optimizing the membrane time constants of each neuron alongside synaptic weights has shown that learnable membrane constants reduce the network's sensitivity to initial values and accelerate the training process [14]. This method outperforms others in terms of accuracy, demonstrating the advantages of dynamically adapting neuron parameters to improve network performance. Introducing adaptive threshold spiking neurons significantly increases the capabilities of recurrent SNNs when trained using backpropagation through time combined with a rewiring algorithm that optimizes network architecture [15]. Another approach involves learning the membrane leak and firing threshold along with network weights, achieving high accuracy and highlighting the importance of dynamic adaptation in neuron models. By simultaneously optimizing these parameters, the network can better adjust to varying conditions and data characteristics, leading to improved performance.

Creating neuronal heterogeneity by randomizing membrane time constants drawn from a uniform distribution has been proposed, leading to improved accuracy and an increase in spikes per neuron in speech recognition tasks [16]. This research underscores the benefits of incorporating heterogeneity into neuron models. All these models adjust individual neuron parameters to bring heterogeneity, ensuring that not all neurons spike in the same way. Some neurons may spike faster, while others may have different firing patterns. In this study, more complex behavior is simulated using the Izhikevich model. Rather than adjusting the time constants, parameters a , b , c , and d are adjusted. These parameters control various aspects of neuron dynamics, such as recovery, sensitivity, and after-spike reset, allowing for a wide range of spiking behaviors.

Despite the lack of a commonly accepted theory on how real neurons encode information with spikes, spike count remains a prevalent method for feature extraction from neural activity. This method has been utilized to decode an LSM with astrocyte-modulated plasticity, achieving high accuracy on MNIST, N-MNIST, and Fashion-MNIST datasets [17]. Spike count has also been employed for speech recognition using extended liquid state machines and in liquid state machines with evolutionary optimization [16]. These studies illustrate the effectiveness of spike count as a feature extraction method in various applications.

Building on this extensive body of work, the proposed research aims to experiment with different neuron types defined by the Izhikevich model and various decoding methods to enhance classification performance. By tuning the reservoir to incorporate different types of neurons, the impact of neuron heterogeneity and dynamic adaptation will be explored to enhance the predictive capabilities of reservoir spiking neural networks.

3 Methods

We have data of the form $\{(X_i, y_i)\}_{i=1}^n = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}$. Here $X_i \in \mathbf{R}^{T \times d}$ is the longitudinal data matrix for the i -th sample and y_i is its corresponding label. First, X is converted into spike trains of the form $S \in \{0, 1\}^{T \times d}$ where each element $s_{ijt} \in \{0, 1\}$ indicates whether a spike occurred (1) or not (0) for the j -th feature at time t . We convert our data into spike trains using a delta threshold-based spike encoder function where a spike (1) is generated if $x_{jt} - x_{jt-1}$ exceeds a certain threshold value θ ; otherwise, no spike is generated (0). In general, the evolution of states of NeuCube over time is governed by $r_{t+1} = g(r_t, s_t)$. For classification, we extract the state vectors from states evolved over time by a sampling function $f : \mathbf{R}^{T \times N} \rightarrow \mathbf{O}^c$ where \mathbf{O} is a c -dimensional output vector and N is the number of neurons in the reservoir. f can be defined using various methods based on rate and temporal coding which will be discussed in the following sections.

In this paper, we generate a network of 1000 spiking neurons based on neurons positioned in a 3D cube with small world connections. The probability of a connection between two neurons is given by:

$$P_{ab} = \begin{cases} C \cdot e^{-\frac{d_{ab}}{\lambda^2}} & \text{if } d_{ab} \leq d_{\text{thresh}} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Equation (1) ensures that neurons closer to each other have a higher connection probability. This is further modulated by two parameters: C , which controls the maximum connection probability, and λ , which adjusts the extent of the connectivity radius in the network. Connections weights for reservoir neurons are stored in a $N \times N$ matrix W_j which are formed based on Equation (1). The inhibitory split was set to 0.2, meaning neurons were 80% excitatory and 20% inhibitory. Inhibitory neurons had negative synaptic weights which help regulate the spiking activity in the reservoir. Our methodology begins by encoding data into spike trains which are then fed into a reservoir of Spiking Neural Networks (SNN) comprising Izhikevich neurons. Initially, we populate the reservoir with neurons of the same type. Subsequent experiments aim to optimize the reservoir by introducing a mix of neuron types. After simulating the reservoir over the input duration, we perform feature extraction by sampling state vectors using various methods detailed in subsequent sections. Finally, we employ an SVM classifier to learn and map these state vectors to labels for classification purposes Fig. 1.

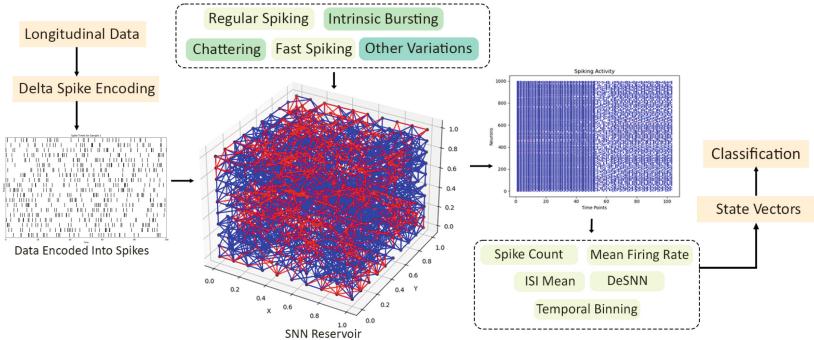


Fig. 1. Experimental setup of this study. First the data is encoded into spikes. Then, a reservoir is initialised and simulated using different Izhikevich neuron parameters. Finally, state vectors (features) are extracted using various sampling methods for further classification.

3.1 Spiking Neuron Model

The discrete-time version of the Izhikevich neuron can be derived by the forward Euler method as follows:

$$\begin{aligned}
v_{t+1} &= v_t + 0.04v_t^2 + 5v_t + 140 - u_t + I[t] \\
u_{t+1} &= u_t + a(bv_t - u_t) \\
\text{if } v_t &\geq 30 \text{ then} \\
v_{t+1} &= c \\
u_{t+1} &= u_t + d
\end{aligned} \tag{2}$$

where v is the membrane potential, u is the membrane recovery variable, and I is the input current. a, b, c, d are the parameters which control various aspects of the dynamics of the neurons. In the above derivation, we have assumed a step size of 1. This step size is relatively large from a numerical point of view because it may not capture very fine details of the neuron's behavior; however, the focus is on capturing the overall spiking behavior of neurons rather than precise sub-threshold dynamics.

The input current into each neuron is given by:

$$I_t = \sum_{i=1}^d W_i \theta_i[t] + \sum_{j=1}^N W_j \theta_j[t] \tag{3}$$

Here W_i represents connection weights from input neurons to reservoir neurons and W_j represents connections within the reservoir. θ_i and θ_j represent whether a spike occurred at neuron i and j .

Table 1. Parameters a, b, c , and d for simulating behavior of different types of known cortical neurons using the Izhikevich model.

Type	a	b	c	d
Regular Spiking	0.02	0.2	-65	8
Intrinsic Bursting	0.02	0.2	-55	4
Chattering	0.02	0.2	-50	2
Fast Spiking	0.1	0.2	-65	2

The Izhikevich neuron can model many different types of neuronal dynamics. In this paper, we limit our focus to intrinsic bursting, chattering, regular spiking, and fast spiking. Table 1 gives the a, b, c, d parameters for simulating these neurons. a controls how quickly the recovery variable μ recovers after a spike, b controls the sensitivity of μ to the changes in membrane potential v , c controls the after spike reset value of v and d effects the after spike reset value of the μ [4]. By adjusting these parameters, we can model different types of firing patterns in neurons. Regular spiking (RS) neurons exhibit a consistent firing pattern commonly seen in normal cortical activity. Fast spiking (FS) neurons fire at high frequencies with little adaptation, typically functioning as inhibitory neurons that regulate neural circuits. Intrinsically bursting (IB) neurons generate a burst of spikes at the onset of a strong input current, before transitioning

into a regular spiking mode. Chattering (CH) neurons produce repeated bursts of high-frequency spikes [18].

Table 2. Adjusted a,b,c,d parameters to create additional variants of RS,CH,IB,FS neurons to stimulate different types of spiking activity in reservoir

Type	a	b	c	d
VFS	1	0.4	-65	2
IBFS	0.02	1	-65	8
IBVFS	0.02	1	-65	2
MCH	0.02	0.2	-55	2
FCH	0.05	0.2	-50	4

Based on the original parameters defining RS, CH, IB, and FS neurons, we refined these settings to create additional variants: Very Fast Spiking (VFS), Intrinsic Bursting Fast Spiking (IBFS), Intrinsic Bursting Very Fast Spiking (IBVFS), Chattering (CH), Medium Chattering (MCH), and Fast Chattering (FCH). Figure 2 illustrates how adjustments detailed in Table 2 influence the spiking patterns when a constant 10 mV current is applied. These adjustments are intended to enhance the intensity, frequency, bursting, and chattering activities. It's important to note that these variations aim to simply stimulate diverse types of spiking activity within the reservoir, rather than precisely replicate real neuronal behaviors.

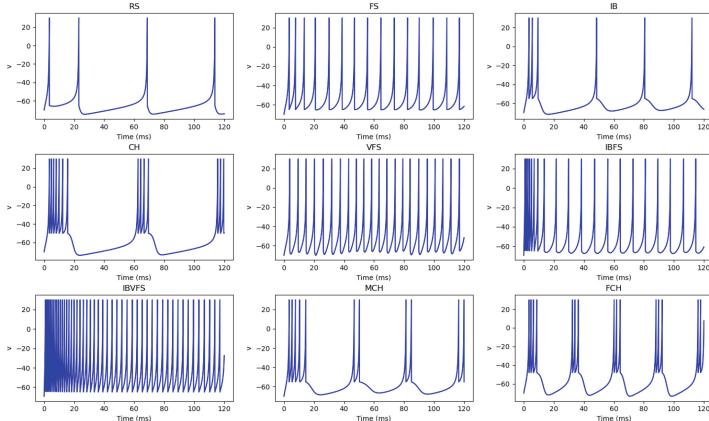


Fig. 2. Spiking activity of neurons with different a , b , c , d parameters when a constant 10 mV current is applied.

3.2 Sampling Methods

We use various methods from rate coding, temporal coding, and hybrid approaches to decode spiking activity into state vectors. Rate coding methods focus on the overall firing rates of neurons providing a high-level overview of neuronal activity. Temporal coding methods, on the other hand, emphasize the exact timing of spikes [19]. Hybrid approaches aim to use the advantages of both strategies. The methods we have used in this study include:

Spike Count: Spike count is a straightforward method of quantifying neuronal activity by counting the total number of spikes generated by each neuron over the simulation period. This method provides a “high-level” snapshot of neuronal activity and is relatively simple to implement, involving a counter that increments with each spike detected.

Mean Firing Rate: The mean firing rate is an averaged measure of neuronal activity over time. It is calculated by dividing the total number of spikes by the duration of the observation period. This method smooths out short-term fluctuations and provides a more stable representation of neuronal firing, which can be particularly useful for reducing noise in the data.

Temporal Binning: Temporal binning involves dividing the observation period into smaller time bins and counting the number of spikes that occur within each bin. This method captures the temporal structure of neuronal activity and can reveal patterns and correlations over time. Implementation requires selecting an appropriate bin size to balance temporal resolution with spiking activity complexity.

Inter-Spike Interval (ISI) Mean: The mean inter-spike interval (ISI) is the average time between consecutive spikes for each neuron. By measuring these intervals, this method can provide insights into the regularity and temporal dynamics of neuronal firing. Calculating the ISI mean involves recording the time of each spike, computing the intervals between successive spikes, and then averaging these intervals.

DeSNN: The DeSNN algorithm first assigns weights based on the timing of the initial spike in each neuron, then it adjusts them according to overall spike activity and inactivity. Parameters such as ‘alpha’ and ‘mod’ govern the initial weights and the influence of spike order, while ‘drift up’ and ‘drift down’ modulate the vectors based on spike presence. This method captures both rate coding and temporal coding features [20].

4 Results

4.1 Dataset

For evaluation, we utilize the LYRIKS Longitudinal Youth at Risk Study (LYRIKS) dataset, which provides RNA sequencing genetic data collected from Ultra-High Risk (UHR) and control groups over a period of two years [21]. The dataset includes three time points: baseline, 12 months, and 24 months. In total, there are 115 samples, comprising 64 controls and 51 UHR subjects. From the

original 21,810 features (genes), we select the top 20 using the signal-to-noise ratio to focus on the most informative genetic markers. To ensure sufficient spiking activity in the neural reservoir, we interpolate the data from the three time points to 104 time points. The primary task is to classify the control group versus the UHR group. The classification process involves using the spiking activity generated by the reservoir, extracting feature vectors by various sampling methods, and then classifying using support vector machines (SVM).

4.2 Experimental Setup

For all experiments, we employ repeated k-fold cross-validation (CV) with 5 splits and 2 repeats to ensure robust evaluation of the model performance. For each fold, the input data is divided into training and testing sets. The spike responses from the Izhikevich neural reservoir are sampled using various methods, resulting in state vectors that serve as features for classification. For Readout, a SVM Classifier, with hyperparameters optimized through a grid search over a parameter grid consisting of different values for the penalty parameter C (ranging from 2 to 8), kernel types (rbf, linear, and polynomial), and gamma (0.1, 0.01, 0.001). The grid search uses a 10-fold CV on the training set to identify the best model based on Matthews correlation coefficient (MCC) as the primary scoring metric, while accuracy is also monitored Table 3. The optimized SVM model is then evaluated on the test set for each fold, and the performance metrics (accuracy and MCC) are recorded. This process ensures a thorough and unbiased assessment of the classification performance across different configurations.

4.3 Experimental Results

The results indicate that among the different samplers, temporal binning, particularly with 10 bins, provided the highest accuracy across all neuron types. Highest accuracy was achieved with chattering neurons and temporal binning (10

Table 3. Comparison of classification accuracy (and standard errors) with different types of samplers and neurons. Values in bold show the highest accuracy for each sampler.

Neuron	DeSNN	ISI Mean	MFR	Spike Count	TB 10 Bins	TB 20 Bins	TB 30 Bins
CH	0.68 ± 0.03	0.58 ± 0.04	0.63 ± 0.03	0.66 ± 0.03	0.83 ± 0.02	0.80 ± 0.03	0.80 ± 0.01
FCH	0.62 ± 0.02	0.67 ± 0.03	0.59 ± 0.03	0.62 ± 0.04	0.80 ± 0.02	0.79 ± 0.03	0.75 ± 0.02
MCH	0.65 ± 0.04	0.58 ± 0.02	0.59 ± 0.03	0.62 ± 0.03	0.80 ± 0.02	0.78 ± 0.02	0.78 ± 0.03
IB	0.70 ± 0.02	0.66 ± 0.03	0.64 ± 0.03	0.63 ± 0.03	0.80 ± 0.01	0.77 ± 0.03	0.76 ± 0.02
IBFS	0.77 ± 0.02	0.77 ± 0.04	0.76 ± 0.03	0.73 ± 0.03	0.81 ± 0.02	0.76 ± 0.03	0.79 ± 0.02
IBVFS	0.78 ± 0.03	0.81 ± 0.03	0.79 ± 0.02	0.80 ± 0.03	0.81 ± 0.02	0.81 ± 0.02	0.78 ± 0.02
RS	0.68 ± 0.03	0.63 ± 0.03	0.67 ± 0.03	0.60 ± 0.03	0.78 ± 0.03	0.77 ± 0.03	0.74 ± 0.03
FS	0.61 ± 0.02	0.49 ± 0.03	0.59 ± 0.03	0.58 ± 0.04	0.77 ± 0.03	0.76 ± 0.02	0.76 ± 0.03
VFS	0.71 ± 0.02	0.65 ± 0.03	0.76 ± 0.03	0.71 ± 0.02	0.82 ± 0.03	0.77 ± 0.03	0.79 ± 0.02

bins). As chattering neurons (CH, FCH, and MCH) fire in bursts or sequences, their activity was better captured in short time intervals (bins) as their fine-grained temporal information is preserved. The spike count method, which involves summing the number of spikes over a longer period, tended to smooth out the temporal dynamics, potentially losing this fine-grained temporal information. Consequently, spike count and other rate coding methods resulted in lower accuracy compared to temporal binning.

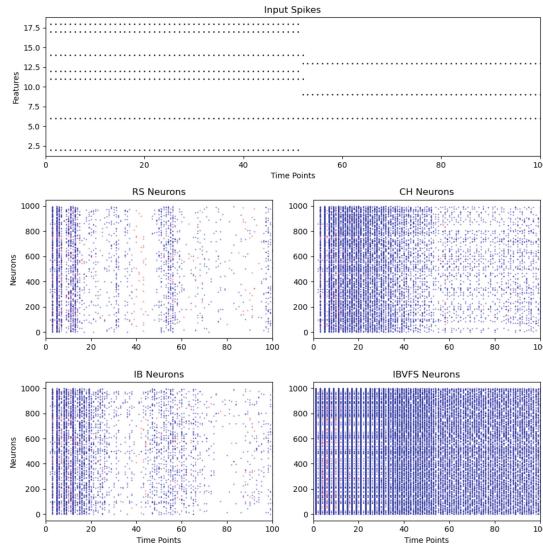


Fig. 3. Input spikes from a randomly selected sample and the resulting spiking activity with different neurons within the reservoir

Intrinsic bursting neurons (IB, IBFS, and IBVFS) demonstrated the highest accuracy across most samplers. This can be attributed to their characteristic behavior of producing bursts of spikes at the beginning of a strong pulse of current. This bursting activity allows for quicker network penetration of spiking activity, making them highly effective for transmitting information within the network. For example, the IBVFS neuron type showed consistent high performance with ISI Mean (0.78 ± 0.03), MFR (0.81 ± 0.03), and maintained high accuracy across different temporal binning approaches (10 bins: 0.80 ± 0.03 , 20 bins: 0.81 ± 0.02 , 30 bins: 0.78 ± 0.02). The results also indicate that increasing the number of bins (20 and 30 bins) did not significantly improve the accuracy beyond what was achieved with 10 bins. In some cases, higher bin counts slightly reduced the accuracy, suggesting that overly fine temporal resolution might introduce noise or redundancy, detracting from the overall performance.

Figure 3 shows that when input spikes are limited, IBVFS neurons are still able generate high level of spiking activity. Similarly CH neurons showed strong spiking acitvity up until 50 timepoints, however, as input spikes decreased the

overall spiking activity was also affected. IB neurons initially showed bursting behaviour but eventually switch to tonic bursting as regular spiking. Despite limited spiking activity due to RS and IB, temporal binning is still able to achieve competitive accuracy when compared to other methods.

These findings underscore the importance of considering the characteristics of different neuron types and their spiking patterns when selecting the appropriate sampling method. Temporal binning, particularly with 10 bins, emerges as a robust method for capturing the dynamics of neurons with burst-like spiking patterns, such as chattering and intrinsic bursting neurons.

5 Conclusion and Future Works

In this experimental study, we explored the integration of Izhikevich neurons into the NeuCube framework using the NeuCube-Py library. We compared the effects of various sampling methods (rate, temporal and hybrid coding) for feature extraction, on the classification accuracy of control and UHR samples using the LYRIKS RNA sequencing dataset. Our experiments demonstrated that temporal binning, particularly with 10 bins, consistently provided the highest accuracy across all neuron types. This method effectively captures the fine-grained temporal dynamics of spiking activity, especially for neurons exhibiting burst-like spiking patterns such as chattering and intrinsic bursting neurons. Our findings indicate that while increasing the number of bins can capture more temporal details, it does not necessarily improve accuracy and may introduce noise or redundancy. Additionally, intrinsic bursting neurons (IB, IBFS, and IBVFS) showed the highest accuracy across most sampling methods, highlighting their effectiveness in transmitting information within the network even when input spiking activity is limited. Chattering neurons (CH, FCH, and MCH) also performed well with temporal binning, which effectively captured their burst-like spiking patterns. In contrast, rate coding methods such as spike count and mean firing rate generally resulted in lower accuracy, likely due to their tendency to smooth out important temporal dynamics. Overall, this study underscores the importance of selecting appropriate sampling methods based on the unique spiking patterns of different neuron types to optimize classification performance within the NeuCube framework. Future research should focus on introducing lifelong learning [22] and integrating multimodal datasets within this framework to enable more comprehensive modelling of real-world problems.

Acknowledgments. This research is supported by the MBIE Catalyst: Strategic New Zealand-Singapore Data Science Research Programme Funding and the National Research Foundation, Singapore under its Industry Alignment Fund - Pre-positioning (IAF-PP) Funding Initiative. The LYRIKS study was supported by the National Research Foundation Singapore under the National Medical Research Council Translational and Clinical Research Flagship Programme (NMRC/TCR/003/2008). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Maass, W.: Chapter 1 liquid state machines : Motivation, theory, and applications (2010)
2. Kasabov, N.K.: NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Netw.* **52**, 62–76 (2014)
3. Gerstner, W., Kistler, W.M., Naud, R., Paninski, L.: *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press (2014)
4. Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **14**(6), 1569–1572 (2003)
5. Wang, Q., Li, P.: D-LSM: deep liquid state machine with unsupervised recurrent reservoir tuning. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2652–2657 (2016)
6. Xuhu, Yu., Wan, Z., Shi, Z., Wang, L.: Lipreading using liquid state machine with STDP-tuning. *Appl. Sci.* **12**(20), 10484 (2022)
7. Liu, C., Wang, H., Liu, N., Yuan, Z.: Optimizing the neural structure and hyper-parameters of liquid state machines based on evolutionary membrane algorithm. *Mathematics* **10**(11), 1844 (2022)
8. Lisman, J.: Bursts as a unit of neural information: making unreliable synapses reliable. *Trends Neurosci.* **20**(1), 38–43 (1997)
9. Saeedinia, S.A., Jahed-Motlagh, M.R., Tafakhori, A., Kasabov, N.: Design of MRI structured spiking neural networks and learning algorithms for personalized modelling, analysis, and prediction of EEG signals. *Sci. Rep.* **11**(1) (2021)
10. Saeedinia, S.A., Jahed-Motlagh, M.R., Tafakhori, A., Kasabov, N.K.: Diagnostic biomarker discovery from brain EEG data using LSTM, reservoir-SNN, and NeuCube methods in a pilot study comparing epilepsy and migraine. *Sci. Rep.* **14**(1) (2024)
11. Manna, D.L., Vicente-Sola, A., Kirkland, P., Bihl, T.J., Di Caterina, G.: Simple and complex spiking neurons: perspectives and analysis in a simple STDP scenario. *Neuromorphic Comput. Eng.* **2**(4), 044009 (2022)
12. Schliebs, S., Mohemmed, A., Kasabov, N.: Are probabilistic spiking neural networks suitable for reservoir computing? In: The 2011 International Joint Conference on Neural Networks. IEEE (2011)
13. Li, X., Chen, Q., Xue, F.: Bursting dynamics remarkably improve the performance of neural networks on liquid computing. *Cogn. Neurodyn.* **10**(5), 415–421 (2016). <https://doi.org/10.1007/s11571-016-9387-z>
14. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2641–2651 (2020)
15. Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., Maass, W.: Long short-term memory and learning-to-learn in networks of spiking neurons. *Adv. Neural Inf. Process. Syst.* **31** (2018)
16. Deckers, L., Tsang, I.J., Van Leeuwijk, W., Latré, S.: Extended liquid state machines for speech recognition. *Front. Neurosci.* **16** (2022)

17. Ivanov, V., Michmizos, K.: Increasing liquid state machine performance with edge-of-chaos dynamics organized by astrocyte-modulated plasticity. *Adv. Neural. Inf. Process. Syst.* **34**, 25703–25719 (2021)
18. Izhikevich, E.M.: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press (2006)
19. Eshraghian, J.K., et al.: Training spiking neural networks using lessons from deep learning. *Proc. IEEE* **111**, 1016–1054 (2021)
20. Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G.: Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Netw.* **41**, 188–201 (2013)
21. Lee, J., et al.: The longitudinal youth at risk study (lyriks) - an asian uhr perspective. *Schizophr. Res.* **151**(1-3), 279–283 (2013)
22. Koprinkova-Hristova, P., Penkov, D., Nedelcheva, S., Yordanov, S., Kasabov, N.: On-line learning, classification and interpretation of brain signals using 3D SNN and ESN. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2023)



Calming the Mind: Spiking Neural Networks Reveal How Havening Touch to Reduce Persistent Distress Attenuates Left Temporal Electroencephalographic Connectivity

Alexander Sumich¹(✉) , Zohreh Dotorjeh^{2,3,4} , Nadja Heym¹ , Aroha Scott⁴ , Kirsty Hunter⁵ , Tony Burgess⁶ , Julie French⁶ , Mustafa Sarkar⁵ , Maryam Dotorjeh^{2,3} , and Nicola Kasabov^{2,3,7}

¹ NTU Psychology, Nottingham Trent University, Nottingham NG1 4FQ, UK
alexander.sumich@ntu.ac.uk

² The Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, Auckland 1010, New Zealand

³ Information Technology and Software Engineering Department, Auckland University of Technology, Auckland 1010, New Zealand

⁴ School of Population Health, Faculty of Medical and Health Sciences, The University of Auckland, Auckland 1023, New Zealand

⁵ Sports Science, Nottingham Trent University, Nottingham NG1 4FQ, UK

⁶ Academy of High Achievers Ltd., Stafford, UK

⁷ Intelligent Systems Research Centre, Ulster University, Londonderry, UK

Abstract. Havening is an innovative psychosensory intervention that uses touch to facilitate recovery from psychological trauma. Whilst Havening is commonly employed by practitioners worldwide, only a few empirical studies have investigated efficacy or mechanism. The current study applies explainable machine learning methods to brain data to better understand mechanisms underpinning the role of Havening touch in recovery from trauma. Participants ($n = 27$) who had experienced an event that caused persistent psychological distress underwent a single Havening session that either did ($H+$, $n = 15$) or did not ($H-$, $n = 12$) include a touch component. Resting-state electroencephalography (EEG) data was recorded before (T1) and after (T2) the intervention. Two $H+$ subgroups were formed based on self-reported-response to the intervention. A recently developed brain-inspired machine-learning model, the Spiking Neural Network (SNN) was applied to compare groups and time points on functional connectivity. Results suggest region-specific reductions (left temporal, left frontocentral) in connection weights following Havening touch. Differentiation of $H+$ and $H-$ was more accurate at T2 than T1; and in $H+$, brain states at T2 were more accurately classified than at T1, particularly for participants who had a greater response to the intervention. Findings support the SNN in distinguishing brain states associated with response to Havening. Reduction in left temporal connectivity may reflect downregulation of anterior temporal lobe activity, downstream from the amygdala.

Keywords: Spiking Neural Networks · Machine Learning ·
Electroencephalography · Connectivity · Havening · Therapy · Touch · Distress ·
Trauma

1 Background and Introduction

Nurturing affiliative touch is fundamental for healthy psychological, physical, and relational development across the lifespan (Cekaite and Goodwin 2021; Jablonski 2021; Jakubiak and Feeney 2017). It has been associated with a sense of esteem, value and group cohesion, mitigating negative experiences of loneliness and fostering prosocial behaviour (Heatley Tejada et al. 2020). Nurturing touch enhances both psychological (Müller-Oerlinghausen and Eggart 2021) and physical healing (e.g., wound repair; de Souza et al. 2017), and is involved in the development and maintenance of several biological systems, including the oxytocin and stress response systems, immune function, and brain chemistry (Jablonski 2021).

Havening is a form of psychosensory therapy that harnesses the power of touch to reduce emotional distress and promote wellbeing (Ruden 2018). Havening is used globally as a therapeutic intervention to facilitate recovery from psychological trauma. However, only a few studies have empirically evaluated its therapeutic efficacy or implicated mechanism (Thandi et al. 2015; Cizmic et al. 2018; Hodgson et al. 2020; Sumich et al. 2022). Recently, for the first time, Sumich et al (2022) empirically demonstrated that the use of Havening touch in a single 20-min therapeutic session was associated with a more rapid and extensive reduction in distress, and that this was paralleled by a reduction in electroencephalographic (EEG) gamma activity (20–50 Hz). This is in line with the idea that Havening helps to downregulate brain regions implicated in the experience of negative emotions (e.g., fear, anxiety, anger), and is relevant to psychopathologies associated with limbic hyperactivity.

Resting-state functional hyperconnectivity, particularly for temporal regions and those involved in the dorsal and ventral (salience) attention networks, is reported in psychopathologies associated with upregulated limbic activity (e.g., schizophrenia Sakakibara et al. 2024; post-traumatic stress disorders; PTSD Li et al. 2022) and in individuals at risk of developing psychosis (Yoon et al. 2015). This may reflect a state of hyperarousal/vigilance, particularly under conditions of threat (Li et al. 2022; Etkin et al. 2019). Evidence-based interventions that help regulate limbic hyperconnectivity in the context of trauma are needed.

Machine learning techniques are increasingly applied to brain data. A challenge here is in integrating complex, dynamic patterns seen in spatial and temporal data, such as comprises EEG. Often, spatial and temporal information is analysed separately rather than within a unified computational model. The current research applies a computational approach based on spiking neural networks (SNNs), a promising advancement in artificial neural networks (Kasabov 2014). SNNs offer a neurobiologically plausible architecture that integrates both spatial and temporal aspects of data into a single model.

As an explainable method, applicable to spatio-temporal data, SNN have been useful in understanding connectivity and classifying brain states. For example, SNN methods

have been used to generate connection weights, and accurately classify brain states in relation to depressive symptoms and responsiveness to psychological interventions (e.g., mindfulness; Doborjeh et al. 2019; 2020). Other work highlights the application of SNN to cross-sectional electroencephalography (EEG) data to differentiate healthy ageing from early stages of dementia (Crook-Rumsey et al. 2023), and to longitudinal measures of brain structure to accurately predict the onset of dementia 2–4 years in advance (Doborjeh et al. 2021).

The current study applies SNN methods to EEG data recorded immediately prior to, and following, a single Havening session. SNN techniques are used to generate connection weights and classify brain states as a function of *Time*, experimental *Condition* and *Response* to intervention. We predicted greater downregulation of temporal activity following Havening Touch, reflected in a reduction of connection weights.

2 Methods

The study was conducted at Nottingham-Trent University (NTU) and was given a positive opinion by the NTU non-invasive ethics committee. Participants ($n = 24$, 21 female and 3 male, $n = 21$ white, age range 18–47 years mean = 25.21, SD = 7.81), were predominantly staff and students, who had experienced a moderately distressing event (self-rated Subjective units of distress = 5–8 on 0–10 scale) that had persisted for at least a month. Participants were pseudo-randomised into full intervention that included a touch component (H+; $n = 15$) or a partial intervention, where no Havening touch components were applied (H-; $n = 12$). Figure 1 shows study protocol.

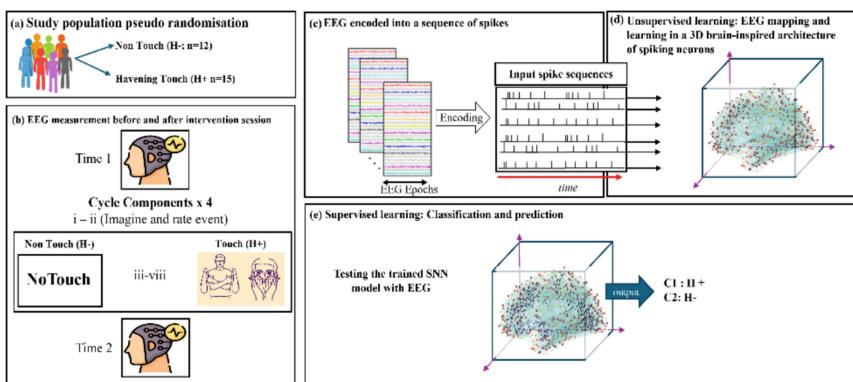


Fig. 1. Electroencephalogram (EEG) data collection and computational spiking neural network modelling. (a) Pseudorandomization into non-touch (H-) and touch (H+) groups; (b) EEG data collection before and after the training; (c) Illustration of the designed methodology, containing EEG encoding into spike sequences; (d) Computational modelling of data into a 3D space of artificial neurons; (e) Pattern classification and prediction.

In both conditions, the protocol comprised a series of components that required participants to: i) close their eyes for 15 s and think about the distressing event/memory;

ii) rate their subjective feelings of distress in relation to the event/memory; iii) name animals beginning with specific letters; iv) imagine photos of happy images; v) hum a childhood song; vi) count their steps as they imagine walking through a beautiful landscape; vii) count the racquet strokes as they imagine watching a tennis match; and viii). State the months of the year in backwards order from December to January. There were 4 cycles of these 8 components in an intervention session. During the H+ condition only, the therapist incorporated Havening Touch during components iii–viii. SUDs were assessed at baseline and as component ii in each of the repeated cycles.

Resting-state EEG (64 channels, Biosemi; sampling rate = 2048 Hz) was recorded for two minutes (eyes closed) prior to (T1) and following (T2) the therapeutic session (which lasted approximately 20 min). Signal processing was performed using Curry 7.12 software (with bug fixes). For further details on Havening and EEG protocols see Sumich et al. (2022). Following cleaning EEG data were downsampled to 256 Hz.

2.1 EEG Modelling and Analysis Using SNN Architecture

NeuCube (Kasabov 2014) was used to apply SNN to EEG data to generate connection weights and classify brain states as a function of *Time* (T1, T2), *Condition* (H–, H+) and *Response* to intervention (H+R–, H+R+). The criterion for response was change in Subject Units of distress (SUDs) greater than the mean change of 3 for the H+ group. In the H+ group, 8 participants were responders (change in SUD mean = 4.75). Seven participants were nonresponders (change in SUD < 3; change in SUD mean = 1.38). The EEG data was pre-processed and encoded into spike trains, retaining only significant changes in activity, which were then trained using the unsupervised learning algorithm which learns from the spatiotemporal relationships from input spikes, Spike Time-Dependent Plasticity, in NeuCube software (Kasabov 2014). Talairach atlas-based 3-dimensional space of spiking artificial neurons was constructed using NeuCube to create a brain template. SNN models were trained on EEG data as a function of *Time*, *Condition* and *Response*. Traditional statistical analysis, Analysis of Variance (ANOVA), tested for the effects of *Time* and *Condition* on connection weights. An output layer classifier was trained in a supervised mode to learn the association between the trained SNN connectivity and the class label information. The dynamic evolving Spiking Neural Network (deSNN) classifier was used to perform the classification task to compare classes based on *Time* for each *Condition*, and classes based on *Condition* for each timepoint. Similar analysis was performed comparing Responders (H+R+) to Nonresponders (H+R–).

3 Results

3.1 Data Analysis

The research outcomes are organised in a three-phase analysis as follows:

- (1) Pattern Visualization: EEG data were modelled using the SNN architecture and are present as functional connection weights, as a function of Condition, Time and Response (see Fig. 2.).

- (2) Inferential Statistics were used to test differences in connection weights as a function Condition and Time.
- (3) Pattern classification: functional connection weight maps were used to classify brain states as a function of Condition, Time and Response.

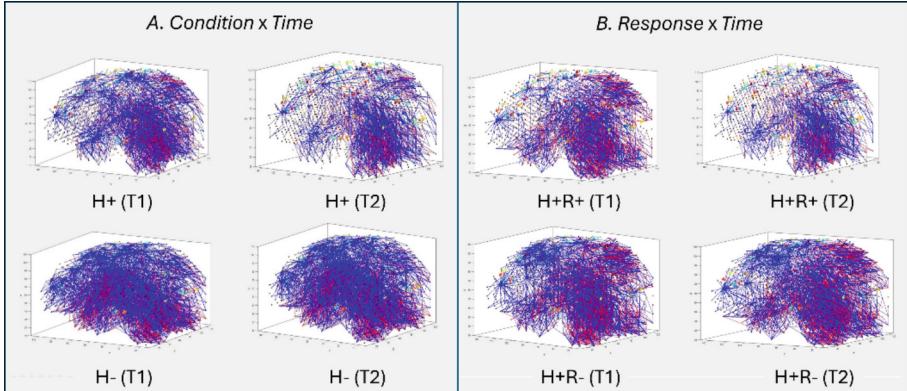


Fig. 2. SNN connectivity trained on EEG samples, as a function of *Condition* and *Time*.

3.2 Pattern Visualization: EEG Data Modelling Using SNN

Figure 1 shows spatio-temporal connections in the SNN models as a function of *Time* (T1, T2), *Condition* (H-, H+) and *Response* (H+R+, H+R-). Grand average connection weights at T1 were: H- = 1.53, H+ = 1.49, H+ R- = 1.43, H+R+ = 1.54; and at T2 were H- = 1.52, H+ = 1.45, H+R- = 1.42, H+R+ = 1.49. In panel A, a reduction in connection weights is apparent over the temporal cortex in H+ at T2, whilst little change is seen for H-. Similarly, responders show greater reduction in temporal cortex connectivity than nonresponders. In order to perform inferential statistics, connection weight data were grouped across five regions for each hemisphere: frontal (Fp1/2, AF7/8, AF3/4, F1/2, F3/4, F5/6), frontocentral (FC5/6, FC3/4, FC1/2, C1/2, C3/4, C5/6), temporal (F7/8, FT7/8, T7/8, TP7/8), centroparietal (CP7/6, CP3/4, CP1/2, P1/2, P3/4, P5/6, P7/8, P9/10), and occipitoparietal (PO7/8, PO3/4, O1/2; please see Fig. 3.).

3.3 Inferential Statistics

Repeated-measures ANOVA tested for differences in connection weights as a function of *Hemisphere* (left, right), *Site* (frontal, frontocentral, temporal, centroparietal and occipitoparietal), *Time* (T1, T2) and *Condition* (H+, H-).

There was a significant effect of *Hemisphere* [$F = 471.57$, $df = 1, 23$, $p < 0.001$, $p\eta^2 = 0.95$; $R > L$] and *Site* [$F = 107.08$, $df = 1.75, 40.18$, $p < 0.001$, $p\eta^2 = 0.82$; Frontocentral < Frontal < Temporal = Centroparietal < Occipitoparietal], as well as a *Hemisphere***Site* interaction ($F = 30.70$, $df = 2.06, 47.38$, $p < 0.001$, $p\eta^2 = 0.57$). The

*Hemisphere***Site* interaction was due to higher scores at centroparietal than temporal sites in the left ($p < .001$), but not right ($p = .176$) hemisphere. Significant *Hemisphere* effects were seen as all sites: temporal [$F = 202.12$, $df = 1, 23$, $p < 0.001$, $p\eta^2 = 0.90$], frontocentral ($F = 216.40$, $df = 1, 23$, $p < 0.001$, $p\eta^2 = 0.90$), frontal [$F = 304.49$, $df = 1, 23$, $p < 0.001$, $p\eta^2 = 0.93$], centroparietal [$F = 131.88$, $df = 1, 23$, $p < 0.001$, $p\eta^2 = 0.85$], and occipitoparietal [$F = 7.14$, $df = 1, 23$, $p = 0.014$, $p\eta^2 = 0.24$].

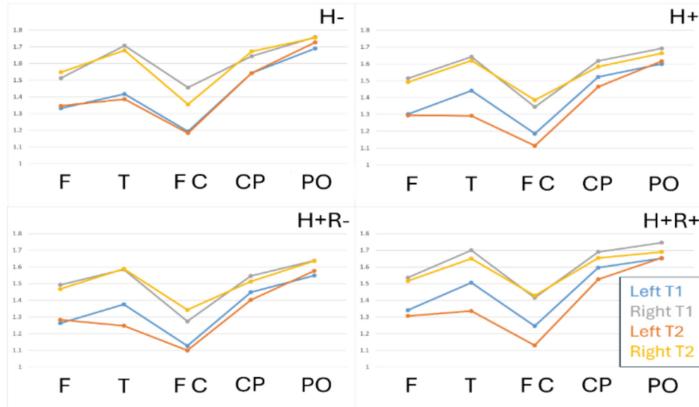


Fig. 3. Mean SNN connection weights as a function of Condition, Time Point and Scalp region (F = Frontal, T = Temporal, FC = Frontocentral, CP = Centroparietal, PO Occipitoparietal).

There were also significant *Time***Site* ($F = 3.65$, $df = 2.69, 61.77$, $p = 0.021$, $p\eta^2 = 0.137$), *Hemisphere***Time***Site* ($F = 3.66$, $df = 2.89, 66.56$, $p = 0.018$, $p\eta^2 = 0.14$) and *Hemisphere***Time***Site***Condition* ($F = 6.43$, $df = 2.894$, $p < 0.001$, $p\eta^2 = 0.22$) interactions. These were partially driven by higher values at T1 than T2 at left temporal sites in H +, but not H- (significant *Hemisphere***Time***Condition* interaction at temporal sites; $F = 4.96$, $df = 1, 23$, $p = .036$, $p\eta^2 = 0.17$). At temporal sites in H +, there was a significant *Time***Hemisphere* interaction ($F = 12.57$, $df = 1, 13$, $p = 0.004$, $p\eta^2 = 0.49$), due to a significant effect of *Time* in the left hemisphere ($F = 13.28$, $df = 1, 13$, $p = 0.003$, $p\eta^2 = 0.51$), but not in the right hemisphere ($F = .348$, $df = 1, 13$, $p = 0.565$, $p\eta^2 = 0.026$). There was no effect of *Time* at either temporal sites in H-. In addition, there was a significant *Hemisphere***Time***Condition* interaction at frontocentral sites ($F = 33.19$, $df = 1, 23$, $p < .001$, $p\eta^2 = 0.59$). This was partly due to an effect of *Time* in the left hemisphere in H+ ($T1 > T2$; $F = 6.92$, $df = 1, 13$, $p = .021$, $p\eta^2 = 0.35$), but not in H- ($F = .073$, $df = 1, 13$, $p = .79$, $p\eta^2 = 0.007$). There was no effect of *Time* in either H+ or H- at right frontocentral sites.

3.4 Pattern Classification of EEG Data Measuring Brain States Before and After Havening in the Designed SNN Model

Table 1 shows results of output classifiers trained to classify A) T1 vs T2 in H+; B) T1 vs T2 in H-; C) H- vs H+ at T1; D) H- vs H+ at T2; E) H+R- vs H+R+ and T1; and

Table 1. Confusion tables showing classification with per class and total accuracy.

	Predicted		Accuracy	
	H+(T1)	H+(T2)	Per Class	Total
A. Real				
H+(T1) (n = 12)	4	2	71.43	76.63
H+(T2) (n = 12)	1	5	81.82	
	H- (T1)	H- (T2)		
B. Real				
H- (T1) (n = 11)	4	2	71.43	71.43
H- (T2) (n = 11)	2	4	71.43	
	H- (T1)	H+ (T1)		
C. Real				
H- (T1) (n = 6)	3	3	0.45	0.56
H+ (T1) (n = 6)	2	4	0.67	
	H- (T2)	H+ (T2)		
D. Real				
H- (T2) (n = 6)	4	2	0.67	0.72
H+ (T2) (n = 6)	1	5	0.77	
	H+R+ (T1)	H+R+ (T2)		
E. Real				
H+R+ (T1) (n = 7)	2	5	0.39	0.63
H+R+ (T2) (n = 7)	1	6	0.87	
	H+ R- (T1)	H+R- (T2)		
F. Real				
H+R- (T1) (n = 7)	4	3	0.57	0.57
H+R- (T2) (n = 7)	3	4	0.57	

F) H+R- vs H+R+ at T2. We extracted 30 s (10 s from beginning, middle and end) of data from each participant at each time point (7680 samples 30×256 hz). Classification was based on random 50/50 training/testing. H+ showed higher accuracy at T2 than T1. However, T1 and T2 accuracy for H- were identical. Similar findings were seen for responders vs nonresponders. At T1, classification of H+ vs H- was low (56%), whilst this increased for T2 (72%).

4 Discussion

The current study is the first to investigate change in EEG connectivity following a single Havening therapy session for a persistently distressing event. Resting-state EEG data were collected prior to and following the intervention in two conditions: those who received Havening touch and those who did not. Analysis methods benefitted from the synthesis of innovative, brain-inspired, explainable machine learning – spiking neural networks - and inferential statistical methods. In line with the hypotheses, people who received the Havening touch showed significant reduction in connection weights over left temporal regions, compared to those in the nontouch condition. However, unexpectedly, significant reductions in connection weights with Havening touch extended to left frontocentral regions. Classification accuracy was higher for T2 than T1 in the touch condition only. Moreover, accuracy in differentiating touch vs nontouch was little over chance at T1, but increased following the intervention, supporting a divergence of brain states between conditions.

The development of the Havening technique is based on theory that it “calms” limbic hyperactivity. By using innovative interpretable machine learning methods to examine connection weights across brain regions, the current study is the first to lend empirical support to this theory. The reduction in connection weights parallels greater reduction in SUDs with Havening touch previously reported in this cohort (Sumich et al. 2022).

The amygdala, situated in the medial temporal cortex, is central to the experience of emotions that carry negative valence (e.g., anger, fear, sadness), and has strong connections with anterior lateral temporal and frontal regions. Reduction in amygdala activity has been implicated in the therapeutic effect of Havening touch, and oxytocin administration reduces amygdaloid gamma activity in rodents (Sobota et al. 2015). Whilst scalp-recorded EEG activity is unlikely to directly reflect amygdala activity, it may well represent downstream temporal and frontal regions. Future approaches should therefore further investigate the current findings using other neuroimaging methods with greater precision in assessing amygdala function, such as functional magnetic resonance imaging.

Havening is commonly used to treat psychological trauma-based problems. Hyperconnectivity has been reported in PTSD using several measures of brain function. For example, using magnetoencephalography, Dunkley et al. (2018) show hypersynchrony in thalamus, ventromedial prefrontal cortex, cingulum gyri, inferior temporal and parietal regions, under conditions of threat. However, heterogeneity in connectivity has been reported in PTSD, and hyperconnectivity may be more evident in specific illness subtypes and under threatening conditions (Etkin et al. 2019). For example, Li et al. (2022) applied unsupervised sparse K-means clustering to EEG data to identify two subtypes of patients with PTSD, one of which was associated with hyperconnectivity, particularly in ventral and dorsal attention networks.

Lower connectivity in PTSD predicts poorer treatment response to prolonged exposure psychotherapy in PTSD (Etkin et al. 2019). In contrast, however, based on the connection weights in the current study, poorer treatment responders to Havening touch appeared to have higher connection weights than responders. However, given small sample size this was not tested using inferential statistics. Nevertheless, comparison across studies raises the possibility of distinct mechanisms of action for Havening compared

to prolonged exposure psychotherapy. Also, mindfulness training (which builds awareness of experience) in people with raised depression symptoms was associated with an increase in connection weights across several regions (Doborjeh et al. 2019). Thus, future studies should use SNN to directly compare brain states that track and predict response to distinct interventions. Implications for such work would be in supporting synergistic effects of multi-modal interventions and more personalized therapies.

Whilst Li et al. (2022) highlight importance of theta, alpha and beta frequency bands, a magnetoencephalography study used graph analysis to associate left hippocampal hyperconnectivity in the high-gamma band with severity of PTSD (Dunkley et al. 2014). Reduction in EEG gamma activity has been reported following Havening (Sumich et al. 2022). Thus, future studies should investigate connection weights for specific frequency bands. Such work would also provide further understanding of the underpinning mechanisms and functional significance of the reduction in connection weights following Havening touch.

Prolonged psychological distress (trauma) is a risk factor for several psychopathologies, such as psychosis and major depressive disorders. Hyperconnectivity has been observed in individuals at ultra-high risk for psychosis (UHR), in first episode psychosis (FE), and schizophrenia (Yoon et al. 2015; Sakakibara et al. 2024). In UHR and FE, increased connectivity was noted between left planum temporal and bilateral dorsolateral prefrontal cortex (Yoon et al. 2015). Such findings mirror the topography of the current results that show the effects of Havening touch at left temporal and frontocentral regions. Thus, Havening should be investigated as an early intervention to reduce risk of illness progression in UHR and FE. In comparison, however, hyperactivity of the right temporal cortex, along with bilateral orbitofrontal cortex is reported in schizophrenia using whole-head near-infrared spectroscopy (Sakakibara et al. 2024). Higher left temporal connection weights were observed in relation to cognitive depression (Doborjeh et al., in press). However, comparison of connection weight maps to current findings suggests differences in topography (more anterior in cognitive depression), suggesting distinct underpinning mechanisms.

SNN methods are emerging as an innovative, interpretable tool for understanding and classifying brain states and have been applied to various types of spatio-temporal brain data with relatively good accuracy compared to some other methods. SNN did not differentiate groups at baseline. This makes sense as high accuracy prior to the interventions would not be expected for the H+ vs H– comparison if the groups were drawn from a similar cohort. However, higher accuracy following the distinct interventions supports the efficacy of Havening Touch in affecting brain states. Future studies should consider using SNN to investigate other types of intervention for trauma and in general for psychopathology. Direct comparison should be made with other interpretable machine learning classification methods.

Whilst the relatively small, nonclinical and predominantly female sample used in the current study restricts generalizability, as noted above, findings are in line with several clinical studies of PTSD. Nevertheless, future work should apply similar methods to clinical cohorts. Larger studies will enable investigation of heterogeneous groups observed in other studies (Etkin et al. 2019), and inferential evaluation of predictors of

response. Future work might also compare other types of touch, including other types of therapeutic touch in order to test unique properties of Havening touch.

4.1 Summary and Conclusion

The synthesis of explainable SNN, visualization and inferential statistical methods currently support the importance of the touch component of Havening in lowering activity over left temporal and frontocentral regions. Such findings contribute to understanding how Havening helps with several trauma-based psychopathologies, by lowering hyper-connectivity. Moreover, the utility of using SNN to study spatio-temporal brain dated is supported.

Acknowledgments. We thank Lorna Hatch for her help in data collection. Software modules used for implementing the proposed method and Spiking Neural Network models of EEG data can be found at: <http://www.kedri.aut.ac.nz/neucube/>.

Disclosure of Interests. Kasabov, N. has developed NeuCube software. Burgess T. and French J. are wellbeing practitioners who employ Havening methods. The authors have no other competing interests to declare that are relevant to the content of this article.

References

- Cekaite, A., Goodwin, M.H.: Touch and social interaction. *Annu. Rev. Anthropol.* **50**, 203–218 (2021)
- Cizmic, Z., Edusei, E., Anoushiravani, A.A., Zuckerman, J., Ruden, R., Schwarzkopf, R.: The effect of psychosensory therapy on short-term outcomes of total joint arthroplasty: a randomized controlled trial. *Orthopedics* **41**(6), e848–e853 (2018)
- Crook-Rumsey, M., et al.: Spatiotemporal EEG dynamics of prospective memory in ageing and mild cognitive impairment. *Cogn. Comput.* **15**, 1273–1299 (2023)
- de Souza, A.L.T., Rosa, D.P.C., Blanco, B.A., Passaglia, P., Stabile, A.M.: Effects of therapeutic touch on healing of the skin in rats. *Explore* **13**(5), 333–338 (2017)
- Doborjeh, M., et al.: Personalised predictive modelling with brain-inspired spiking neural networks of longitudinal MRI neuroimaging data and the case study of dementia. *Neural Netw.* **144**, 522–539 (2021)
- Doborjeh, Z., et al.: Spiking neural network modelling approach reveals how mindfulness training rewires the brain. *Sci. Rep.* **9**(1), 6367 (2019)
- Doborjeh, Z., et al.: Interpretability of spatiotemporal dynamics of the brain processes followed by mindfulness intervention in a brain-inspired spiking neural network architecture. *Sensors (Basel)* **20**(24), 7354 (2020)
- Doborjeh, Z., et al.: Neurocomputational modelling of multimodal data of brain connectivity. Depression, Inflammation, and Gut Microbiome Using Spiking Neural Networks. *ICONIP* (2024). (in press)
- Dunkley, B.T., Wong, S.M., Jetly, R., Wong, J.K., Taylor, M.J.: Post-traumatic stress disorder and chronic hyperconnectivity in emotional processing. *Neuroimage Clin.* **20**, 197–204 (2018)
- Dunkley, B.T., et al.: Resting-state hippocampal connectivity correlates with symptom severity in post-traumatic stress disorder. *Neuroimage Clin.* **5**, 377–384 (2014)

- Etkin, A., et al.: Using fMRI connectivity to define a treatment-resistant form of post-traumatic stress disorder. *Sci. Transl. Med.* **11**(486), eaal3236 (2019)
- Heatley Tejada, A., Dunbar, R.I.M., Montero, M.: Physical contact and loneliness: being touched reduces perceptions of loneliness. *Adapt. Hum. Behav. Physiol.* **6**, 292–306 (2020)
- Hodgson, K.L., et al.: A psychophysiological examination of the mutability of Type D personality in a therapeutic trial. *J. Psychophysiol.* **35**, 116–128 (2020)
- Jablonski, N.G.: Social and affective touch in primates and its role in the evolution of social cohesion. *Neuroscience* **464**, 117–125 (2021)
- Jakubiak, B.K., Feeney, B.C.: Affectionate touch to promote relational, psychological, and physical well-being in adulthood: a theoretical model and review of the research. *Pers. Soc. Psychol. Rev.* **21**(3), 228–252 (2017)
- Kasabov, N.K.: NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Netw.* **52**, 62–76 (2014)
- Li, Q., et al.: Resting-state EEG functional connectivity predicts post-traumatic stress disorder subtypes in veterans. *J Neural Eng.* **19**(6) (2022)
- Müller-Oerlinghausen, B., Eggart, M.: The Significance of Touch in Psychiatry. MDPI, Basel (2021)
- Ruden, R.A.: Harnessing electroceuticals to treat disorders arising from traumatic stress: theoretical considerations using a psychosensory model. *Explore* **15**(3), 222–229 (2018)
- Sakakibara, E., et al.: Abnormal resting-state hyperconnectivity in schizophrenia: a whole-head near-infrared spectroscopy study. *Schizophr. Res.* **270**, 121–128 (2024)
- Sobota, R., Mihara, T., Forrest, A., Featherstone, R.E., Siegel, S.J.: Oxytocin reduces amygdala activity, increases social interactions, and reduces anxiety-like behavior irrespective of NMDAR antagonism. *Behav. Neurosci.* **129**(4), 389–398 (2015)
- Sumich, A., et al.: The power of touch: The effects of Havening touch on subjective distress, mood, brain function, and psychological health. *Psychol. Neurosci.* **15**(4), 332–346 (2022)
- Thandi, G., Tom, D., Gould, M., McKenna, P., Greenberg, N.: Impact of a single session of havening. *Health Sci. J.* **9**(5), 1–5 (2015)
- Yoon, Y.B., et al.: Altered fronto-temporal functional connectivity in individuals at ultra-high-risk of developing psychosis. *PLoS ONE* **10**(8), e0135347 (2015)



SCA-LSTM: A Deep Learning Approach to Golf Swing Analysis and Performance Enhancement

Chengwei Feng¹(✉) , Boris Bačić^{1,2,3} , and Weihua Li¹

¹ Engineering, Computer and Mathematical Sciences (ECMS), Auckland University of Technology, Auckland 1010, New Zealand
chengwei.feng@autuni.ac.nz

² Institute of Biomedical Technologies (IBTec), Auckland University of Technology, Auckland 1010, New Zealand

³ Sports Performance Research Institute New Zealand (SPRINZ), Auckland University of Technology, Auckland 1010, New Zealand

Abstract. In this research, a novel deep-learning approach is introduced for detecting and analysing defects in golf swing actions, aiming to enhance performance and prevent injuries in sports biomechanics. Utilizing Google MediaPipe for real-time anatomical landmark estimation, a dataset of 160 golf swings was created under controlled conditions, capturing a range of swings including common errors. Joint angles were calculated from anatomical landmarks to provide comprehensive feature sets. To efficiently manage the high-dimensional data, autoencoder models were employed to compress the features while preserving critical information. The primary innovation is the development of the SCA-LSTM model, which integrates Long Short-Term Memory (LSTM) networks with Squeeze-and-Excitation (SENet) and Contextual Transformer (CoT) attention mechanisms. The integration of SENet and CoT attention mechanisms with LSTM networks significantly enhances feature representation and contextual understanding, resulting in superior performance in analysing complex motion sequences. Experimental results demonstrate that the proposed model achieves a classification accuracy of 96.88%, substantially higher than the 87.5% accuracy of baseline LSTM models. The findings underscore the model's effectiveness in detecting and analysing golf swing defects, suggesting broad applicability to various sports and training scenarios within Human Motion Modelling and Analysis (HMMA).

Keywords: Human Motion Modelling and Analysis (HMMA) · Golf swing analysis · Sports performance analysis · Squeeze-and-Excitation Attention · Contextual Transformation Attention · LSTM

1 Introduction

The golf swing is a complex motion requiring precise coordination of various body movements. Incorrect techniques not only hinder performance but also increase the risk of injuries, such as muscle strains, joint pain, overuse injuries and chronic conditions [1, 2]. Traditional coaching methods, which often rely heavily on the subjective judgment

of coaches, can struggle to provide the precision, detailed and timely feedback needed to effectively correct swing mistakes [3]. Research in motor learning and skill acquisition highlights the difficulty of unlearning ingrained faulty techniques, emphasizing the need for early and accurate feedback [4]. Therefore, early and accurate feedback is essential in developing optimal movement patterns and preventing the reinforcement of incorrect techniques.

Recent advancements in computer vision and sensor technology have paved the way for automated and enhanced golf swing detection and analysis [5–7], enabling more objective and precise evaluations. However, most machine learning-based golf models primarily focus on binary classification or scoring of swings, often falling short in helping players determine how and where adjustments should be made [8]. These methods typically provide broad quantitative analyses rather than specific, interpretable feedback that players can use to improve their game. This limitation highlights the need for more advanced models that can detect and classify swing defects, thereby providing insights on how and where to make corrections.

In this research, we propose a novel sequence classification model, SCA-LSTM, which integrates Long Short-Term Memory (LSTM) networks with Squeeze-and-Excitation (SENet) and Contextual Transformer (CoT) attention mechanisms. The model is designed to identify and classify common swing defects (i.e. common errors), thereby offering a tool for coaches and players to improve technical execution and reduce the risks of injury.

Our approach introduced a dual-attention mechanism that significantly enhances the model's sensitivity to dynamic features and temporal context. The effectiveness of our model is demonstrated through experiments, outperforming several baseline models commonly used in classification tasks.

The remainder of this paper is organised as follows: Sect. 2 reviews related work in the field of golf swing analysis and AI applications in sports biomechanics. Section 3 details the methodology, including data collection, preprocessing, and model development. Section 4 presents the experimental results and performance comparisons. Finally, the paper is concluded in Sect. 5.

2 Related Work

Artificial intelligence (AI) has revolutionised various aspects of golf, from motion analysis and optimization to shot data analysis, simulation, and prediction [5–7, 9–12]. However, existing research primarily focuses on detecting, evaluating, and scoring swing motions by comparing them with expert techniques, often falling short in providing specific adjustments needed to correct errors. A significant gap exists in the ability of current systems to offer detailed and actionable feedback to players.

With the advent of deep learning, more sophisticated and accurate sports analysis has become possible. Convolutional Neural Networks (CNNs) have been employed in several studies to classify golf swing data obtained from embedded sensors [6, 13–15]. Recurrent Neural Networks (RNNs), including LSTM and Bidirectional Long Short-Term Memory (Bi-LSTM), have also been used to classify actions like swing types and table tennis strokes [16, 17]. These studies demonstrate that LSTM and Bi-LSTM models can outperform or match the performance of CNNs in specific contexts.

The integration of pose estimation frameworks such as OpenPose [18] and Google’s MediaPipe [19] with deep learning models has enhanced real-time sports analysis [20]. These integrations allow systems to analyse mechanics and player posture, providing instant feedback. For instance, Roopa et al. demonstrated the effectiveness of such integrations in improving the accuracy and responsiveness of golf swing analysis systems [21].

Attention mechanisms have shown great promise in enhancing the performance of models for sequential data. Squeeze-and-Excitation networks (SENet), proposed by Hu et al., recalibrate feature responses through channel-wise attention, leading to significant improvements in image classification tasks [22]. Similarly, the Contextual Transformation mechanism (CoT) has effectively encoded contextual information for sequential data [23]. These mechanisms have the potential to improve the representation and analysis of complex motion sequences like golf swings.

In light of these advancements, our research aims to bridge the gap in golf swing analysis by identifying defects. We introduce the SCA-LSTM model, which incorporates both SENet and CoT attention mechanisms within an LSTM framework. This novel approach captures both local and global contextual information, significantly enhancing the model’s ability to classify complex motion sequences.

3 Methodology

The primary objective of this research is to develop a robust and accurate method for analysing golf swings to detect and correct defects, thereby improving performance and reducing the injury risks. The main objectives and contributions of this research include:

1. Captured dataset: Utilised 160 golf swing videos across four categories – lateral hip movement, body sway, incorrect arm bend, and normal swing – with common camera positions to capture front and side views.
2. Multi-time series processing: Used Google MediaPipe for 3D joint angle extraction and autoencoders for feature compression of high-dimensional data.
3. Classifier modelling: Developed SCA-LSTM, integrating SENet and CoT attention mechanisms, achieving 96.88% accuracy, significantly better than the baseline LSTM’s 87.5%.

3.1 Data Collection

The authors compiled a dataset consisting of 160 golf swings, divided into four categories. Each category includes 40 videos of golf swings captured from different angles and distances. Each video comprises 50 frames, with each frame containing posture landmark coordinates and joint angle features. The four swing categories are:

1. Lateral Hip Movement: Swings showing exaggerated lateral movement of the hips, potentially causing balance issues and inconsistent strikes.
2. Body Sway: Swings where the golfer’s upper body moves excessively backward and forward, impacting shot stability and accuracy.
3. Arm Bend: Swings with incorrect bending of the arms, affecting control and power.

4. Normal Swing: Technically correct swings for intermediate to advanced players with proper alignment, balance, and mechanics resulting in a straight ball flight.

To capture the three-dimensional aspects of sagittal and front planes for each swing, cameras were placed at various angles: 0° (front view) and 90° (side view) to the golfer's lateral side (sagittal plane). The cameras were positioned at waist height (or higher due to driving range fixtures) to consistently capture the sagittal plane, from the golfer's shoes to the golf ball.

We used Google MediaPipe to overlay stick figures on the videos, which facilitated the extraction of dynamic movement data. The technology facilitated recording landmark coordinates as time-series data for each frame, enhancing our Human Motion Modelling and Analysis (HMMA). The animated and overlayed stick figure topology, as depicted in Fig. 1, shows how landmarks are utilised to extract information from the videos.

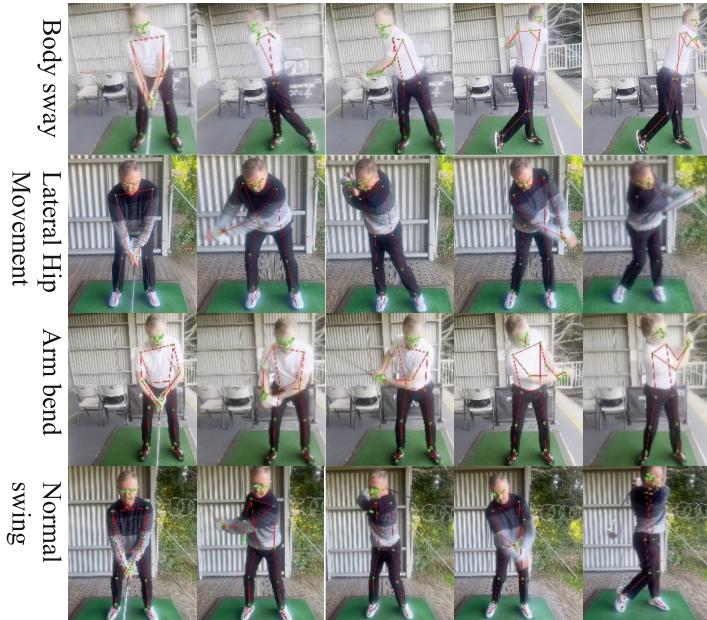


Fig. 1. Visualization of common golf swing mistakes and normal swings using MediaPipe stick figure overlay and robustness to point-of-view (POV) distortions.

Each video sequence started with a “STARTING COLLECTION” prompt to standardise the posture and swing execution. During capture, data from each frame was processed to extract pose landmarks and compute joint angles (e.g., shoulder, elbow, hip, knee) using specific mathematical formulas detailed in subsequent sections. These data were saved as.npy files for further analysis. The workflow for data collection is illustrated in Fig. 2.

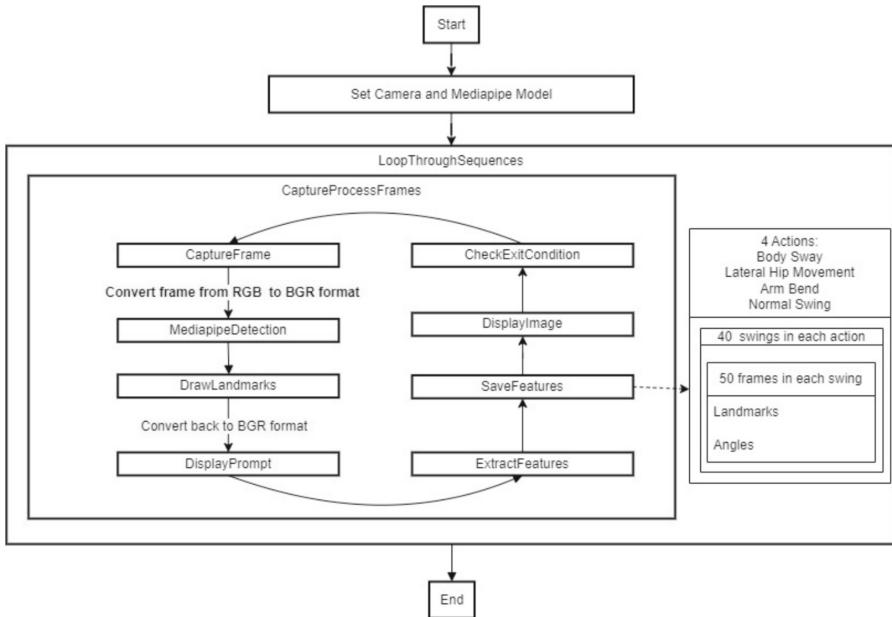


Fig. 2. Workflow diagram for data collection using MediaPipe technology.

3.2 Joint Angle Calculations

In this research, we utilised a mathematical approach to calculate the angles between key body nodes. Understanding these angles is crucial as they provide quantifiable measures of body positioning and movement dynamics during the swing. Each vector represents the position between specific body joints, enabling a detailed analysis of the golfer's form. By calculating these angles, we can objectively compare swings, identify deviations from optimal swing mechanics, and pinpoint specific areas for improvement. These angles serve as a quantifiable benchmark for assessing the correctness of the swing, allowing for precise and targeted feedback to enhance overall performance.

To analyse the bending of an arm during a swing, we define two vectors based on human joint locations:

Vector **a**: From the shoulder to the elbow.

Vector **b**: From the elbow to the wrist.

To quantify the degree of arm bending, the angle θ between two vectors is calculated by first computing the dot product of the normalised vectors to determine the cosine of the angle. The angle θ is then found using the arccosine function. This method allows for a precise measurement of the bending angle, essential for detailed motion analysis:

$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}\right) \quad (1)$$

where $\mathbf{a} = [a_x, a_y, a_z]$ and $\mathbf{b} = [b_x, b_y, b_z]$ are the 3D vectors representing the arm segments. The norm of vector **a** is calculated as $\|\mathbf{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}$.

The joint angle calculation method allows for movement analysis such as arm bending, hip movement, and body swaying in golf swings. By calculating joint angles, we can identify common errors among beginners and provide targeted feedback for improvement.

3.3 Data Pre-processing and Feature Extraction

During the data collection phase, this research utilised the MediaPipe holistic model to identify thirty-three human body landmarks. Each landmark includes four dimensions (x , y , z , visibility), resulting in 132 dimensions per frame. Additionally, angles between landmarks were calculated based on previous formulas, selecting eleven angular features in 3D space, such as arm bend angle, knee bend angle, and shoulder balance angle, etc., to construct an eleven-dimensional feature set.

To handle the high-dimensional nature of our dataset, we designed two autoencoder models specifically for pose and angle features (Fig. 3). These autoencoders compress the high-dimensional data into lower-dimensional representations while preserving critical information. This dual-autoencoder approach not only reduces computational complexity but also enhances the effectiveness of our feature extraction process, leading to more accurate and efficient model training. The workflow diagram for data pre-processing and feature extraction is below.

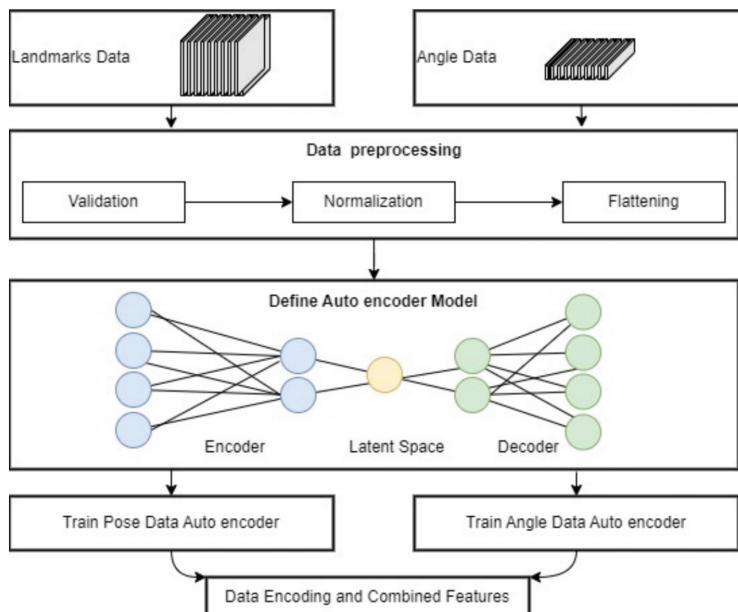


Fig. 3. The workflow diagram for data pre-processing and feature extraction.

In the data preprocessing stage, feature data were normalised to the $[0,1]$ range to stabilise the training process. A looping statement was configured in the data loading

step to ensure that each frame of every swing action had corresponding landmark and angle files, and that the data length within the files matched the predefined length. Upon successfully loading the landmark and angle data, the data were flattened to meet the input requirements of the model.

To handle these high-dimensional data, we constructed two autoencoder models, one for the pose features and another for the angle features. Each autoencoder compresses the high-dimensional data into a low-dimensional representation via an encoder layer and subsequently reconstructs the input data through a decoder layer. The encoder layer utilises the ReLU activation function, while the decoder layer employs the Sigmoid activation function. The models are trained using mean squared error (MSE) as the loss function and the Adam optimiser.

After training, we extracted low-dimensional features using the encoder and transformed them back into their original time series format. Finally, the encoded landmarks features, and angle features were concatenated to serve as input for subsequent classification tasks.

3.4 Classifier Modelling

The proposed SCA-LSTM model architecture integrates Long Short-Term Memory (LSTM) networks with Squeeze-and-Excitation (SENet) and Contextual Transformer (CoT) attention mechanisms to enhance the classification of golf swing actions. The detailed structure of the SCA-LSTM model is depicted in Fig. 4.

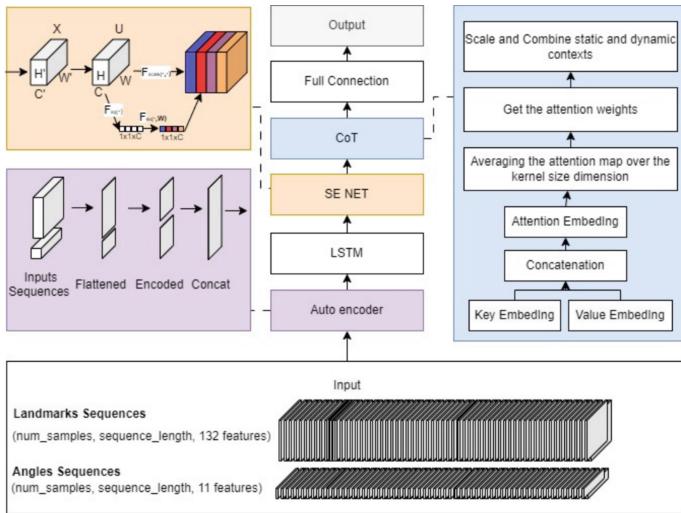


Fig. 4. Model architecture for SCA-LSTM.

Golf swings involve complex, coordinated movements where the importance of various joints and body parts can vary during different phases of the swing. To accurately capture these nuanced dependencies, the SENet mechanism is utilised to adaptively

recalibrate the feature responses by explicitly modelling the interdependencies between the channels.

Given an input $X \in \mathbb{R}^{W' \times H' \times C'}$, a set of transformations F_{tr} produces $U \in \mathbb{R}^{W \times H \times C}$:

$$U = F_{tr}(X) \quad (2)$$

The features U undergo a Squeeze operation $F_{sq}(.)$ across the spatial dimensions $H \times W$ to generate a channel descriptor $Z \in \mathbb{R}^{1 \times 1 \times C}$:

$$Z = F_{sq}(U) \quad (3)$$

Next, the Excitation operation $F_{ex}(.)$ is applied. The descriptor $Z \in \mathbb{R}^{1 \times 1 \times C}$ passes through two fully connected layers to learn channel dependencies, producing the weight $S \in \mathbb{R}^{1 \times 1 \times C}$:

$$S = F_{ex}(Z) = \sigma(W_2 \delta(W_1 Z)) \quad (4)$$

where W_1 and W_2 are the weights of the fully connected layers, δ is the ReLU activation function, and σ is the sigmoid function. These weights S are then multiplied elementwise with the input features U to recalibrate the channels:

$$\text{output}_{SE} = S \odot U \quad (5)$$

The SENet mechanism enhances the model's ability to focus on the most critical parts of the input features, thus improving the detection of subtle defects in the swing that may otherwise be overlooked.

In addition to capturing channel dependencies, understanding the spatial and contextual relationships between different body parts is crucial for accurately classifying golf swing actions. The CoT attention mechanism initially encodes static context information using convolution operations. A 1×1 convolution layer encodes the input features, generating the value matrix V . The context-aware keys K_1 are derived by encoding the spatial information of all adjacent keys within a $K \times K$ grid. The key matrix K and the original features Q (queries) are concatenated along the channel dimension and passed through two additional 1×1 convolution layers.

The resulting matrix A is the attention matrix, where a softmax operation is applied to produce weights. For each spatial position (i, j) in the k -th head, the local attention matrix is computed based on the query features $Q_{(i,j)}$ and the context-aware key features:

$$\text{AttentionMatrix}_{(i,j,k)} = \text{softmax} \left(\frac{(Q_{(i,j)} W_k^Q)(K_{1(i,j)} W_k^K)^T}{\sqrt{d_k}} \right) \quad (6)$$

Where $Q_{(i,j)}$ is the query vector at position (i, j) , W_k^Q is the learned projection matrix for the query in the k -th head, $K_{1(i,j)}$ is the context-aware key vector at position (i, j) , W_k^K is the learned projection matrix for the key in the k -th head, d_k is the dimension of the keys (and queries).

The value matrix V is then weighted and summed using these weights to generate the dynamic context representation K_2 :

$$K_2 = A \odot V \quad (7)$$

Finally, the static context representation K_1 and the dynamic context representation K_2 are fused to form the final output:

$$\text{output} = \text{Fusion}(K_1, K_2) \quad (8)$$

The dataset consists of feature vectors and corresponding labels. We use a LabelEncoder to encode the feature vectors, converting them into one-hot encoded format. The dataset is split into training and testing sets, and DataLoader is utilised to facilitate batch training. During training, the model uses the cross-entropy loss function and the Adam optimiser. For each batch, forward propagation is performed to compute outputs and loss, followed by backpropagation to update the model parameters. Throughout the training process, we monitor and output the loss value to assess model convergence.

In the evaluation phase, we perform forward propagation on the test set and calculate the prediction accuracy. By comparing the model's predicted values with the actual labels, we evaluate the model's performance. The overall architecture design and the application of attention mechanisms endow the model with enhanced feature extraction and representation capabilities when processing sequence data, thereby improving classification performance.

The integration of SENet and CoT mechanisms within the LSTM framework provides a dual advantage:

1. Enhanced Feature Representation: SENet mechanism ensures that the model emphasises the most relevant features, dynamically adjusting its focus based on the interdependencies of the input features.
2. Improved Contextual Understanding: CoT mechanism captures both local and global contextual information, allowing the model to understand the intricate spatial relationships and temporal dependencies within the golf swing data.

3.5 Dataset and Experimental Setup

We split our proprietary dataset of 160 swing actions into training and testing sets in an 80:20 ratio. The input to the model is the feature-extracted time series data, and the output is the defect category of the swing. The key parameters (Table 1) for the experiments are as follows:

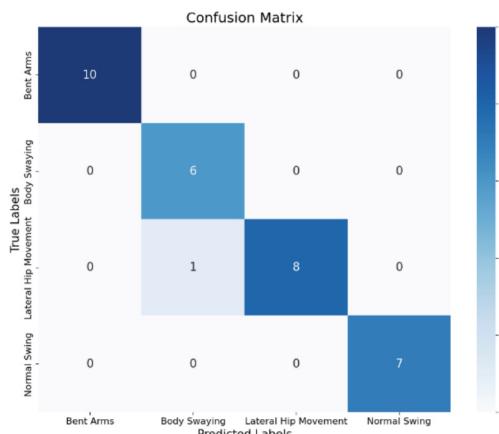
4 Results

4.1 Confusion Matrix

The confusion matrix for the test set was plotted to further analyse the model's performance across different action categories. Figure 5 illustrates the confusion matrix of the SCA-LSTM model on the test set.

Table 1. Model configuration parameters

Parameter	Value
LSTM hidden layer dimension	128
Number of LSTM layers	2
Channel dimension for SENet attention mechanism	128
Dimension for CoT attention mechanism	128
Kernel size for CoT convolution	3
Training epochs	50
Learning rate	0.001

**Fig. 5.** Confusion matrix of the SCA-LSTM model on the test set.

The confusion matrix shows that the model achieves high classification accuracy in most categories, with occasional misclassifications in certain categories. To address these misclassifications, we can further optimise the feature extraction methods or increase the amount of training data to enhance the model's performance.

4.2 Training and Validation Loss Curves

To further evaluate the performance of our proposed SCA-LSTM model, we analysed the training and validation loss curves over the 50 epochs (Fig. 6). The loss curves provide insights into the learning process and model convergence. The training loss consistently decreased from the initial epoch to the final epoch, indicating effective learning and optimization. Similarly, the validation loss showed a steady decline with minor fluctuations, reflecting the model's ability to generalise well to unseen data. The minimal gap between the training and validation losses towards the later epochs suggests that the model effectively mitigates overfitting, achieving a balance between learning the training data and maintaining performance on the test set. These observations further

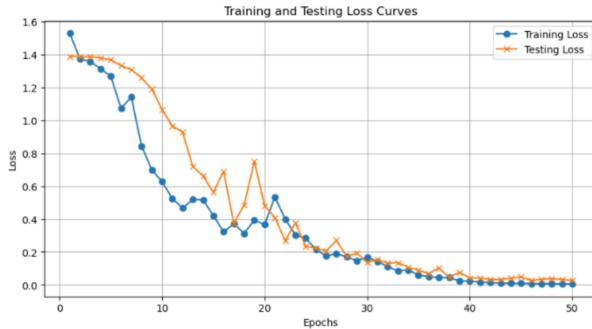


Fig. 6. Training and validation loss curves of the SCA-LSTM model over the 50 epochs

confirm the robustness and reliability of our SCA-LSTM model in accurately classifying golf swing defects.

4.3 Performance Comparison

We also compared the proposed model with several commonly used sequence classification models for detecting defects in golf swing actions. Tables 2 and 3 present the classification accuracy, F1 scores, recall, and precision of each model. The experimental results reveal that the SCA-LSTM model significantly outperforms the other models, achieving an accuracy of 96.88%, with an F1 Score of 0.9641, Recall of 0.9583, and Precision of 0.9750. The performance advantage stems from the inclusion of Squeeze-and-Excitation (SENet) and Contextual Transformer (CoT) attention mechanisms, which greatly enhances the model's performance. In comparison, both the Bi-LSTM and Transformer models have high test accuracies of 90.62%, but their F1 scores are lower at 0.9035 and 0.8923. Models like GRU, LSTM, RNN, and R-CNN perform with test accuracies, and F1 scores are all below 90%.

Table 2. Model performance comparison

Model	Test Accuracy	Precision	Recall	F1 Score
GRU	78.12%	0.8482	0.7812	0.7314
BI-LSTM	90.62%	0.9164	0.9062	0.9035
Transformer	90.62%	0.9	0.8889	0.8923
LSTM	87.5%	0.8902	0.8591	0.8667
RNN	81.25%	0.8447	0.8056	0.7992
R-CNN	87.5%	0.8693	0.8472	0.85
SCA-LSTM (Our Model)	96.88%	0.9641	0.9583	0.9750

The results indicate that the SCA-LSTM model has superior classification performance in the task of detecting defects in golf swing actions.

4.4 Ablation Study

To verify the effectiveness of the SENet attention mechanism and CoT attention mechanism in enhancing model performance, we conducted ablation experiments. We compared the proposed model (SCA-LSTM) with the traditional LSTM model and the LSTM model with SENet attention mechanism in terms of classification accuracy.

Table 3 presents the performance of different models on a classification task and the results of an ablation study. The models include the Baseline LSTM Model, LSTM Model without SENet Mechanism, and the SCA-LSTM Model. The evaluation metrics are Accuracy, F1 Score, Recall, and Precision.

Table 3. Model performance and ablation study results

Model	Accuracy	F1 Score	Recall	Precision
Baseline LSTM Model	87.5%	0. 8667	0. 8591	0. 8902
With SENetNET Mechanism	93.75%	0.9375	0.9375	0.9375
SCA-LSTM Model	96.88%	0. 9641	0. 9583	0.9750

The Baseline LSTM Model achieved an accuracy of 87.5%, an F1 score of 0.8486, a recall of 0.8472, and a precision of 0.8648. Metrics for the baseline model suggest reasonable performance, yet they highlight potential areas for enhancement.

The LSTM Model with SENet Mechanism showed significant improvements across all metrics, achieving 93.75% in accuracy, F1 score, recall, and precision. The results affirm the beneficial impact of the SENet mechanism on model performance for this task.

The SCA-LSTM Model demonstrated the best performance, with an accuracy of 96.88%, an F1 score of 0.9641, a recall of 0.9583, and a precision of 0.9750. The proposed LSTM model combining SENet and CoT attention mechanisms significantly outperforms other baseline models in classification accuracy. This indicates that the simultaneous application of SENet and CoT attention mechanisms allows the model to capture critical features and contextual information more effectively, thus improving classification performance.

The experimental results demonstrate that the proposed SCA-LSTM model achieves a classification accuracy of 96.88%, significantly outperforming baseline models such as traditional LSTM, Transformer, and Recurrent Neural Network (RNN). These findings underscore the potential of our approach in the field of Human Motion Modelling and Analysis (HMMA), suggesting its applicability to a wide range of sports and training/coaching scenarios.

5 Discussion

The results of this study clearly demonstrate the efficacy of the proposed SCA-LSTM model in identifying and classifying common defects in golf swings (i.e. common errors). The high classification accuracy of 96.88% significantly surpasses baseline models,

indicating that the integration of SENet and CoT attention mechanisms within the LSTM framework enhances both feature representation and temporal context processing. These improvements suggest that the model can capture critical biomechanical nuances in complex movements like golf swings.

As a limitation of this research, the relatively small sample size of 160 swings could have impact on the generalizability of the findings to broader populations of golfers. However, it is also a common knowledge that novice and advanced beginners share typical i.e. common errors. An extended dataset that includes various skill levels, such as beginner, intermediate, and professional golfers, would help strengthen the conclusions and better validate the model's robustness across different swing styles and conditions. Regarding the concept of approximate reasoning, extended dataset collection integrating environmental factors such as lighting conditions and diverse playing surfaces may help simulate real-world variability and data capture inconsistencies.

6 Conclusion

This research introduces a deep learning-based approach for detecting defects in golf swings, utilizing a novel dataset and model for complex motion analysis. The collected dataset includes 160 golf swings categorised into common errors (lateral hip movement, body sway, arm bend), and normal swings, collected under diverse conditions for robustness. Using Google MediaPipe, high-dimensional time-series data were generated for detailed analysis. Our SCA-LSTM model combines Long Short-Term Memory (LSTM) networks with Squeeze-and-Excitation (SENet) and Contextual Transformer (CoT) attention mechanisms, significantly improving feature representation and contextual understanding. The SCA-LSTM model achieved a classification accuracy of 96.88%, surpassing the baseline LSTM model's 87.5%. The successful application of this model in golf swing analysis highlights its potential for more general applications in sports biomechanics and motion analysis. In addition to golf, the methodology presented here can be adapted to other sports where motion analysis and defect detection are critical. For example, in tennis, baseball, and even running, where the form and technique are crucial for performance and injury prevention, this model can be applied to provide athletes and coaches with comprehensive, explainable and actionable insights. The feature extraction methods utilizing 3D joint angles and pose landmarks can be extended to different motion sequences, providing a wide scope for further research and development in human motion modelling and sports performance analysis.

Future research will focus on expanding the dataset and exploring real-time model implementation for immediate feedback during training. Further refinement of the attention mechanisms and exploration of advanced sequence modelling techniques will be pursued to optimise performance across diverse sports applications. This study provides a solid foundation for future innovations in automated sports training tools, contributing to enhanced athlete performance and injury prevention.

Disclosure of Interests. The authors declare that there are no competing interests in this paper. The inclusion of one of the co-author's photographs in the paper, was in compliance with ethical requirements and exceptions of Auckland University of Technology Ethics Committee (AUTEC)

guidelines (exception section 6.4. “Research and teaching in which a single investigator is the subject of his/her own research and where no physically or psychologically hazardous procedure is involved.” – <https://www.aut.ac.nz/research/researchethics/guidelines-and-procedures#6>, accessed 25 June 2024).

References

1. Knudson, D.V.: Fundamentals of biomechanics, vol. 183. Springer
2. McHardy, A., Pollard, H., Luo, K.: Golf injuries. *Sports Med.* **36**(2), 171–187. <https://doi.org/10.2165/00007256-200636020-00006>
3. McCormack, S., Jones, B., Elliott, D., Rotheram, D., Till, K.: Coaches’ assessment of players physical performance: subjective and objective measures are needed when profiling players. *Eur. J. Sport Sci.* **22**(8), 1177–1187
4. Lee, T.D., Swinnen, S.P., Serrien, D.J.: Cognitive effort and motor learning. *Quest* **46**(3), 328–344
5. Ball, K.A., Best, R.J.: Different centre of pressure patterns within the golf stroke I: cluster analysis. *J. Sports Sci.* **25**(7), 757–770. <https://doi.org/10.1080/02640410600874971>
6. Ko, K.R., Pan, S.B.: CNN and Bi-LSTM based 3D golf swing analysis by frontal swing sequence images. *Multimedia Tools Appl.* **80**, 8957–8972 (2021). <https://doi.org/10.1007/s11042-020-10096-0>
7. Zhang, L., Hsieh, J.-C., Ting, T.-T., Huang, Y.-C., Ho, Y.-C., Ku, L.-K.: A Kinect based golf swing score and grade system using GMM and SVM. In: 2012 5th International Congress on Image and Signal Processing, vol. 10, pp. 711–715 (2012). <https://doi.org/10.1109/CISP.2012.6469827>
8. Hao, Y.: Research on the applications of artificial intelligence in golf. In 2022 3rd International Conference on Artificial Intelligence and Education (IC-ICAIE 2022), pp. 1588–1595. Atlantis Press. https://doi.org/10.2991/978-94-6463-040-4_240
9. McNally, W., Lambeth, J., Brekke, D.: Combining physics and deep learning models to simulate the flight of a golf ball. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5119–5128
10. Sugawara, S., Kawamura, H., Suzuki, K.: Skill-based simulation model for optimizing strategy in golf. In: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1591–1596. <https://doi.org/10.1109/AIM.2013.6584323>
11. Baćić, B.: Predicting golf ball trajectories from swing plane: an artificial neural networks approach. *Expert Syst. Appl.* **65**, 423–438. <https://doi.org/10.1016/j.eswa.2016.07.014>
12. Bacic, B., Meng, Q., Chan, K.Y.: Privacy preservation for esports: A case study towards augmented video golf coaching system. In: 2017 10th International Conference on Developments in eSystems Engineering (DeSENNet), pp. 169–174. IEEE (2017)
13. Connaghan, D., Kelly, P., O’Connor, N.E., Gaffney, M., Walsh, M., O’Mathuna, C.: Multi-sensor classification of tennis strokes. *IEEE Sensors* (2011). <https://doi.org/10.1109/icsens.2011.6127084>
14. Jiao, L., Bie, R., Wu, H., Wei, Y., Ma, J., Umek, A., et al.: Golf swing classification with multiple deep convolutional neural networks. *Int. J. Distrib. Sens. Netw.* **14**(10), 1550147718802186. <https://doi.org/10.1177/1550147718802186>
15. Jiao, L., Wu, H., Bie, R., Umek, A., Kos, A.: Multi-sensor golf swing classification using deep CNN. *Procedia Comput. Sci.* **129**, 59–65. <https://doi.org/10.1016/j.procs.2018.03.046>
16. Anand, A., Sharma, M., Srivastava, R., Kaligounder, L., Prakash, D.: Wearable motion sensor-based analysis of swing sports. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 261–267. <https://doi.org/10.1109/ICMLA.2017.0-149>

17. Tabrizi, S.S., Pashazadeh, S., Javani, V.: Comparative study of table tennis forehand strokes classification using deep learning and SVM. *IEEE Sens. J.* **20**(22), 13552–13561 (2020). <https://doi.org/10.1109/JSENNetN.2020.3005443>
18. Cao, Z., Simon, T., Wei, S.-E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299
19. Lugařes, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., et al.: MediaPipe: a framework for building perception pipelines. arXiv preprint [arXiv:1906.08172](https://arxiv.org/abs/1906.08172) (2019)
20. Bačić, B., Bandara, I.: Tennis strokes recognition from generated stick figure video overlays. In: 17th International Conference on Computer Vision Theory and Applications, pp. 397–404. Online Streaming: SCITEPRESS - Science and Technology Publications. <https://doi.org/10.5220/0010827300003124>
21. Roopa, D., Prabha, R., Sridevi, S.: Senthil An artificial intelligence improved golf self-training system using deep learning. In: IEEE Conference Publication. IEEE Xplore (2022)
22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141
23. Li, Y., Yao, T., Pan, Y., Mei, T.: Contextual transformer networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(2), 1489–1500 (2023). <https://doi.org/10.1109/TPAMI.2022.3164083>



Unsupervised Document Image Tampering Localization via Anomaly Detection

Yuan Li^{1,2(✉)}, Yan-Ming Zhang², Fei Yin², and Lin-Lin Huang¹

¹ Beijing Jiaotong University, Beijing 100044, China
{22120095,huangll}@bjtu.edu.cn

² State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
{ymzhang,fyin}@nlpr.ia.ac.cn

Abstract. Digital document images play a critical role in daily life. With the great advances in image editing techniques, document image tampering localization is becoming increasingly important. Most existing methods for document image tampering localization heavily rely on tampered image synthesis and pixel-level annotations for training. However, both acquiring tampered document images and labeling the tampered regions in real-world scenarios are expensive, which motivates us to solve this task by unsupervised learning using authentic images only. In this work, we propose a feature reconstruction network to learn global comprehension of authentic images, and then identify anomalies with high reconstruction errors as tampered pixels. Particularly, we integrate visual features and frequency domain compression artifacts to expose tampering traces from multi-views. We design a random rectangular mask (RRM) strategy that leverages the prior knowledge of text shapes to prevent over-reconstruction of tampered regions. Evaluation on the benchmark dataset DocTamper-FCD/SCD demonstrates that our approach dramatically outperforms other unsupervised baselines, and shows the effectiveness of our method.

Keywords: Tampering localization · Unsupervised anomaly detection · Document image editing · Masked transformer

1 Introduction

Document images play an indispensable role in modern society, enabling efficient information transmission across fields like Office Automation, E-government and E-commerce. Containing amounts of critical and private information, document images are highly sensitive to semantic alterations. Even minor tampering with individual characters can result in significant semantic discrepancies. Meanwhile, advancements in image editing technologies have made it easier than ever to manipulate document images, creating opportunities for malicious tampering

and posing significant risks to information security. As a result, accurately localizing tampered text has become a critical concern for ensuring the authenticity and integrity of document images.

Digital image tampering detection and localization have been extensively studied, particularly focusing on natural and facial images [10, 12, 20]. However, localizing forgery in document images presents unique challenges due to the prominence of characters, which exhibit distinctive features such as regular distribution and uniform background textures. In the early stages of tampered text localization, detectors relied on statistical modeling using hand-crafted features like printer identification texture [26], mismatched fonts [3], inconsistent text lines [15] and splicing boundaries [28]. These hand-crafted features, derived from a limited scale of typical samples, were effective only for specific document types and tampering scenarios. Recently, with advancements in deep neural models, several robust methods for localizing document tampering have emerged [11, 21, 29, 36]. These approaches leverage pixel-level annotated synthetic datasets of tampered document images and demonstrate the effectiveness of supervised deep learning networks.

While several effective deep learning methods have been proposed, their performance heavily hinges on the quality and scale of annotated data. For instance, the DTD model [25], which achieves notable results, relies on a substantial dataset of 120,000 annotated synthetic images. These synthetic datasets, like DocTamper [25], generated through specific tailored tampering pipelines with fixed traces left, may not capture the complexity and diversity of real-world tampering scenarios. Meanwhile, real tampered document images, such as contracts, certificates, and receipts, present challenges for collection due to the sensitive nature of the information they contain, and obtaining fine-grained annotated labels for real-world tampered images is often impractical or unavailable. On the other hand, acquiring datasets of untampered document images is considerably easier compared to tampered ones.

In this paper, we assume that tampered text displays discernible differences compared to surrounding pristine regions, appearing as anomalous representations within a multi-view feature space. Based on the assumption, we propose an unsupervised anomaly localization network trained exclusively on normal document samples. It is designed to accurately reconstruct features within the pristine range while failing to do so for tampered regions, thereby identifying tampered pixels as anomalies with high reconstruction errors. In detail, our approach integrates RGB images and the DCT coefficients as inputs to capture visual inconsistency features and compression artifices in the frequency domain, compensating for deficiencies stemming from monochromatic color. We employ a masked vision transformer-based encoder-decoder network [31] to learn the discrimination between normal and tampered regions through feature reconstruction. We introduced the random rectangular mask strategy to enhance the global comprehension of normal features and prevent the over-reconstruction of anomalous features. Our approach is inspired by the feature reconstruction-based anomaly detection model UniAD [37] but enhances it with the aforementioned multi-view

feature extraction and novel mask strategy. To the best of our knowledge, our approach represents the first unsupervised deep-learning method for document tampering localization. By analyzing latent features from authentic samples to reveal abnormal traces of tampered text, our method more closely aligns with human intuition. Experiments on the benchmark dataset DocTamper-FCD/SCD demonstrate our approach outperforms other unsupervised networks in document tampering localization tasks. The main contributions of this work lie in four aspects:

- We address the document image tampering localization problem with a feature reconstruction framework for unsupervised anomaly detection, where anomalous representations can be regarded as outliers in the feature space.
- We propose a multi-view feature extractor combines visual and frequency domain features, revealing more detailed traces.
- We propose a random rectangular mask strategy to mitigate the over-reconstruction of anomalies, thereby improving model robustness and global feature understanding.
- We conducted extensive experiments on the DocTamper-FCD/SCD datasets and substantiated the suitability of our method.

2 Related Works

2.1 Natural Image Tampering Localization

With the rapid development of deep learning, advanced neural network techniques are increasingly used to extract multi-view tampering traces. These traces, such as compression artifacts, noise patterns, edge inconsistencies, and visual similarities, are critical for accurately identifying tampered images. One prominent area of focus is the fusion of RGB and noise information. For instance, Zhou et al. [39] incorporated Steganalysis Rich Model (SRM) filters into Faster-RCNN. Similarly, Yang et al. [2] enhanced their CR-CNN model by using BayarConv. MVSS-Net [10] employs a two-stream CNN to jointly analyze noise distributions and boundary artifacts. Beyond noise fusion, other works have explored integrating both visual and frequency domain features. For example, CAT-Net [18] integrates DCT coefficients into a segmentation network, while ObjectFormer [32] combines high-frequency features with RGB features, embedding them as multi-modal patches.

While above data-driven methods have made significant strides, their efficacy heavily relies on the quality and size of training datasets. From this perspective, several methods exploit rich-model features for unsupervised tampering localization. Splicebuster [5] utilizes Expectation-Maximization (EM) clustering for co-occurrence-based local features, fed into an Auto-Encoder (AE) to identify spliced images regions. Additionally, the EM framework is applied in Noiseprint [6] for camera fingerprint features extraction. However, these methods typically focus on specific tampering types or image categories. In contrast to ManTra-Net [35], which utilizes FCN and Long Short-Term Memory (LSTM)

for identifying local anomalies, Chen et al. [4] proposed the VAE-ViT framework treats tampering localization as local anomaly detection, utilizing cliques, Noiseprint, high-pass filter residuals, and Laplacian edge maps to identify forged areas with high reconstruction errors. While the VAE-ViT anomaly localization framework is the most relevant to ours to the best of our knowledge, the target category of images and the proposed solution, strongly model-based, differ entirely from ours.

2.2 Document Image Tampering Localization

Different from natural images, the tampered text regions in document images usually have much more visual consistency with the authentic regions in the background. Therefore, early studies on document image tampering localization mainly focused on document-specific printer identification features [26], font-based features [3], alignment or skew-based features [15]. Francisco Cruz et al. [7] extract Local Binary Patterns (LBP) texture features from character patches, leveraging Support Vector Machines (SVM) to classify neighboring patches as either forged or non-forged. James et al. [16] treating each text block as a graph node employ a graph neural network (GNN) for node classification to detect text splicing tampering. However, these methods only perform well on clear and neatly manipulated documents with specific alterations.

In this case, some works propose robust feature extractors by incorporating additional information and prior knowledge as auxiliary cues. Xu et al. [36] introduced dual-stream feature extractors: one stream focuses on spatial information from document image patches, while the other detects abnormal characteristics using skip connections. Wang et al. [34] employed a two-stream CNN operating in the RGB-frequency domain to capture spatial and frequency domain clues simultaneously. Liao et al. [22] presented the Character Texture Stream and the Image Texture Stream, focusing respectively on extracting textual features and leveraging texture features across entire documents. Yu et al. [38] addressed smartphone screenshot forgery by integrating an optical character recognition (OCR) stream with RGB stream. To overcome the challenge caused by the lack of annotated tampered document images, some specially designed synthetic tampered datasets have been generated [21, 22, 24, 36], though access to these datasets remains limited. Trained and evaluated on the largest public dataset DocTamper, Qu et al. [25] fed visual and frequency embeddings into a multi-modality Transformer-based encoder-decoder to locate the tampered text regions. Shao et al. [27] concatenated RGB signals, noise residual and texture information as multi-view forensic perspectives.

Despite these advancements, a significant challenge persists: the dependency on high-quality labeled datasets for supervised training. So far, the only blind detection method proposed by Sun et al. [28], calculated first-order differences in row and column sizes to identify abnormal variations at image splicing boundaries. However, this simple method is limited to specific documents with clearly discernible splice edges.

3 Methodology

Inspired by UniAD [37], we propose an unsupervised document tampering localization framework based on feature reconstruction, illustrated in Fig. 1. The framework consists of two main sections: multi-view feature extraction and masked feature reconstruction. Initially, we employ a dual-branch multi-view feature extractor (Sect. 3.1) to process input images and their DCT coefficients, extracting integrated visual and frequency-domain features. Subsequently, these multi-view features are fed into a Masked Transformer Encoder-Decoder (MTED) for feature reconstruction. To prevent over-reconstruction that may overlook anomalies, we design a Random Rectangular Mask (RRM) strategy (Sect. 3.2) to transform multi-view features into masked feature tokens. Next, the masked feature tokens are passed through the self-attention encoder and layer-wised query decoder so as to generate the reconstructed feature (Sect. 3.3). During inference, by computing feature reconstruction errors, we obtain anomaly scores and perform pixel-level localization predictions (Sect. 3.4).

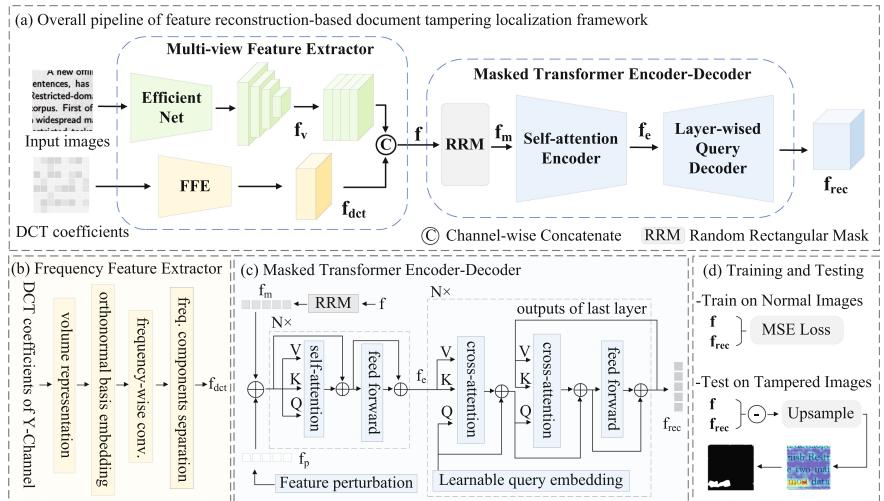


Fig. 1. (a) The pipeline of our feature reconstruction-based network. (b) Framework of frequency feature extractor. (c) Framework of MTED. (d) Training and testing phase.

3.1 Multi-view Feature Extraction

It is essential to extract features that can detect subtle anomalies at the pixel level. In particular, features from the visual domain are essential since humans often detect tampered regions by checking for inconsistencies in color, texture, edges, or content. However, as image tampering techniques evolve, the visual

traces become harder to detect, making it necessary to explore other techniques, such as analyzing the frequency domain.

When images are captured, they are often compressed for storage using a process that includes Discrete Cosine Transform (DCT) block processing and quantization, which leads to what is known as Block Artifact Grids (BAG) [14]. Tampering typically occurs post-acquisition of images, causing a misalignment in DCT coefficient distribution between tampered and authentic compression grids, causing anomalies in the DCT statistics [33].

Our Dual-branch Multi-view Feature Extractor separately obtains multi-scale visual features and compression artifices in the frequency domain. For input RGB images $I \in \mathbb{R}^{C \times H \times W}$, we adopt a fixed-weight EfficientNet-b4 [30] pre-trained on ImageNet [8] as visual branch to obtain multi-scale features with $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$ resolutions. For frequency branch, as shown in Fig. 1(b), we convert I to YCrCb space to obtain 8×8 blocked DCT coefficient map of Y-channel map equal-size of $H \times W$. Since the spatial positions of DCT coefficients correspond to different frequency components, to preserve pixel-level positional information, we represent the raw coefficients as a binary volume [18]. Then we use an orthonormal basis as pre-trained weights for volume representation head embedding. Aiming at extracting features among different DCT blocks, we employ a frequency-wise convolution block consisting of 8×8 convolutions with 8 dilation to prevent admixture among various frequency, alongside a 1×1 convolution for dimension reduction. Then separate 64 frequency components, reshaping $f_{dct'} \in \mathbb{R}^{c' \times H \times W}$ to $f_{dct} \in \mathbb{R}^{64c' \times \frac{H}{8} \times \frac{W}{8}}$, each channel represent a frequency component. Lastly in the feature extraction stage, we merge the multi-scale visual features into the dimensions of f_{dct} , denoted as f_v and concatenate the features in channel dimension as the multi-view features $f \in \mathbb{R}^{c \times h \times w}$.

3.2 Random Rectangular Mask Strategy

According to our hypothesis, the feature distributions of real and tampered pixels are different in the visual and frequency domains, and we define all these unknown differences as anomalies. However, given the great capacity of deep neural networks, training of reconstruction models easily will probably result in over-reconstruction, where both normal and tampered features are perfectly reconstructed, making it difficult to distinguish anomalies. To mitigate this, inspired by the Masked Auto Encoder (MAE) [13], we mask the multi-view features before the reconstruction stage. This strategy aims to enhance holistic image understanding and specifically avoid over-reconstruction due to information leakage. Unlike image block-level reconstruction, features extracted from multi-view CNN networks encompass information from adjacent regions, where nearby points exhibit significant similarity. Therefore, random single point masking, as shown in Fig. 2(a), allows vanilla Transformers to easily transfer information from surrounding points for reconstruction, regardless of latent global correlations. To address this issue, we design a random rectangular mask strategy to obtain masked multi-view features $f_m \in \mathbb{R}^{c \times h \times w}$.

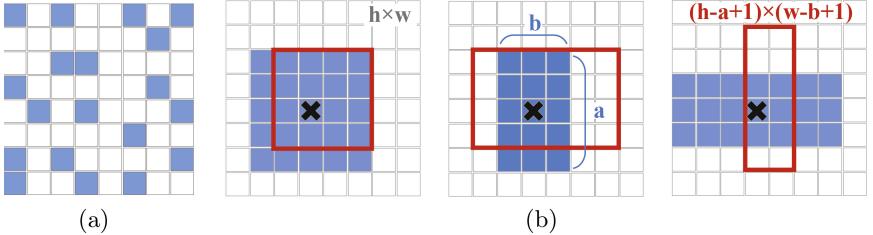


Fig. 2. Random Mask Strategy. (a) Random single mask. (b) Random rectangular mask. The blue block indicates masked feature point. In (b), the black cross is the selected center point of rectangular mask, and the red boundary outlines the candidate region for mask center selection. (Color figure online)

Our mask strategy reflects stochasticity in two aspects: random rectangular shapes and random position selection. Due to varying character lengths in tampered texts, the aspect ratios of tampered regions differ. Thus, we design the shapes of rectangular masks, denoted as $a \times b$, to vary randomly within a specified range, aligning with the diverse characteristics of tampered texts. As shown in Fig. 2(b), the center of the mask is selected to generate masked feature tokens within a candidate region sized $(h - a + 1) \times (w - b + 1)$. To ensure that each token has an equal probability p of being masked, a token within the candidate region is chosen as the center of the mask is determined by a Bernoulli distribution with the probability β , which is determined as follows:

$$\beta = \frac{p}{a \cdot b} \cdot \frac{h \cdot w}{(h - a + 1) \cdot (w - b + 1)}, \quad (1)$$

Applying such non-uniform random masks with varying shapes to different channels effectively mitigates excessive information loss, which might lead to reconstruction failures, and also effectively avoids the risk of identity mapping that fails to distinguish between normal and tampered regions.

3.3 Multi-view Feature Reconstruction

As shown in Fig. 1(c), following the vanilla transformer architecture [31], we apply N self-attention layers with an attention module and a feed-forward network as the encoder. After mapping the f_m to $h \times w$ feature tokens $f_{tok} \in \mathbb{R}^c$, we inspire the model to learn robust distribution of normal samples by adding feature perturbations tokens f_p with a fixed probability. f_p is generated from Gaussian noise distribution:

$$f_p \sim N(\mu = 0, \sigma^2 = (\gamma \cdot \frac{\|f_{tok}\|_2}{c})^2), \quad (2)$$

where γ controls the noisy degree. We linearly project the fused tokens to reduce the channel number to C , and create query, key, and value matrices for the self-attention encoder to obtain the encoded embedding f_e .

As UniAD [37], we also incorporate learnable query embedding into N transformer decoder layers to mitigate over-reconstruction. Each decoder layer consists of two attention modules and a feed-forward network. Initially, the encoder embedding f_e is fused with the learnable query embedding in the first cross-attention block. Then in the second cross-attention block it integrates with the outputs of the preceding layer. For the first decoder layer, the second cross-attention block is replaced with self-integration. Since we implemented the random masks before encoding to avoid the information leak effectively, our attention blocks do not necessitate multiple masks like UniAD. Additionally, learnable position embedding [9] is integrated into all attention modules to encode spatial information. The attention blocks facilitate multi-view information interaction and feature fusion. Residual connections aid in training stability. The resulting reconstructed feature is denoted as $f_{rec} \in \mathbb{R}^{c \times h \times w}$.

During the authentic features reconstruction training stage, we utilized the mean square error (MSE) as the reconstruction loss:

$$Loss = \frac{1}{h \cdot w} (\|f - f_{rec}\|_2^2). \quad (3)$$

3.4 Anomaly Localization

Once the reconstruction learning stage is completed, the network parameters are frozen. To perform tampering localization, a document image is processed with the network to generate a pixel-wise anomaly score map at the original image size, denoted as $A \in \mathbb{R}^{H \times W}$, which is achieved by up-sampling the L2 norm of the reconstruction differences using an interpolation operation:

$$A = Upsample(\|f - f_{rec}\|_2). \quad (4)$$

Finally, we scale anomaly score map A to the range [0,1] to form an abnormal probability heat map. Pixels with abnormal probabilities greater than a threshold are identified as tampered.

4 Experiments

4.1 Experimental Setting

Datasets: Publicly available datasets for document tampering localization are limited [11, 34]. Most datasets focus exclusively on tampered data without including authentic counterparts due to privacy concerns. Considering factors such as dataset scale, document diversity, and various tampering techniques, we utilize DocTamper [25], the largest dataset released in 2023. DocTamper includes a broad range of document types, such as contracts, invoices, and receipts in both Chinese and English, and covers three tampering methods: copy-move, splicing, and generation. In our experimental setup, we use 35,000 randomly cropped untampered document images from DocTamper for training. Testing is conducted on pixel-level annotated tampered datasets, DocTamper-FCD/SCD.

Performance Metrics: Following previous works [25, 34, 37], we consider document tampering localization as a pixel-level binary classification task and adopt the Area Under the Receiver Operating Curve (AUC), Precision (P), Recall (R) and F-score (F) as the evaluation metric.

Implementation Details: Recognizing the potential loss of tampering clues resulting from image reshaping operations, we directly crop authentic images at a fixed size of 224×224 . The visual features extracted from stages 1–4 of EfficientNet-b4 [30] are merged to $272 \times 28 \times 28$ dimensions, which are then concatenated with the 256-channel frequency features to create the input multi-view feature maps. The reduced channel dimension is set as 512. Both encoder and decoder is set 4-layers. The noisy degree of feature perturbations γ is set to 20. Each token undergoes masking with an equal probability of 0.5. The side lengths of the rectangular masks are randomly generated odd numbers within the range [3, 17], with the limited area less than 9×9 . Specifically, during each training iteration, we create random shapes of masks with varying positions to enhance the robustness of our model. Our model is trained for 1000 epochs on 4 NVIDIA Titan RTX GPUs with batch size 24, using the AdamW optimizer with the learning rate of 1e-4 and dropped by 0.1 after 800 epochs.

We utilize sliding windows on larger tampered images during testing phase, maintaining the parameter settings from training phase to obtain 224×224 anomaly score maps within a larger image, then average the results from these sliding windows to restore anomaly score maps to the original image size. The difference is we omit the RRM operation, directly feed the raw multi-view features into the feature reconstruction stage for inference.

4.2 Ablation Study

Multi-view Features: The multi-view feature extractor is designed to identify tampering clues in both visual and frequency domains with RGB images and DCT coefficients. To evaluate the effectiveness of our multi-view features, we analyze the effect of different features on localization performance: RGB-only features in 14×14 and 28×28 , raw DCT coefficients features and DCT volume processed features. We conduct ablation study of multi-view features with the same reconstruction network and report the tampered text localization performance on the FCD/SCD datasets in Table 1.

When using only RGB images, performance is lower, indicating that visual features alone are insufficient for detecting subtle tampering traces. While using smaller-scale features reduced the computational complexity of the Transformer-based network, it resulted in a decline in pixel-level localization accuracy, suggesting that finer-scale features are critical for this task. The raw DCT coefficients features directly extracted from three frequency-wise convolution layers with frequency components separation retain ample high-frequency details, which significantly improves precision but noticeably decreases recall. This is likely due to the over-reconstruction caused by the abundance of high-frequency details, which results in minor visual tampering traces being over-reconstructed

Table 1. Ablation study on multi-view features extractor.

Input Feature	FCD				SCD			
	AUC	P	R	F	AUC	P	R	F
RGB (14×14)	71.19	24.99	27.23	26.07	79.14	19.34	29.35	23.32
RGB (28×28)	75.89	34.55	36.96	35.71	82.68	29.15	29.25	29.20
RGB + raw DCT	76.48	52.86	33.27	40.84	83.58	36.85	26.98	31.15
RGB + volume DCT	79.52	55.97	35.21	43.23	85.12	34.29	33.95	34.12

and thus some anomalies being missed. Among the different processing methods, the DCT volume processing we used is the most suitable frequency feature, offering a balanced representation of compression artifacts and high-frequency details.

Random Rectangular Masking (RRM): For feature reconstruction, the masking strategy efficiently prevents information leakage, improving global comprehension of the model. As shown in Table 2, we evaluate the effectiveness of our RRM strategy by comparing it with Random Single Mask (RSM), Random Block Mask (RBM), and Neighbor Masked Attention (NMA) proposed in UniAD [37], which masks the neighbor tokens when calculating the attention map in the encoder and decoder. We set an equal masked probability of each feature point in RSM, RBM, and RRM, while RSM uses random single point mask embedding as illustrated in Fig(a), RBM employs a fixed square block size of 7×7.

Table 2. Ablation study on different masking strategies.

Mask Strategy	FCD				SCD			
	AUC	P	R	F	AUC	P	R	F
w/o Mask	76.65	83.85	23.09	36.21	82.26	41.16	23.78	30.15
RSM	75.76	50.94	32.86	39.95	81.68	34.35	27.08	30.29
RBM	75.61	47.42	34.89	40.19	84.81	33.61	30.09	31.75
NMA	76.73	69.83	27.78	39.75	81.78	35.83	27.29	30.98
RRM	79.25	55.97	35.21	43.23	85.12	34.29	33.95	34.12

Without any masking, recall decreases severely, and reconstruction losses becoming extremely small, indicating serious information leakage. Similarly, RSM is insufficient as the recovery can be completed by copying neighboring regions. Compared to the fixed block size of RBM and NMA, our RRM based on the aspect ratio of tampered text regions, creates mask rectangles with varying positions and shapes to avoid excessive information loss and enhance the robustness of our model.

4.3 Comparison with Other Methods

To the best of our knowledge, we propose the first unsupervised deep learning method for the document image tampering localization task. As reference methods, we display supervised approaches, Mantra-Net [35], MVSS-Net [10], BEiT [1], CAT-Net [19] and DTD [25], which are all trained on 12,000 annotated 512×512 images for tampering localization. Furthermore, we compare our method with unsupervised reconstruction-based anomaly localization methods UniAD [37], MSTAD [17], and HVQ-Trans [23]. Table 3 presents the evaluation results, demonstrating the superiority of our proposed framework over other unsupervised reconstruction-based methods. Specifically, the lower performance on the SCD dataset can be attributed to its high proportion of Chinese invoice images. Chinese characters are notably more challenging to reconstruct compared to English letters, and the sparse textual distribution in invoices limits the availability of authentic document features for analysis.

Table 3. Comparison with other methods.

	Method	FCD			SCD		
		P	R	F	P	R	F
Sup.	Mantra-Net [35]	17.50	26.10	20.90	12.40	21.80	15.70
	MVSS-Net [10]	48.00	38.10	42.40	47.80	36.60	41.40
	BEiT-Uper [1]	55.00	43.60	48.70	40.80	39.50	40.20
	CAT-Net [19]	64.40	48.40	55.30	64.50	61.80	63.10
	DTD [25]	84.90	78.60	81.60	74.50	76.20	75.40
Unsup.	UniAD [37]	28.71	24.72	26.56	27.15	27.25	27.20
	MSTAD [17]	37.82	42.19	39.89	24.32	33.80	28.29
	HVQ-Trans [23]	14.35	36.88	20.66	15.24	18.05	16.58
	Ours	55.97	35.21	43.23	34.29	33.95	34.12

4.4 Visualization

In Fig. 3, we showcase anomaly score maps for tampering localization on the FCD and SCD. As we can see in the first four columns, our approach effectively identifies Generation tampering where foreground font styles exhibit visible inconsistencies. Analyzing failure cases, we observe instances of imperceptible Copy-Move tampering within the image (last column of Fig. 3), yet reconstructed based on visual consistency. Balancing accurate reconstruction of genuine regions in complex backgrounds while retaining subtle anomaly traces remains a challenge in unsupervised networks for document tampering localization. This also highlights the significance of extracting multi-modal information in this field,

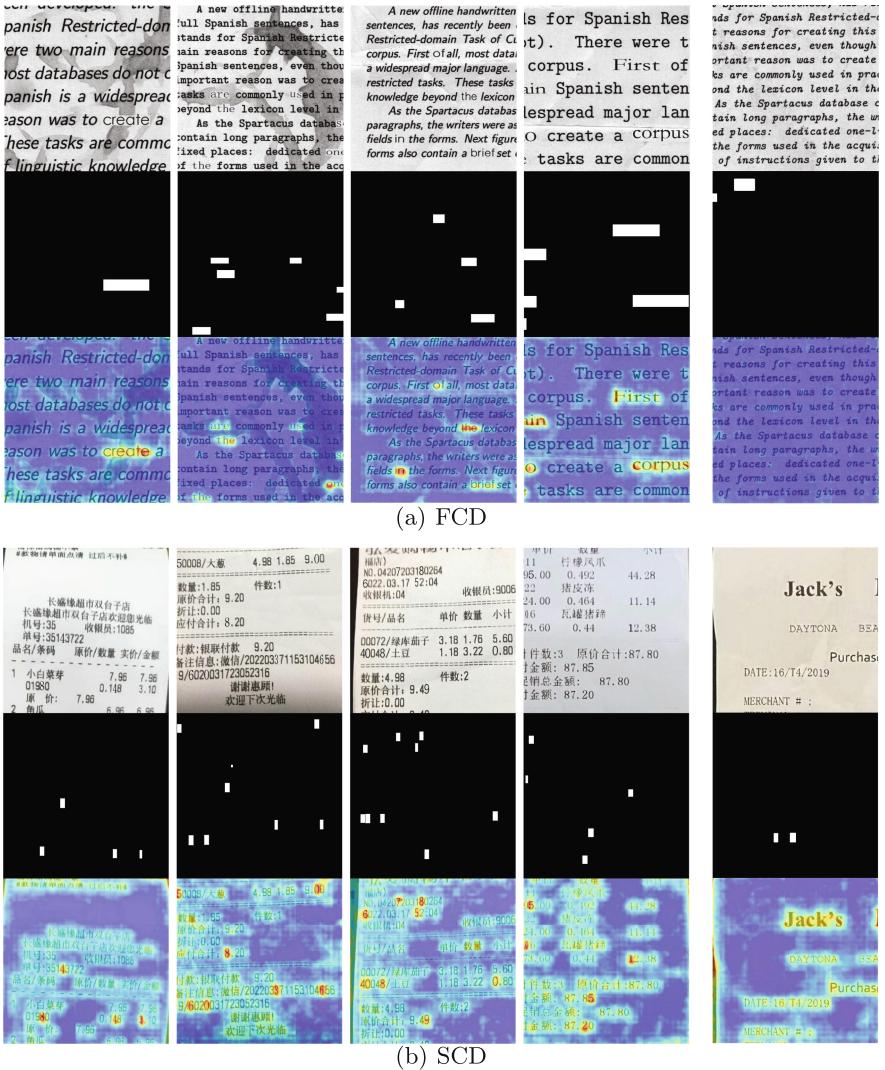


Fig. 3. Tampering localization results on the FCD/SCD datasets. Display the tampered samples, ground-truth, and tampering localization heat-maps (from top to bottom), and the last column shows failure Copy-Move cases.

where visual domain alone is inadequate for localizing various tampering operations. Our approach, which integrates a multi-view feature extractor with auxiliary frequency domain information, still has a lot of room for improvement.

5 Conclusion

In this paper, we present an unsupervised network for document image tampering localization, utilizing anomaly detection to learn normal features from authentic images and detect anomalies through high reconstruction errors in tampered images. Our method incorporates multi-view feature learning, fusing visual and frequency domain information, and employs a random rectangular mask strategy to enhance global feature understanding and prevent information leakage. As the first unsupervised approach evaluated on DocTamper, our work opens new avenues for unsupervised text tampering localization. Future research could explore semi-supervised learning with limited annotations and integrate edge enhancement and noise residue features to further improve localization accuracy.

Acknowledgments. This work is supported by the National Key Research and Development Program Grant 2020AAA0109700, and the Natural Science Foundation of China (NSFC) Grant U23B2029 and No. 62276258.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022)
2. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans. Inf. Forensics Secur.* **13**(11), 2691–2706 (2018)
3. Bertrand, R., Terrades, O.R., Gomez-Krämer, P., Franco, P., Ogier, J.M.: A conditional random field model for font forgery detection. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 576–580 (2015)
4. Chen, T., Li, B., Zeng, J.: Learning traces by yourself: blind image forgery localization via anomaly detection with VIT-VAE. *IEEE Signal Process. Lett.* **30**, 150–154 (2023)
5. Cozzolino, D., Poggi, G., Verdoliva, L.: Splicebuster: A new blind image splicing detector. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2015)
6. Cozzolino, D., Verdoliva, L.: NoisePrint: a CNN-based camera model fingerprint. *IEEE Trans. Inf. Forensics Secur.* **15**, 144–159 (2020)
7. Cruz, F., Sidère, N., Coustaty, M., D'Andecy, V.P., Ogier, J.M.: Local binary patterns for document forgery detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 01, pp. 1223–1228 (2017)
8. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019)
10. Dong, C., Chen, X., Hu, R., Cao, J., Li, X.: MVSS-Net: multi-view multi-scale supervised networks for image manipulation detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(3), 3539–3553 (2023)

11. Dong, L., Liang, W., Wang, R.: Robust text image tampering localization via forgery traces enhancement and multiscale attention. *IEEE Trans. Consum. Electron.* **70**(1), 3495–3507 (2024)
12. Guillaro, F., Cozzolino, D., Sud, A., Dufour, N., Verdoliva, L.: TruFor: leveraging all-round clues for trustworthy image forgery detection and localization. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20606–20615 (2022)
13. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15979–15988 (2022)
14. Iakovidou, C., Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: Content-aware detection of jpeg grid inconsistencies for intuitive image forensics. *J. Vis. Commun. Image Represent.* **54**, 155–170 (2018)
15. Jain, H., Gupta, G., Joshi, S., Khanna, N.: Passive classification of source printer using text-line-level geometric distortion signatures from scanned images of printed documents. *Multimedia Tools Appl.* **79**, 7377–7400 (2017)
16. James, H., Gupta, O., Raviv, D.: Learning document graphs with attention for image manipulation detection. In: International Conferences on Pattern Recognition and Artificial Intelligence (2022)
17. Kang, B., Zhong, Y., Sun, Z., Deng, L., Wang, M., Zhang, J.: MSTAD: a masked subspace-like transformer for multi-class anomaly detection. *Knowl. Based Syst.* **283**, 111186 (2024)
18. Kwon, M.J., Nam, S.H., Yu, I.J., Lee, H.K., Kim, C.: Learning JPEG compression artifacts for image manipulation detection and localization. *Int. J. Comput. Vis.* **130** (2022)
19. Kwon, M.J., Yu, I.J., Nam, S.H., Lee, H.K.: CAT-Net: compression artifact tracing network for detection and localization of image splicing. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 375–384 (2021)
20. Li, S., Ma, W., Guo, J., Xu, S., Li, B., Zhang, X.: UnionFormer: unified-learning transformer with multi-view representation for image manipulation detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12523–12533 (2024)
21. Liang, W., Dong, L., Wang, R., Yan, D., Li, Y.: Robust document image forgery localization against image blending. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 810–817 (2022)
22. Liao, X., Chen, S., Chen, J., Wang, T., Li, X.: CTP-Net: character texture perception network for document image forgery localization (2023)
23. Lu, R., et al.: Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. ArXiv abs/2310.14228 (2023)
24. Okamoto, Y., Genki, O., Yahiro, I., Hasegawa, R., Zhu, P., Kataoka, H.: Image generation and learning strategy for deep document forgery detection. ArXiv abs/2311.03650 (2023)
25. Qu, C., et al.: Towards robust tampered text detection in document image: new dataset and new solution. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5937–5946 (2023)
26. Shang, S., Memon, N., Kong, X.: Detecting documents forged by printing and copying. *EURASIP J. Adv. Signal Process.* **2014**(1), 1–13 (2014). <https://doi.org/10.1186/1687-6180-2014-140>

27. Shao, H., Huang, K., Wang, W., Huang, X., Wang, Q.: Progressive supervision for tampering localization in document images. In: International Conference on Neural Information Processing (2023)
28. Sun, K., Cao, G., Zhao, Q., Zhang, J.: Differential abnormality-based tampering detection in digital document images. In: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), pp. 145–149 (2019)
29. Sun, Y., Ni, R., Zhao, Y.: MFAN: multi-level features attention network for fake certificate image detection. *Entropy* **24**(1) (2022)
30. Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. ArXiv abs/1905.11946 (2019)
31. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010 (2017)
32. Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S.N., Jiang, Y.G.: Objectformer for image manipulation detection and localization. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2354–2363 (2022)
33. Wang, Q., Zhang, R.: Double JPEG compression forensics based on a convolutional neural network. *EURASIP J. Inf. Secur.* **2016** (2016)
34. Wang, Y., Zhang, B., Xie, H., Zhang, Y.: Tampered text detection via RGB and frequency relationship modeling. *Chinese J. Netw. Inf. Secur.* **8**(3), 29–40 (2022)
35. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-Net: manipulation tracing network for detection and localization of image forgeries with anomalous features. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9535–9544 (2019)
36. Xu, W., et al.: Document images forgery localization using a two-stream network. *Int. J. Intell. Syst.* **37**(8), 5272–5289 (2022)
37. You, Z., et al.: A unified model for multi-class anomaly detection. In: Advances in Neural Information Processing Systems, vol. 35, pp. 4571–4584 (2022)
38. Yu, Z., Li, B., Lin, Y., Zeng, J., Zeng, J.: Learning to locate the text forgery in smartphone screenshots. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023)
39. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1053–1061 (2018)



3VNet: Topological-Structure Driven Triple-V Network for Retinal Vessel Segmentation

Wei Zhou¹ , Xiaorui Wang¹ , Bin Zhou², and Yugen Yi²

¹ Shenyang Aerospace University, Shenyang 110136, China
wei.zhou@sau.edu.cn

² Jiangxi Normal University, Nanchang 330022, China
yiyg510@jxnu.edu.cn

Abstract. Automatic segmentation of retinal vessels plays a crucial role in the diagnosis of fundus diseases. Despite significant advancements made by U-shaped networks, they face two main limitations. First, they struggle to capture fine details and contextual information in raw images, especially in regions with poor image quality and contrast, leading to challenges in differentiating between vessels and background. Second, their dependence on traditional convolution with fixed sizes and shapes limits their adaptability to vessels with varying forms, particularly slender and tortuous local features. To address these limitations, we introduce 3VNet, comprising a 3V-shaped framework driven by skeleton topology. This architecture employs a skeleton supervision mechanism to capture and map vascular structures, refining features progressively through a cascaded model with skip connections. To comprehensively extract intricate vessel topological details such as bends and bifurcations, we developed the Multi-View Deformable Vessel-shape Extract Unit (MDVEU). This unit is seamlessly integrated into both the encoding and decoding stages of 3VNet, dynamically capturing vessel topology from multiple perspectives and employing adaptive fusion techniques for precise and detailed feature utilization. Extensive experiments on DRIVE, STARE, and CHASE_DB1 reveals that our method achieves higher accuracy than current approaches, surpassing the state-of-the-art (SOTA) on all datasets. Sensitivity exceeds SOTA by 1.71% and 1.30% on the DRIVE and CHASE_DB1 datasets, respectively. The source code is released at <https://github.com/wangwxr/3VNET>.

Keywords: Retinal Vessels Segmentation · Triple-V Network · Deformable Vessel-shape Convolution · Skeleton Supervision

1 Introduction

Vessel segmentation is essential for diagnosing ocular diseases such as glaucoma and diabetic retinopathy [1–4], playing a vital role in early detection and treatment. Clinically, specialists perform vessel segmentation manually, yet this approach is challenged by vessel irregularities, poor contrast, and blurred boundaries, which are further compounded by equipment and tissue variability [3, 4]. Moreover, the manual process is labor-intensive, slow, and susceptible to errors by humans [5–7]. The limitations of

manual segmentation have spurred the development of automated retinal vessel segmentation techniques, broadly categorized into two main approaches [8]: methods based on machine learning and those based on deep learning. Traditional machine learning techniques relied on handcrafted features and classical classifiers, which required extensive expert knowledge and frequently struggled with the complexity and variability of retinal images [9–12]. In contrast, methods based on Convolutional Neural Networks (CNNs) have shown superior performance by directly learning vessel features from original images. Recently, the U-Net architecture [13], a type of CNN model, has gained significant recognition in medical image segmentation tasks.

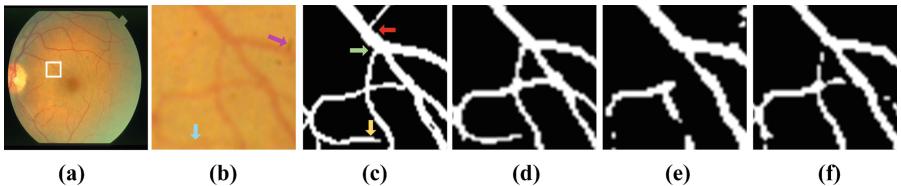


Fig. 1. Challenges in retinal vessel segmentation. (a) Fundus image, (b) Image patch, (c) Label, (d) 3VNet (Ours), (e) MMDC-Net [21] and (f) IterNet [17].

Inspired by its advantages, researchers have proposed U-Net variants for vessel segmentation. In 2017, Feng et al. [14] introduced a patch-based network, leveraging entropy to expedite training and compute vessel pixel proportions. In 2018, Oliveira et al. [15] devised a method proficient in managing changes in the direction and width of vessel formations by combining the multi-scale full CNN with the stationary wavelet transform analysis functions. In 2019, Alom et al. [16] developed the RRU-Net to enhance segmentation capability. In 2020, Li et al. [17] proposed IterNet, an iterative FCN with skip connections, designed to improve blood vessel segmentation accuracy using fewer manually labeled samples. Yuan et al. [18] developed an ACA-MLA-D-Unet in 2021, improving the segmentation accuracy by integrating a multi-level attention mechanism into network. In 2022, Chowdhury et al. [31] developed a method for artery-vein segmentation called MSGANet-RAV, which utilized a multiscale guided attention network to effectively perform both artery-vein segmentation and classification. In 2023, Liu et al. [19] proposed the ResDO-conv network, designed to extract robust contextual features. Additionally, Ding et al. [20] proposed RCAR-Unet in 2024, integrating sophisticated feature learning techniques with rough sets and rough neurons to effectively address uncertainties.

Despite the effectiveness of various U-Net variants, they struggle with two main challenges. **First**, their direct application of U-shaped networks on raw images for blood vessel extraction may overlook fine vessel structures due to their limited capacity to capture intricate details and contextual information, particularly given the minimal visual contrast between blood vessels and the background, as illustrated in Fig. 1b. Additionally, noise interference, highlighted by the purple arrows in Fig. 1b, further compounds this challenge. **Second**, the reliance of most existing methods on fixed-shape convolution kernels is a significant limitation, particularly for vessels with varying sizes, shapes, and orientations. This challenge is exemplified in Fig. 1c, where blood vessels

of varying dimensions (red and yellow arrows) and contorted vessels (green arrows) present segmentation difficulties. Consequently, these methodologies often result in vessel discontinuities and missed detections, as depicted in Fig. 1e–f.

To address these challenges, we introduce a novel vessel segmentation network named 3VNet, which incorporates three cascaded V-shaped subnetworks with advanced encoder-decoder structures designed to enhance feature extraction and adapt to the complex topological structure of retinal vessels. (1) To improve the ability to identify the little differences between vessels and background, we introduce a 3V-shaped framework driven by skeleton topology, progressively refine blood vessel details from topology structure features instead of raw image input. (2) To address the challenge of adapting fixed conventional convolutions to flexible vessel topology structures, we introduce the Multi-View Deformable Vessel-shape Extract Unit (MDVEU), comprising the Multi-View Deformable Vessel-shape Convolution (MDVC) and Adaptive Fusion Block (AFB). This unit adjusts convolution kernels to varying vessel scales and morphologies from multiple viewpoints, while enhancing segmentation accuracy and robustness through adaptive fusion. The main contributions of this work are below:

1. We propose 3VNet, comprising three cascaded V-shaped subnetworks with advanced encoder-decoder structures driven by skeleton topology to progressively refine morphology of retinal vessels.
2. We design MDVEU, integrating MDVC and AFB, adjusting convolution kernels for different vessel morphologies while enhancing segmentation precision through adaptive fusion.
3. Extensive experiments confirm that our approach surpasses SOTA techniques.

2 Method

This study proposed a 3VNet for fundus vessel segmentation, as depicted in Fig. 2. 3VNet comprises a 3V-shaped framework and Multi-View Deformable Vessel-shape Extract Unit (MDVEU) serving as the first-stage encoder and final-stage decoder. Firstly, we outline the structure of the network 3V-shaped framework, followed by an introduction to the MDVEU.

2.1 3V-Shaped Framework

Traditional U-shaped networks often struggle to accurately extract vessel characteristics and adapt to complex scenes when processing raw images directly due to their limited capacity to capture fine details and diverse shape. To address this limitation, this study focuses on extracting blood vessel details from topological feature maps instead of raw image inputs and proposes a 3V-shaped framework. The framework incorporates a cascading approach with a distinct supervision strategy across its stages, designed to progressively refine the segmentation output.

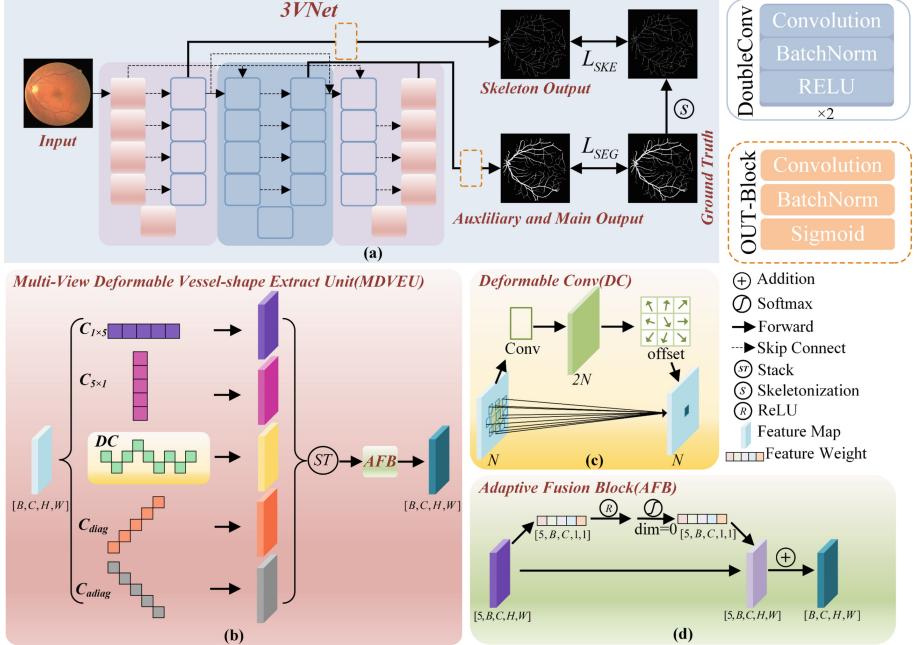


Fig. 2. Overview of the proposed 3VNet.

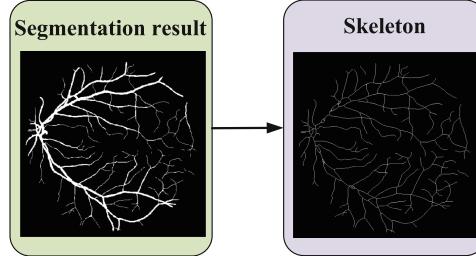


Fig. 3. Convert the segmentation result into skeleton.

As shown in Fig. 2a, the fundus image undergoes processing through a U-shaped network equipped with skeleton supervision mechanisms. The primary objective is to delineate the vascular skeleton map, which serves a dual purpose: encoding the connectivity and topology of retinal vasculature and facilitating finer feature extraction in later stages. Skip connections transmit the rich skeleton topology information from the last feature maps of the initial V-shaped stage to subsequent V-shaped nets, acting as a detailed vascular topology reference. This information aids in progressively enhancing segmentation. By using skeleton features as a reference as shown in Fig. 3, the network can focus on capturing the full vessel profile, including finer vessels that are typically challenging to segment because of their low contrast and proximity to other retinal structures. This principle extends to the second V-shaped network, where the final feature map

is relayed to the ultimate network. In this framework, the output of the initial V-shaped structure is termed the Skeleton Output. Subsequently, the outputs from the second and final V-shaped structures are designated as the Auxiliary and Main outputs, respectively.

As the network advances through the stages, it transitions from a broad, topological view to a precise, pixel-level segmentation. This is achieved by adjusting the focus from the generalized vessel tree to the specific vessel boundaries. The loss functions, L_{SKE} in the initial stage and L_{SEG-i} ($i = 2, 3$) in the subsequent stages, are designed to complement each other, ensuring a harmonious optimization that reinforces both the macro-structures and the micro-details of the vasculature, defined as:

$$L_{SKE} = 1 - \frac{2|Y_{SKE} \cap \hat{Y}_{SKE}|}{|Y_{SKE}| + |\hat{Y}_{SKE}|} \quad (1)$$

$$L_{SEG-i} = 1 - \frac{2|Y \cap \hat{Y}_i|}{|Y| + |\hat{Y}_i|} \quad (2)$$

where Y represents the ground truth and \hat{Y}_i represents the output of the i -th stage. Then, they are summed up with certain weights as:

$$L_{Total} = L_{SKE} + \alpha L_{SEG-2} + \beta L_{SEG-3} \quad (3)$$

where α and β are hyperparameters that determine the significance of each sub-network in the overall output.

In essence, our 3V-shaped framework embodies a sophisticated information flow, where the early extraction of skeletal features sets the stage for detailed segmentation work. Each successive network module in the cascade receives enriched feature maps via skip connections, ensuring that the refinement process is deeply rooted in the initial skeletal segmentation. Our approach not only provides a thorough representation of the vessels but also enhances the network's capability to refine its predictions, resulting in a highly accurate segmentation output.

2.2 Multi-view Deformable Vessel-Shape Extract Unit (MDVEU)

The convolutional kernels of fixed size, typically $N \times N$, are commonly employed in most CNNs, and have demonstrated considerable success in tasks involving natural scenes [13]. However, the retinal vessels are characteristically slender, elongated, and frequently continuous. Therefore, utilizing a larger $N \times N$ convolutional kernel may inadvertently encompass numerous irrelevant pixels, reducing the model's effectiveness in focusing on relevant vascular details. Conversely, employing a smaller kernel constrains the receptive field, rendering it insufficient for capturing the continuity of vascular structures.

Building upon these observations, we have devised the Multi-View Deformable Vessel-shape Extract Unit (MDVEU), comprising the Multi-View Deformable Vessel-shape Convolution (MDVC) and the Adaptive Fusion Block (AFB). Serving as units both in encoder and decoder, the MDVEU dynamically extracts and integrates intricate vessel features, as illustrated in Fig. 2b. In the MDVEU, a unit receives an input feature map with dimensions $[B, C, H, W]$ and processes it to generate an output of the same dimensions. Here, B denotes the batch size, C is the number of channels, while H, W are the image's height and width, respectively.

Multi-view Deformable Vessel-shape Convolution (MDVC). It consists of five individual convolutions, comprising four narrow with fixed-shape convolutions and one deformable convolution, as shown in Fig. 2b. The fixed convolutions operate at kernel scales of 1×5 and 5×1 , along with five consecutive pixels on both the main and sub-diagonals. The 1×5 and 5×1 kernels are executed using standard convolution methods. The diagonal kernel is implemented by zeroing out the irrelevant areas of the 5×5 kernel and locking the gradient. The traditional convolution operation is composed of two main elements, sampling through a structured grid and aggregating the sampled values by weight, which can be formulated as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \quad (4)$$

In this context, y denotes the output feature map at p_0 location and p_n represents each position in the regular grid R . Besides, w and x denote the weight and feature value, respectively.

To allow the convolution kernel greater adaptability in concentrating on the complex geometric characteristics of the vessel, inspired by [21, 22], we introduce deformable convolution in this unit, as shown in Fig. 2c. In deformable convolution, the standard grid R is expanded by adding offsets Δp_n , along both x and y axes, followed by interpolation to attain a deformable convolution. For further details on deformable convolution, please refer to [21]. Consequently, Eq. (4) is transformed into:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \quad (5)$$

Based on the provided descriptions, five feature maps (F_1, F_2, F_3, F_4, F_5) from different convolution operations in MDVC are as follows:

$$F_1 = C_{1 \times 5} \times X \quad (6)$$

$$F_2 = C_{5 \times 1} \times X \quad (7)$$

$$F_3 = C_{diag} \times X \quad (8)$$

$$F_4 = C_{adiag} \times X \quad (9)$$

$$F_5 = C_{def}(X, \Delta) \quad (10)$$

In this context, X represents the input feature map, the $C_{1 \times 5}$, $C_{5 \times 1}$, C_{diag} , C_{adiag} and C_{def} represent the horizontal 1×5 convolution kernel, the vertical 5×1 convolution kernel, the convolution kernel along the main diagonal, the convolution kernel along the anti-diagonal, and the deformable convolution kernel, respectively. Δ represents the learned offsets in the deformable convolution.

By integrating the multidirectional fixed convolution kernel with the deformable convolution, our MDVC module proficiently captures the intricate topological features of vascular structures. This enhancement significantly bolsters our model's capability to

segment blood vessels with complex geometries. When feature maps from N channels are input to MDVC, they are duplicated and processed through these five parallel kernels. The results are then merged, forming $5N$ feature maps, which are subsequently input into the Adaptive Fusion Block (AFB).

Adaptive Fusion Block (AFB). The features processed by the MDVC may contain irrelevant information due to the complex nature of vascular structures, limited relevance of some features to the segmentation task, potential redundancy in extracted features. Therefore, we enable the network to perform feature recalibration via the AFB, as depicted in Fig. 2d. This block allows the network to fully leverage the available feature information, selectively accentuating information-rich features while diminishing the impact of less pertinent ones.

In the AFB, we implement a dynamic weight updating strategy for $[5, B, C, 1, 1]$ continuously adjusting them according to the feature maps, as depicted in Fig. 2d. In this $[5, B, C, 1, 1]$ weight, each feature from the MDVC has its own weight. Following ReLU activation, we ensure that the weights remain non-negative to preserve information integrity. The activated weights are subsequently normalized using the Softmax function, which produces a final weight distribution $[5, B, C, H, W]$. This distribution is utilized to compute a weighted sum across all channels and batches, getting the final output with $[B, C, H, W]$. By applying attention at both the feature and channel levels, this process enhances integration, effectively improving the network’s representation of the input data. The steps involved in the computational process are summarized as follows:

$$O = \sum_{k=1}^5 \alpha_k \cdot \text{RELU}(F_k) \quad (11)$$

where O is the output of AFB, and α_k are the weights computed via the following equation:

$$\alpha_k = \frac{\exp(s_k)}{\sum_{j=1}^5 \exp(s_j)} \quad (12)$$

where s_k is a learnable parameter indicating the importance of the k -th feature map.

In summary, we seamlessly incorporate the MDVEU into both the encoding and decoding phases of our 3VNet. During encoding, the MDVEU dynamically adapts its shape to match the vessel structures, enhancing the extraction of critical topological features. In the decoding phase, it further refines these features to ensure precise segmentation, with a particular emphasis on accurately delineating challenging regions characterized by twists and forks in vessel structures.

3 Experiments

3.1 Datasets and Evaluation Metric

To assess the efficacy of our approaches, we carried out extensive experiments on three widely available datasets specifically curated for the task of retinal vessel segmentation, i.e., DRIVE [23], STARE [24], and CHASE_DB1 [25]. The DRIVE consists of 40 pairs

of fundus color images (565×584), each paired with corresponding vessel segmentation labels. This dataset is evenly divided into equal training and testing subsets. The STARE contains 20 color fundus photographs (700×605) without predefined training or test sets, researchers often randomly select 10 for training or apply Leave-One-Out training method. The CHASE_DB1 contains 28 color retinal images, encompassing both right and left eyes. According to prior research protocols, the first 20 images are set aside for training, with the rest designated for testing. The DRIVE includes fields of views(FoVs) for each image, enhancing its ease of use. However, the STARE and CHASE_DB1 datasets do not provide FoV details. Therefore, for this research, we utilized the masks of FoV created by Marin et al. [11]. The metrics were calculated using only the pixels contained within these masks. We evaluate the segmentation performance using the widely used Accuracy (Acc , %), Specificity (Sp , %) and Sensitivity (Se , %) are defined as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

$$Se = \frac{TP}{TP + FN} \quad (14)$$

$$Sp = \frac{TN}{FP + TN} \quad (15)$$

In this context, TP denotes the accurately segmented counts of vascular pixels. The TN denotes the accurately segmented counts of background pixels. The FP indicates cases where mistaking background pixels for vascular, and FN reflects the misclassification of vascular pixels as background. The metric Acc is employed to evaluate the efficacy of segmentation, with higher Acc values indicating superior performance. Additionally, the metrics Se and Sp are utilized to gauge the precision of segmentation for TP and TN pixels, respectively.

3.2 Implementation Setup

The 3VNet is implemented on an NVIDIA 4090 GPU with 24 GB of memory and implemented utilizing the PyTorch 2.0.1 framework. We feed the whole image into the network instead of dividing it into smaller patches. Each input is resized to 512×512 and augmented by adding Gaussian noise, flipping, and rotating. The Adam optimizer is used during the training phase, with initial learning rate $lr_0 = 0.001$ and beta values set to $(0.9, 0.99)$. The learning rate reduces progressively based on a polynomial decay formula, defined as $lr = lr_0 \times (1 - t/T)^{0.9}$, where $T = 200$ stands for the total number of epochs, and t is the current epoch. We use a batch size of 2 for training. The parameters α and β are set empirically to values of 1 and 3, respectively.

3.3 Experimental Results

To confirm the efficacy of the 3VNet, we conducted a comprehensive comparison experiments with current SOTA techniques using the DRIVE, STARE and CHASE_DB1 datasets and the comparative results are presented in Table 1.

The performance metrics in Table 1 highlight that our 3VNet demonstrates superior results across various evaluation metrics. The table clearly indicates that our method achieves the highest *Acc* value across all three datasets. This superior performance underscores our model’s ability to adapt to various complex scenarios and distinguish between foreground and background more effectively than competing methods. Specially, the *Se* value of our method exceeds SOTA by 1.3% on the CHASE_DB1 dataset and performs well on the other two datasets. This performance indicates that our model is highly capable of extracting blood vessels features and offers superior adaptation to various shape of blood vessels than other approaches.

Table 1. Various method comparisons in retinal vessel segmentation. The “–” indicating that a particular metric was not reported. The metrics of our methodology, as well as the best metrics, are highlighted **in bold**.

Methods	DRIVE			STARE			CHASE_DB1		
	<i>Acc</i>	<i>Se</i>	<i>Sp</i>	<i>Acc</i>	<i>Se</i>	<i>Sp</i>	<i>Acc</i>	<i>Se</i>	<i>Sp</i>
IterNet [17]	95.73	77.35	98.38	97.01	77.15	98.86	96.55	79.70	98.23
MMDC-Net [26]	96.07	80.74	97.55	95.91	85.09	96.89	96.46	84.40	97.28
Zhai et al. [27]	95.71	79.82	98.18	96.11	78.45	98.48	–	–	–
CSGNet [28]	95.76	79.43	98.14	96.92	82.98	98.55	96.81	79.47	98.55
TiM-Net [29]	96.38	78.05	98.16	97.11	78.67	98.80	97.11	76.97	98.65
Shao et al. [30]	–	78.50	98.21	–	78.45	97.10	–	80.57	98.24
ResDO-Unet [19]	95.61	79.85	97.91	95.67	79.63	97.92	96.72	80.20	97.94
RCAR-UNet [20]	95.37	74.87	98.36	95.94	69.79	99.05	95.66	74.75	97.98
Ours	96.74	82.61	98.10	97.39	82.71	98.60	97.52	85.79	98.31

Figure 4 visually compares the segmentation maps produced by our network with those from IterNet [17], MMDC-Net [26], CSGNet [28], and Tim-Net [29], against the Ground Truth (“GT”). The first and second rows illustrate our methodology’s superior performance in segmenting vessels with diverse shapes and configurations, particularly in cases of vessel intersection. This superiority arises from our model’s adaptability to the varied topological structures of blood vessels. Additionally, our model outperforms competing methods, especially in scenarios with low contrast where blood vessels blend with the background, as illustrated in the third row of Fig. 4. This highlights the model’s robust capability to accurately discern and delineate blood vessel features. These evaluations, both quantitative metrics and visual assessments, affirm the effectiveness of our network in generating segmentation results closely aligned with the GT. These results provide further proof and validation for the effectiveness of our proposed approach.

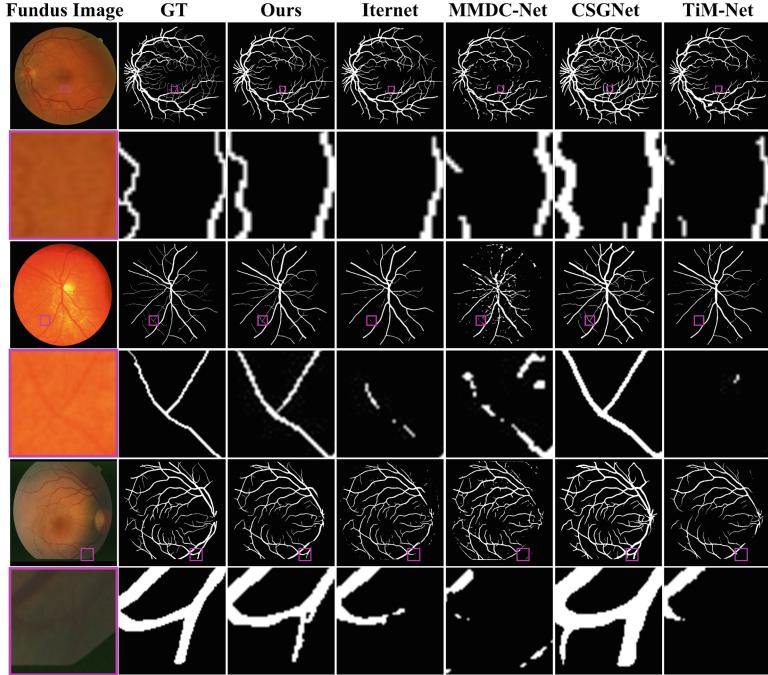


Fig. 4. Visual results from different networks on the DRIVE, CHASE_DB1, and STARE datasets.

3.4 Ablation Experiments

To rigorously assess the contribution of each component within the proposed 3VNet for segmenting retinal vessels, we performed a series of comprehensive ablation experiments using the DRIVE, STARE, and CHASE_DB1 datasets. 3VNet comprises four major components, including skeleton supervision (Skeleton), Adaptive Fusion Block (AFB), Multi-View Deformable Vessel-shape Extract (MDVEU) and 3V-shaped framework (3V). We conducted evaluations by systematically excluding specific components: models without skeleton supervision (“w/o Skeleton”), models without the Adaptive Fusion Block (“w/o AFB”), models without the Multi-View Deformable Vessel-shape Extract Unit (“w/o MDVEU”), and models without the 3V-shaped framework (“w/o 3V”). Table 2 illustrates the contribution of each component to performance enhancement, while Fig. 5 visually represents the corresponding segmentation results. These results highlight a significant decline in the network’s performance when skeleton supervision is omitted, leading to reductions in both *Acc* and *Se* and resulting in coarser vessel segmentation outcomes. This decline is attributed to the absence of skeleton topology information in the early stages, limiting the model’s ability to effectively delineate vessel morphology. Furthermore, removal of the MDVEU (“w/o MDVEU”) adversely impacts both *Acc* and *Se*, indicating challenges in accurately capturing vessel morphology and underscoring the pivotal role of MDVEU in comprehensively apprehending vessel topology. Similarly, exclusion of the AFB (“w/o AFB”) exacerbates performance degradation, particularly at vessel intersections, confirming the indispensable role of AFB in

suppressing extraneous features and enhancing the relevance of extracted features in vessel segmentation. Additionally, when the 3V-shaped framework is removed (“w/o 3V”), there is a substantial decrease in Se , Sp , and Acc in the CHASE_DB1 dataset, and reduced Se and Acc in the DRIVE and STARE datasets. Notably, the Se of the STARE dataset experiences a significant decrease of 5.28%, highlighting the crucial role of our 3V-shaped framework in enhancing network segmentation ability. Figure 5 further illustrates the disruption in vessel continuity, emphasizing the importance of our 3V-shaped framework in accurately extracting vessel structures. These evaluations underscore the critical role of each component and the significance of their collective integration in achieving optimal performance.

Table 2. Results of ablation experiments. The metrics are in %.

Methods	DRIVE			STARE			CHASE_DB1		
	Acc	Se	Sp	Acc	Se	Sp	Acc	Se	Sp
w/o Skeleton	96.72	82.32	98.10	97.34	80.28	98.76	97.49	85.63	98.29
w/o AFB	96.73	82.08	98.13	97.30	80.49	98.70	97.49	85.34	98.30
w/o MDVEU	96.73	81.89	98.16	97.32	80.54	98.71	97.51	85.80	98.30
w/o 3V	96.69	81.28	98.17	97.19	78.43	98.75	97.43	84.72	98.29
Ours	96.74	82.61	98.10	97.39	82.71	98.60	97.52	85.79	98.31

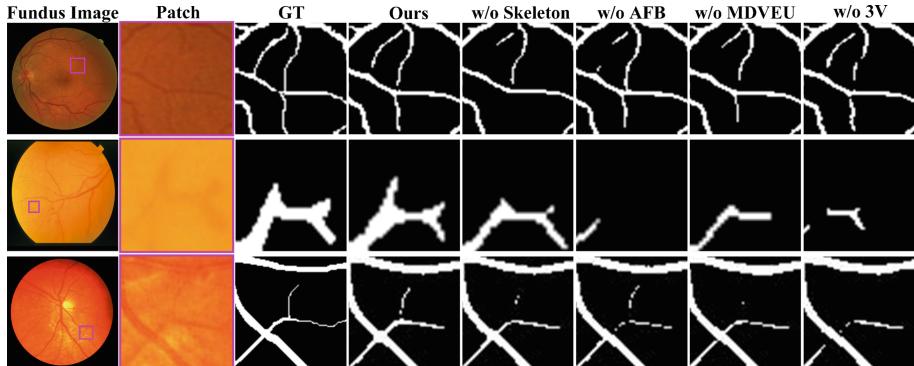


Fig. 5. Visual outcomes from the DRIVE, CHASE-DB1, and STARE datasets arranged for comparison from the top row to bottom row.

4 Conclusion

This paper introduces an innovative deep learning architecture tailored for retinal vessel segmentation named 3VNet, which addresses the significant challenges posed by the existing methods in accurately capturing the complex morphology of retinal vessels. By

integrating the Multi-View Deformable Vessel-shape Extract Unit (MDVEU) with cascaded 3V-shaped structures driven by skeleton topological structure, 3VNet dynamically captures and accurately segments the complex topological structures of vessels, particularly focusing on bends and bifurcations. Comprehensive experiments conducted on three public datasets, i.e., DRIVE, STARE, and CHASE_DB1, have demonstrated that 3VNet not only outperforms existing SOTA models but also shows robustness across different imaging conditions. The results affirm the effectiveness of our approach in dealing with the inherent challenges of retinal vessel segmentation, marking a notable advancement in the automatic analysis of fundus images. For future research, we plan to concentrate on enhancing the computational complexity and runtime performance of our 3VNet model. Moreover, we will make our model better to handle lighting or other real-world conditions like low contrast area, making it beneficial to the diagnosis of eye diseases.

Acknowledgement. This research is partially funded through grants provided by the National Natural Science Foundation of China (Nos. 62062040, 62102270, 62041702), the Jiangxi Province Key Subject Academic and Technical Leader Funding Project (No. 20212BCJ23017), the Outstanding Youth Project of Jiangxi Natural Science Foundation (No. 20212ACB212003), the project of Natural Science Foundation of Liaoning province (No. 2023-MS-246), and the Fundamental Research Funds for the Universities of Liaoning province (Nos. 20240211, z20240219).

References

1. Grélard, F., Baldacci, F., Vialard, A., Domenger, J.P.: New methods for the geometrical analysis of tubular organs. *Med. Image Anal.* **42**, 89–101 (2017)
2. Leipsic, J., et al.: SCCT guidelines for the interpretation and reporting of coronary CT angiography: a report of the Society of Cardiovascular Computed Tomography Guidelines Committee. *J. Cardiovasc. Comput. Tomogr.* **8**(5), 342–358 (2014)
3. Yau, J.W., Rogers, S.L., Kawasaki, R., Lamoureux, E.L., Kowalski, J.W., Bek, T., Meta-Analysis for Eye Disease (META-EYE) Study Group: Global prevalence and major risk factors of diabetic retinopathy. *Diabetes Care* **35**(3), 556–564 (2012)
4. Fraz, M.M., et al.: Blood vessel segmentation methodologies in retinal images—a survey. *Comput. Methods Programs Biomed.* **108**(1), 407–433 (2012)
5. Jin, Q., Meng, Z., Pham, T.D., Chen, Q., Wei, L., Su, R.: DUNet: a deformable network for retinal vessel segmentation. *Knowl.-Based Syst.* **178**, 149–162 (2019)
6. Ortega, M., Penedo, M.G., Rouco, J., Barreira, N., Carreira, M.J.: Personal verification based on extraction and characterisation of retinal feature points. *J. Vis. Lang. Comput.* **20**(2), 80–90 (2009)
7. Simon, C.: A new scientific method of identification. *N. Y. State J. Med.* **35**(18), 901–906 (1935)
8. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(1), 131–137 (2003)
9. Orlando, J.I., Blaschko, M.: Learning fully-connected CRFs for blood vessel segmentation in retinal images. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. MICCAI 2014. LNCS, vol. 8673. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10404-1_79

10. Ganjee, R., Azmi, R., Gholizadeh, B.: An improved retinal vessel segmentation method based on high level features for pathological images. *J. Med. Syst.* **38**, 1–9 (2014)
11. Marín, D., Aquino, A., Gegúndez-Arias, M.E., Bravo, J.M.: A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. *IEEE Trans. Med. Imaging* **30**(1), 146–158 (2010)
12. Lázár, I., Hajdu, A.: Segmentation of retinal vessels by means of directional response vector similarity and region growing. *Comput. Biol. Med.* **66**, 209–221 (2015)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. LNCS, vol. 9351. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
14. Feng, Z., Yang, J., Yao, L.: Patch-based fully convolutional neural network with skip connections for retinal blood vessel segmentation. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 1742–1746. IEEE (2017)
15. Oliveira, A., Pereira, S., Silva, C.A.: Retinal vessel segmentation based on fully convolutional neural networks. *Expert Syst. Appl.* **112**, 229–242 (2018)
16. Alom, M.Z., Yakopcic, C., Hasan, M., Taha, T.M., Asari, V.K.: Recurrent residual U-Net for medical image segmentation. *J. Med. Imaging* **6**(1), 014006 (2019)
17. Li, L., Verma, M., Nakashima, Y., Nagahara, H., Kawasaki, R.: Iternet: retinal image segmentation utilizing structural redundancy in vessel networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3656–3665 (2020)
18. Yuan, Y., Zhang, L., Wang, L., Huang, H.: Multi-level attention network for retinal vessel segmentation. *IEEE J. Biomed. Health Inform.* **26**(1), 312–323 (2021)
19. Liu, Y., Shen, J., Yang, L., Bian, G., Yu, H.: ResDO-UNet: a deep residual network for accurate retinal vessel segmentation from fundus images. *Biomed. Signal Process. Control* **79**, 104087 (2023)
20. Ding, W., et al.: RCAR-UNet: retinal vessel segmentation network algorithm via novel rough attention mechanism. *Inf. Sci.* **657**, 120007 (2024)
21. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (pp. 764–773) (2017)
22. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)
23. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **23**(4), 501–509 (2004)
24. Hoover, A.D., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med. Imaging* **19**(3), 203–210 (2000)
25. Fraz, M.M., et al.: An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Trans. Biomed. Eng.* **59**(9), 2538–2548 (2012)
26. Zhong, X., Zhang, H., Li, G., Ji, D.: Do you need sharpened details? Asking MMDC-Net: multi-layer multi-scale dilated convolution network for retinal vessel segmentation. *Comput. Biol. Med.* **150**, 106198 (2022)
27. Zhai, Z., Feng, S., Yao, L., Li, P.: Retinal vessel image segmentation algorithm based on encoder-decoder structure. *Multimedia Tools Appl.* **81**(23), 33361–33373 (2022)
28. Guo, S.: CSGNet: cascade semantic guided net for retinal vessel segmentation. *Biomed. Signal Process. Control* **78**, 103930 (2022)
29. Zhang, H., et al.: TiM-Net: transformer in M-Net for retinal vessel segmentation. *J. Healthcare Eng.* **2022**(1), 9016401 (2022)

30. Shao, H.C., et al.: Retina-transnet: a gradient-guided few-shot retinal vessel segmentation net. *IEEE J. Biomed. Health Inform.* (2023)
31. Chowdhury, A.E., Mann, G., Morgan, W.H., Vukmirovic, A., Mehnert, A., Sohel, F.: MSGANet-RAV: a multiscale guided attention network for artery-vein segmentation and classification from optic disc and retinal images. *J. Optometry* **15**, S58–S69 (2022)



Identification of Proton Exchange Membrane Fuel Cell Parameters Using a Parameterless Swarm Intelligent Algorithm

Pankaj Sharma¹, Rohit Salgotra^{2,3(✉)}, Sarvanakumar Raju¹,
Szymon Lukasik², and Amir H. Gandomi^{3,4}

¹ School of Electrical Engineering, Vellore Institute of Technology, Vellore, India

² Faculty of Physics and Applied Computer Science, AGH University of Krakow,
Krakow, Poland

rohits@agh.edu.pl

³ Data Science Institute, University of Technology Sydney, Sydney, NSW 2007,
Australia

⁴ Óbuda University, Budapest, Bécsi út 96/B, Budapest 1034, Hungary

Abstract. This work proposed a new hybrid parameterless optimization algorithm named the grey cuckoo differential (GCD) algorithm, using the components of the grey wolf optimizer (GWO), cuckoo search (CS), and differential evolution (DE). By integrating the strengths of these three algorithms, the hybrid model aims to obtain a balance between the exploration phase as well as exploitation phase, leading to greater convergence speed and a higher quality of the solution. The performance of GCD has been evaluated using the CEC 2017 and CEC 2019 benchmarks. In addition, three-proton exchange membrane fuel cell (PEMFC) stack parameter extraction experiments were performed to verify the performance and accuracy of the GCD algorithm. The results of the GCD algorithm are compared with the improved artificial humming bird algorithm (IHBO), young double-slit experiment (YDSE), subtraction average-based optimizer (SABO), artificial colony differential evolution optimizer (ABCDE), jDE100 (winner of CEC 2019 competition) and others to demonstrate its effectiveness. The GCD algorithm showed the lowest sum of square error (SSE) based on the comparison result. Finally, the mean absolute error (MAE), mean square error (MSE), root mean squared error (RMSE) and Friedman and Wilcoxon's statistical tests show the robustness of the GCD algorithm with respect to other competitive algorithms found in the literature.

Keywords: Multi-hybrid algorithms · CEC benchmarks · PEMFC · Numerical Optimization · Renewable energy Optimization

1 Introduction

Modern society faces a significant challenge in achieving the goal of reducing carbon emissions from the energy sector. Despite the projected increase in the

use of renewable energy to replace fossil fuels, many other measures are required to completely eliminate carbon emissions from the industry. Hydrogen energy is recognized as a highly promising alternative energy source, particularly for PEMFC, due to its numerous advantages [27]. These advantages include relatively low operating temperature, high efficiency, potential cost reduction, very low emissions, fast start-up, low corrosion rate, low noise, and others. As a result, fuel cells are considered an appropriate choice for various applications, including the residential, commercial, and industrial sectors [28].

In the past few years, there has been a significant increase in the focus on the modeling of PEMFCs. This research aims to create an extremely accurate model for PEMFCs that is appropriate for software simulations and accurately reflects experimental models. There are two main techniques to categorize the modeling of PEMFCs. The I^{st} approach utilizes mechanistic models to replicate mass transport, heat transfer, as well as electrochemical reactions. Another approach involves utilizing models that rely on semiempirical or empirical formulas, which are often derived from mechanistic principles. Each model has distinct mathematical formulations with unknown parameters that are not disclosed in the manufacturer's datasheets [18]. These parameters are essential for building a model that is both efficient and reliable.

The empirical modeling is based on semiempirical as well as nonlinear equations and employs the I-V polarization curve. Amphlette was the I^{st} to provide an empirical model based on experimental evidence [10, 31]. This empirical model enables exact parameter design with minimum error, potentially leading to increased performance and cost savings. It is critical to calculate the unknown parameters of the PEMFC with maximum precision. Due to the unknown parameters and the change according to the load condition, the stack presents nonlinear challenges. As a result, it is very difficult to make a model using the traditional method due to the more complex and time-consuming process [18].

Therefore, optimization algorithms (OAs) are commonly used by researchers to obtain the optimal parameter identification. The OAs are probabilistic in nature as well as generate solutions from a population for a specific problem [25]. Thus, optimization techniques demonstrate high efficiency in addressing complex engineering optimization challenges and can be effectively utilized in various applications. The estimation of PEMFC unknown parameters is a unique application that has been explored in the literature using various OAs, including Improved Heap-Based Optimizer (IHBO) [4], Imperialist Competitive Algorithm (ICA), Firefly Optimization Algorithm (FOA) [18], Improved Artificial Hummingbird Algorithm (IAHA) [3], Artificial Bee Colony DE Optimizer (ABCDE), Enhanced Lévy Flight Bat Algorithm (ELBA) [15], Adapted Sparrow Search Algorithm (ASSA) [43], Manta Rays Foraging Optimizer (MRFO) [27], Coyote Optimization Algorithm (COA) [36], as well as Whale Optimization Algorithm (WOA) [12]. Although many previous research studies have been conducted to address these challenges, there are shortcomings that must be addressed in terms of stability, accuracy, as well as effectiveness.

In this paper, a novel and efficient multi-hybrid MH algorithm known as the GCD algorithm is taken to solve the PEMFC stack parameter extraction challenges. The multi-hybrid algorithm (GCD) uses the search equations of GWO, CS, and DE to improve the exploration and the exploitation capabilities of the basic algorithms. In order to demonstrate the efficacy of the GCD algorithm, CEC 2017 and CEC 2019 benchmarks are used and a comparison is performed with algorithms such as BKA [39], NRBO [33], NMRA [26], NBA [41], HLOA [23], DE (jDE100) [9] (winner of CEC 2019), Young's Double Slit Experiment (YDSE) [2], Manis Pentadactyla Optimizer (MPO) [14] and others. For the PEMFC stack, a comparison is made with the Zebra Optimization Algorithm (ZOA) [37], YDSE [2], Sinh Cosh Optimizer (SCHO) [8], Subtraction Average Based Optimizer (SABO) [38], Harris Hawks Optimization (HHO) [16], GWO [22], and Exponential Distribution Optimizer (EDO) [1]. The main contributions of this article are presented as follows:

- A new hybrid OA named GCD, which uses the search equations of GWO, CS, and DE to improve the exploration and exploitation capabilities of existing optimization techniques.
- The proposed algorithm is parameterless and uses optimized equations for adaptation over the course of iterations.
- The GCD algorithm is evaluated utilizing CEC 2017 as well as CEC 2019 benchmark problems.
- The proposed algorithm is applied to solve three different PEMFC (Stack 250 W, BCS 500 W, as well as Horizon H-12 Stack) parameter estimation challenges.
- Statistical Friedman and Wilcoxon tests have been performed to compare the performance of GCD with others.

In the rest of the paper, the PEMFC parameter estimation design problem is discussed in Sect. 2, the GCD algorithm is proposed in Sect. 3, the CEC 2017 and CEC 2019 benchmarks are discussed in Sect. 4, and finally the conclusion with future insights in Sect. 5.

2 Problem formulation

This section presents the mathematical formulation utilized to describe the PEMFC. The terminal voltage (S_V) of the PEMFC can be stated in Eq. 1.

$$S_V = S_{cells} * (E_{nernst} - S_{acti} - S_{ohm} - S_{conc}) \quad (1)$$

where, the number of cells S_{cells} connected in series, thermodynamic potential (E_{nernst}), ohmic potential (S_{ohm}), concentration (S_{conc}) as well as activation losses (S_{acti}). This E_{ner} is expressed in Eq. 2:

$$\begin{aligned} E_{nernst} = 1.229 + 4.3085 * 10^{-5} * T_{PEMFC} * & \left[\ln \left(\sqrt{PO_2} \right) + \ln (PH_2) \right] \\ & + .00085 * (T_{PEMFC} - 298.15) \end{aligned} \quad (2)$$

Also, the (S_{acti}), as well as (S_{ohm}) losses are calculated in Eq. 3, and 5, respectively.

$$S_{acti} = -(\chi_1 + \chi_2 * T_{PEMFC} + \chi_3 * T_{PEMFC} * \ln(CO_2) + \chi_4 T_{PEMFC} * \ln(I_c)) \quad (3)$$

$$CO_2 = \frac{P_{O_2}}{5.8 * 10^6 \exp\left(-\frac{498}{T_{PEMFC}}\right)} \quad (4)$$

$$S_{ohm} = I(R_m + R_{con}) \quad (5)$$

where, semi-empirical coefficients (χ_1, χ_2, χ_3 and χ_4), and cell operating current is denoted by I_c . The P_{O_2} , and P_{H_2} is the partial pressure of oxygen, and hydrogen. The C_{O_2} is denoted as the concentration of oxygen. The parameters R_{con} denote the proton exchange membrane equivalent resistance as well as R_m is the electron transfer resistance. The R_m is expressed in the following equations 6:

$$R_m = \rho_m \left(\frac{l}{A} \right) \quad (6)$$

$$\rho_m = \frac{\left(181.6 \left[1 + 0.062 \left(\frac{T_{PEMFC}}{303} \right)^{2.000} * J_i^{2.500} + 0.0300 * J_i \right] \right)}{\left([\lambda - 3.000 * J_i - 0.63400] \exp \left(4.1800 \left(\frac{T_{PEMFC} - 303}{T_{PEMFC}} \right) \right) \right)} \quad (7)$$

where, membrane resistivity ρ_m , and adjustable parameter ($\lambda = [10-24]$).

The S_{conc} mathematically it can be written in Eq. 8:

$$S_{conc} = -\beta * \ln \left[1 - \left(\frac{J_i}{J_{max}} \right) \right] \quad (8)$$

where, semi-empirical coefficient β , current density (J_i), as well as maximum current density (J_{max}).

2.1 Objective Function

In this paper, the optimization objective function (OO_F) is defined as the minimization of the SSE between the experimental PEMFC voltage, and the estimated model voltage. Thus, the problem of estimating the parameters of PEMFCs is approached as an objective target. The OO_F is written in 9 as follows [4] [18].

$$OO_F = \text{Minimize}_{SSE} = F^{obj} = \sum_{k=1}^T [S_{Exper} - S_{Esti}(I_{PEMFC}, X_{PEMFC})]^2 \quad (9)$$

where, experimental stack voltage (S_{Exper}), estimated voltage values (V_{Esti}) at distinct data points corresponding to the PEMFC stack currents I_{PEMFC} , and experimental data series length (T). The $X_{PEMFC} = (\chi_1, \chi_2, \chi_3, \chi_4, \lambda, R_{con}$ and β) are 7 unknown parameters of the PEMFC stack.

3 Proposed Algorithm

In optimization research, hybrid metaheuristic models that use teamwork principles can be classified into low-level hybrids (LLH) and high-level hybrids (HLH). These models represent different ways of integrating and coordinating multiple algorithms to solve complex problems. LLHs focus on integrating specific components or operators from different algorithms at a granular level. They enhance individual aspects of the algorithm, such as using a local search within the genetic algorithm. HLHs combine entire algorithms, often running in parallel or in a hierarchical structure, with mechanisms for communication and collaboration. They aim to enhance the strengths of multiple complete algorithms to improve overall performance. These models reflect the diversity and flexibility of hybrid metaheuristic approaches in optimization, using teamwork principles to tackle complex and large-scale optimization problems more effectively.

In accordance with this principle, we have proposed a new GCD algorithm in the present work. We are using a LLH algorithmic framework to formulate the new algorithm. The newly proposed algorithm is named the gray cuckoo differential (GCD) and consists of GWO [22], CS [40] and DE [34], which are basic algorithms. We are using certain components/operators of these algorithms to propose an LLH strategy. A major reason to use these algorithms is because their simplicity in implementation allows for easy customization, while the synergy between different algorithmic components often results in performance improvements greater than the sum of their parts. In general, LLHs represent an effective approach to tackling complex optimization problems, providing high-quality solutions with enhanced efficiency.

3.1 Initialization

In this phase, the algorithm is initialized between the upper bounds $x_{u,d}$, and the lower bounds $x_{l,d}$ of the problem, for a d dimensional problem, and is given as

$$x_{i,d} = x_{l,d} + r \times (x_{l,d} - x_{u,d}) \quad (10)$$

where $i \in [1, 2, \dots, n]$ and n is the number of individuals for $r \in [0, 1]$. After initialization, the iterative search mechanism is followed by dividing the maximum iterations (t_{max}) into two halves.

3.2 Phase I : The I^{st} Half of Iterations

This phase consists of extensive exploration and intensive exploitation and is given as:

3.3 Extensive Exploration:

Here, we generate a new solution using CS local search operators enhanced by adapting the parameters using Lévy flights operators, as:

$$x_i^{t+1} = x_i^t + \alpha \otimes L(\lambda)(x_{best} - x_i^t) \quad (11)$$

where x_i^t is a random solution, x_i^{t+1} is the current solution, \otimes is entry-wise multiplication, and $\alpha > 0$ is the fixed step size. The Lévy flight based step size which has highly explorative tendencies, and is given by Eq. (12)

$$L(\lambda) \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_0 \gg 0) \quad (12)$$

where $s = \frac{U}{|V|^{1/\lambda}}$, $U \sim N(0, \sigma^2)$, $V \sim N(0, 1)$ and $\sigma^2 = \left\{ \frac{\Gamma(1+\lambda)}{\lambda\Gamma[(1+\lambda)/2]} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right\}$. Also, $\Gamma(\lambda)$ is a gamma function and $\lambda = 1.5$. The parameter N follows a Gaussian distribution with *mean* = 0 and *variance* = σ^2 .

Intensive Exploitation: In this step, we divide the population into 2 sub-populations, by using the crossover equations of 1) basic DE/rand/1; and 2) JADE to add diversity in the solution sets. This helps the algorithm in searching simultaneously in multiple directions with in the search space and ultimately leading to an exploitation operation with in the exploration search. The general equations are given by

$$x_i^t = x_i^t + F.(x_i^t - x_j^t); \quad "DE/rand/1" \quad (13)$$

$$x_i^t = x_i^t + F_i^t.(x_{pbest}^t - x_i^t) + F_i^t.(x_{r1}^t - x_{r2}^t) \quad (14)$$

where x_{pbest}^t is the personal best; r_1 and r_2 are uniformly distributed random numbers; F_i^t is the scaling factor and is evaluated using

$$F = \begin{cases} \frac{1}{2} \times (\sin(2\pi \times freq \times t) \times \frac{t_{max}-t}{t_{max}} + 1); & \text{if } r_1 < 0.5 \\ \frac{1}{2} \times (\sin(2\pi \times freq \times t + \pi) \times \frac{t_{max}-t}{t_{max}} + 1); & \text{if } r_1 > 0.5 \end{cases} \quad (15)$$

3.4 Phase II : 2_{nd} Half Iterations

During this phase, an algorithm should start moving toward extensive exploration and still perform certain exploration so that the algorithm may not get stuck in some local minima.

Intensive Exploration. In GWO, the generalized exploitation operation is suitable for this kind of operation. We are modifying the general equation of CS using a combination with the GWO and is defined by

$$x_{aa} = x_i - P_1(Q_1.x_{new} - x_i^t); \quad x_{bb} = x_i - P_2(Q_2.x_{new} - x_i^t); \quad x_{cc} = x_i - P_3(Q_3.x_{new} - x_i^t) \quad (16)$$

$$x_i^{t+1} = \frac{x_{aa} + x_{bb} + x_{cc}}{3} \quad (17)$$

where x_{new} is the new solution and P_1, P_2, P_3 and Q_1, Q_2, Q_3 are generated from P and Q and are given by $P = 2a.r_1 - a$; $Q = 2.r_2$. The parameter $a \in [0, 2]$ is linearly decreasing.

Extensive Exploitation. Here we use the current best solution and find the optimal solutions in its close proximity. This is because towards the end, there is a possibility that the global solution is close to the current best solution. Thus, for this case, we are using the ‘DE/best/1’ equation and is given by

$$x_i^t = x_{best}^t + F_i^t \cdot (x_i^t - x_j^t); \quad "DE/best/1" \quad (18)$$

3.5 Extensive Switching

There is only one parameter in the proposed algorithm, and to make it adaptive, we are using an exponential decreasing weight that helps to slowly move the algorithm from exploration towards exploitation. This also helps to converge faster and hence better exploration initially and exploitation towards the end [30], and for $\zeta_{min} = 0.25$ and $\zeta_{max} = 0.95$, it is given as

$$p_a(t) = \zeta_{min} + (\zeta_{max} - \zeta_{min}) \exp \left[-\frac{t}{(\frac{t_{max}}{10})} \right] \quad (19)$$

3.6 Selection

This operation is meant to check if the newly obtained solution is better than the best solution from the previous iteration. If the fitness of the current solution $f(x_i^{t+1})$ is better than the previous solution $f(x_i^t)$, the new solution is passed on to the next generation and vice versa.

Algorithm 1. Pseudocode of the proposed GCD algorithm

```

Begin
Define: population size ( $N$ );
problem dimension ( $D$ ); stopping criteria;
if  $i = 1 : \frac{max\_iter}{2}$  then
    global search using Eqn. (11)
    if  $j = 1 : \frac{n}{2}$  then
        local search using Eqn. (13)    DE/rand/1
    else
        local search using Eqn. (14)    JADE
    selection using Eqn. (20)
else
    global search using Eqn. (17)
    local search using Eqn. (18)
    selection using Eqn. (20)
close;
update final best
End

```

$$x_i^{t+1} = \begin{cases} x_i^t & \text{iff } f(x_i^{t+1}) < f(x_i^t) \\ x_i^{t+1} & \text{otherwise} \end{cases} \quad (20)$$

This process is followed unless and until we get the global best solution or the stopping criteria is satisfied. In the next section, we provide the time complexity of the proposed algorithm. The pseudocode of the GCD algorithm is given in Algorithm 1.

4 Results and Discussion

This section discusses the outcomes of simulations performed utilizing the proposed GCD algorithm on CEC benchmark and PEMFC extraction problems. All simulations are carried out using Windows 11 with Matlab R2022b, i5 processor clocked at 2.50 GHz and 16 GB of RAM. This section has 2 subsections, where the I^{st} subsection presents results on the CEC 2017 [7] as well as CEC 2019 [21] benchmark problems, and the second subsection provides extensive details on the PEMFC extraction problem. The 2 statistical tests such as Wilcoxon's ranksum test in terms of "win/loss/tie" (w/l/t) as well as the Friedmann test as average rank are presented. Here, "*win* : +" corresponds to the superior performance of the test algorithm contrasted to the proposed GCD, "*loss* : -" means worse, and "*tie* :=" means equivalent performance or no statistical relevance between the comparative algorithms. For all experiments, the initial population is 50 and the final population is 20. Also, the results for the PEMFC extraction problem are given in terms of MAE, MSE, RMSE, SSE, MBE, and IAE to illustrate the effectiveness of the proposed GCD algorithm contrasted to the 8 other MH optimization algorithms.

4.1 Statistical Results on Numerical Benchmarks Problems

This section presents the results of the comparison of the GCD algorithm on two highly challenging benchmark datasets. These include:

CEC 2017 Benchmarks. The effectiveness of the GCD algorithm in the CEC 2017 benchmarks is compared to BKA [39], CPO [14], CDO [32], SSOA [6], SCHO [8], PO [20], NRBO [33], NMRA [26], NBA [41] and HLOA [23]. For a fair comparison, the total iterations are fixed at 6000 and run equal to 51. Table 1 shows that GCD surpasses other competitive algorithms for the numerical problem $GCD_3 - GCD_5$, GCD_7 , GCD_8 , GCD_{11} , $GCD_{14} - GCD_{21}$, $GCD_{23} - GCD_{26}$, and $GCD_{28} - GCD_{30}$; for GCD_1 and GCD_{12} , HLOA is the best; for GCD_6 , SCHO is better; for GCD_9 , BKA is superior; for GCD_{10} and GCD_{27} , NMRA is better; GCD_{13} and GCD_{22} , PO is the best. In general, it has been observed that for 21 of the 29 problems, the proposed GCD algorithm performs the best.

Table 1. Statistical results on CEC 2017 benchmarks

Problem	BKA [39]	CPO [14]	CDO [32]	SSOA [6]	SCHO [8]	PO [20]	NRBO [33]	NMRA [26]	NBA [41]	HLOA [23]	GCD GCD
w/l/t	02/27/00	00/29/00	00/29/00	00/29/00	03/26/00	05/24/00	01/28/00	04/25/00	00/29/00	20/27/00	NA
Average f-rank	3.896	8.517	8.172	9.793	4.931	3.620	5.517	3.896	11.000	4.931	1.689
Overall f-rank	3.0	9.0	8.0	10.0	5.0	2.0	7.0	4.0	11.0	6.0	1.0

CEC 2019 Benchmarks. The effectiveness of the GCD algorithm on the CEC 2019 benchmarks is compared with jDE100 [9] (winner of the CEC 2019 competition), YDSE [2], PSO [29], NMRA [26], GWO [22], NBA [41], AOA [5], CDO [32], LCA [17], and MPO [14]. For a fair comparison, the total iterations are fixed at 500, and the outcomes are given as mean as well as *std* values of 51 runs. Table 2 shows that GCD surpasses other competitive algorithms for the numerical problem GCD_2 , GCD_3 , GCD_5 , GCD_6 , GCD_7 and GCD_{10} ; for GCD_1 , MPO is the best; for GCD_4 , GCD_8 and GCD_9 , PSO is better. In general, it has been analyzed that for 6 of the 10 problems, GCD performs the best. The statistical outcomes in Table 2, the convergence profiles in Fig. 1, and the box plots in Fig. 2 show that GCD scores the 1ST rank among all other MH algorithms in comparison. The convergence profiles in Fig. 1 show that GCD converges more quickly than YDSE, GWO, NMRA, NBA, AOA, CDO, LCA, and MPO for all the numerical problems. From the box plot Fig. 1 it can be analyzed that the GCD algorithm has a smaller inter-quartile range (IQR) than YDSE, GWO, NMRA, NBA, AOA, CDO, LCA, and MPO for all numerical problems.

Table 2. Statistical results on 100-digit challenges (CEC 2019)

Problem	jDE100 [9]	NBA [41]	PSO [29]	YDSE [2]	NMRA [26]	GWO [22]	AOA [5]	CDO [32]	LCA [17]	MPO [14]	GCD
w/l/t	1/9/0	00/10/00	04/06/00	03/07/00	00/10/00	01/09/00	02/08/00	01/09/00	01/09/00	01/09/00	NA
Average f-rank	9.5	9.9	5.1	2.9	7.0	3.8	5.2	5.6	8.2	6.2	2.5
Overall f-rank	10.0	11.0	4.0	2.0	8.0	3.0	5.0	6.0	9.0	7.0	1.0

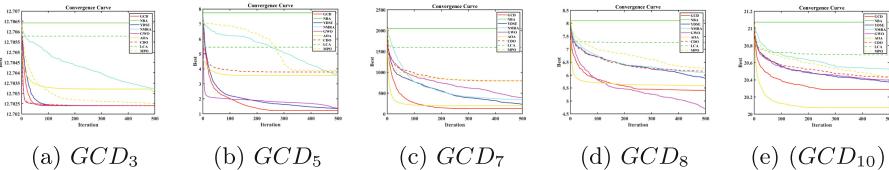
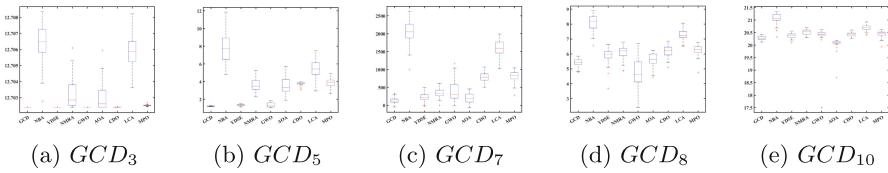


Fig. 1. Convergence profiles for CEC 2019 numerical problems

4.2 Parameter Identification of Proton Exchange Membrane Fuel Cells

In this subsection, we address the problems of extracting parameters from three different PEMFC models using the GCD algorithm. Three distinct PEMFC stacks, namely the BCS 500 W, Stack 250 W, as well as the Horizon H-12 stack, have been employed to evaluate the efficacy of the GCD algorithm. The technical parameters of the PEMFC stack are taken from [3, 13, 42].

**Fig. 2.** Boxplots for CEC 2019 numerical problems**Table 3.** Other statistical error tests

S.NO	PEMFC Stack	MAE	MSE	RMSE	SSE	MBE	IAE
CASE I : BCS500-W							
1	GCD	0.012764	0.00065	0.025495	0.011699	0.000256	0.229748
2	YDSE	0.012869	0.00065	0.025499	0.011704	0.000125	0.231641
3	EDO	0.011622	0.000656	0.025622	0.011817	0.001641	0.209195
4	ZOA	0.014677	0.000699	0.026441	0.012584	-0.00032	0.264191
5	RIME	0.012523	0.000682	0.026123	0.012283	0.000188	0.225417
6	GWO	0.013109	0.000665	0.025795	0.011977	-0.00145	0.235965
7	SCHO	0.014859	0.000868	0.02946	0.015623	0.004869	0.267459
8	SABO	0.022361	0.001608	0.040105	0.028951	0.014933	0.402499
9	HHO	0.017546	0.000783	0.027976	0.014087	-0.00036	0.315825
CASE II: Horizon H-12							
1	GCD	0.060701	0.005359	0.073205	0.096461	5.59E-05	1.09262
2	YDSE	0.060727	0.005359	0.073205	0.096462	-0.00016	1.093083
3	EDO	0.060717	0.005359	0.073206	0.096465	-8E-05	1.092901
4	ZOA	0.060677	0.00536	0.07321	0.096474	2.76E-05	1.092193
5	RIME	0.060722	0.005359	0.073205	0.096463	-0.00017	1.092995
6	GWO	0.060737	0.005359	0.073208	0.096469	-0.00024	1.093266
7	SCHO	0.061159	0.005378	0.073332	0.096796	-0.0038	1.100855
8	SABO	0.06075	0.005361	0.073217	0.096492	-0.00048	1.093497
9	HHO	0.060683	0.00536	0.07321	0.096474	7.52E-07	1.09229
CASE III: Stack 250 W							
1	GCD	0.124859	0.02209	0.148627	0.33135	3.64E-05	1.872883
2	YDSE	0.124822	0.022091	0.14863	0.331361	-0.00061	1.872324
3	EDO	0.124336	0.0221	0.148661	0.331503	-0.0008	1.865033
4	ZOA	0.124899	0.022095	0.148644	0.331426	-0.00023	1.873484
5	RIME	0.124649	0.022093	0.148636	0.331391	-7.1E-05	1.869742
6	GWO	0.125777	0.022113	0.148704	0.331694	-0.00092	1.886652
7	SCHO	0.127948	0.022225	0.149082	0.333382	0.000782	1.919213
8	SABO	0.125911	0.023294	0.152623	0.349408	0.017678	1.888672
9	HHO	0.126549	0.022236	0.149118	0.333542	0.000257	1.898232

The boundary limits of the minimum (LB) as well as the maximum (UB) of unknown parameters ($\chi_1, \chi_2, \chi_3, \chi_4, \lambda, \beta$ and R_{con}) are given taken from [11] [19]. The results of the GCD algorithm are contrasted with some well-known MH algorithms, including ZOA [37], YDSE [2], SCHO [8], SABO [38], RIME [35], HHO [16], GWO [22] and EDO [1]. To provide a fair comparison, all MH algorithms have been assigned the same population size ($P = 50$) as well as the maximum iteration ($M = 500$) for 30 runs.

Statistical calculators such as MAE, MSE, RMSE, SSE, MBE, and IAE; and the optimal parameters of Case I: BCS500-W, Case II: Horizon H-12 as well as Case III: stack 250 W obtained by the proposed GCD algorithm are presented in Table 3 and 4 respectively. From the table, the minimum F_{obj} (SSE) values for Horizon H-12, BCS500-W, as well as stack 250 W, are 0.011699469, 0.09646111, and 0.331349911, respectively. It is observed that the GCD algorithm has obtained the minimum values of the OO_F compared to the other algorithms.

The computed outcomes obtained from the GCD algorithm for all three PEMFC stacks are utilized to match the IP polarization curves with the corresponding measured curves, as depicted in Fig. 3a, 4a, and 5a. This figure demonstrates a strong correlation between the calculated and observed curves. In addition, the IV polarization curves for all three PEMFC stacks are presented in Figs. 3b, 4b, and 5b. This figure also demonstrates a strong correlation between the calculated and observed curves.

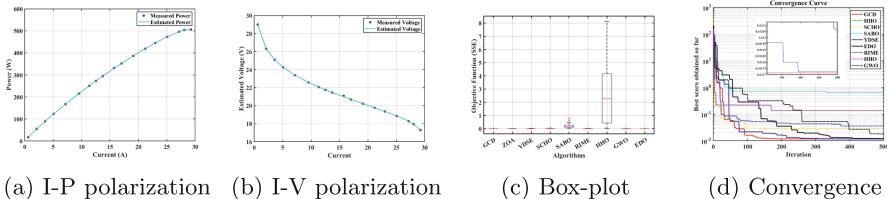


Fig. 3. Case I: BCS500-W

In a similar simulation environment, the boxplot curves of 30 independent runs of the proposed GCD, ZOA, YDSE, SCHO, SABO, RIME, HHO, EDO, and GWO algorithms for the three PEMFC stacks are shown in Figs. 3c, 4c, and 5c. From the figure, it is found that the GCD algorithm has a narrow IQR range. From the box plot analysis, it is concluded that the effectiveness of the proposed GCD algorithm is remarkable contrasted to the other MH algorithms. Also, the convergence curves are presented in Figs. 3d, 4d, and 5d for all three stacks of PEMFCs. From the convergence curve analysis, it is observed that the GCD algorithm has faster convergence as contrasted to the other MH algorithms for all three PEMFC stacks.

Furthermore, the effectiveness and robustness of the GCD algorithm have been tested using the Friedman-Wilcoxon rank test. The results of the Friedman and Wilcoxon rank test are shown in Table 5 for the three PEMFC stacks in

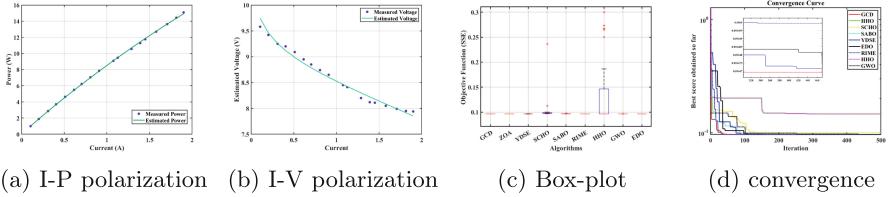


Fig. 4. Case II: Horizon H-12

Table 4. Optimal parameters for all the cases obtained by the various algorithms

S.no	Algorithm	min	mean	Std	Worst	χ_1	χ_2	χ_3	χ_4	λ	β	R_{con}
CASE I : BCS500-W												
1	GCD	0.011699469	0.011719	1.27E-05	0.01175289	-1.03642	0.003259281	7.01E-05	0.00019	20.89938	0.01614	0.0001
2	ZOA	0.012583935	0.021583	0.004573	0.03044587	-0.86588	0.00265866	6.45E-05	-0.00019	19.3468	0.014092	0.000303
3	YDSE	0.011703925	0.011722	1.31E-05	0.01175722	-1.02963	0.003070876	5.93E-05	-0.00019	20.96028	0.016164	0.0001
4	SCHO	0.015622508	0.026158	0.006588	0.04653217	-0.98629	0.003326097	8.43E-05	-0.00019	23.94831	0.015429	0.00048
5	SABO	0.028951265	0.220998	0.17599	0.77564072	-1.1095	0.003371798	6.37E-05	-0.00019	20.7344	0.014905	0.000359
6	RIME	0.01228305	0.021474	6.88E-03	0.03582771	-0.89022	0.0032534	9.80E-05	-0.00019	20.14238	0.015618	0.000109
7	HHO	0.0140873372	0.543881	2.110583	8.1190935	-1.19963	0.003928331	8.16E-05	-0.0002	22.8539	0.016946	0.0001
8	GWO	0.011977338	0.014809	0.00291	0.02287594	-0.96063	0.003018065	6.92E-05	-0.00019	22.00193	0.016135	0.000191
9	EDO	0.011807954	0.012181	3.25E-04	0.01295149	-0.94044	0.002648608	4.93E-05	-0.00019	20.80928	0.015979	0.000124
10	IHBO [4]	0.01170	0.01174	0.00006	0.01185	-1.19970	0.00331	4.20E-05	-0.000193	20.877	0.01613	0.0001
11	ICA [18]	0.011856	-	0.005863	0.034665	-0.908643	0.0024798	4.45E-05	-0.000193	22.66264	0.016238	0.000246
12	FOA [18]	0.011819	-	0.004172	0.030233	-0.992829	0.002621	3.74E-05	-0.000193	21.101126	0.016269	0.0001
13	AHA [3]	0.011831	0.018488	0.010275	0.056728	-1.0497	0.0029	3.84E-05	-0.000193	22.0516	0.016360	0.00018
CASE II: Horizon H-12												
1	GCD	0.09646111	0.096466	3.04E-06	0.09647175	-0.97516	0.002654316	8.54E-05	-9.54E-05	24	0.187549	0.0001
2	ZOA	0.096474158	0.096484	8.50E-06	0.09650448	-0.8536	0.001571915	3.61E-05	-9.54E-05	10.23423	0.164261	0.000111
3	YDSE	0.09646242	0.096467	3.30E-06	0.09647627	-1.01543	0.002683135	7.78E-05	-9.54E-05	24	0.187442	0.000114
4	SCHO	0.096796164	0.103253	0.02594	0.23638833	-1.13326	0.002779152	5.64E-05	-9.54E-05	16.17688	0.177406	0.000398
5	SABO	0.096491889	0.096738	3.30E-04	0.09766144	-0.8532	0.001569286	3.60E-05	9.54E-05	10	0.161575	0.000349
6	RIME	0.096462504	0.096949	1.39E-05	0.096517	-1.19916	0.003250718	7.49E-05	-9.54E-05	23.70921	0.187371	0.000106
7	HHO	0.096474123	0.136127	0.070986	0.30009532	-0.8532	0.001610436	3.90E-05	-9.54E-05	10.28593	0.164503	0.000102
8	GWO	0.096469424	0.096505	2.54E-05	0.09657334	-0.87799	0.00165335	3.62E-05	-9.54E-05	22.10757	0.185914	0.000134
9	EDO	0.096464902	0.096474	5.64E-06	0.09648462	-0.947	0.001984114	4.36E-05	-9.54E-05	22.58797	0.186533	0.0001
10	ABCDE [15]	0.09653605	0.0965368	4.1101e-06	0.09655896	-0.85435	0.002009613	6.76E-05	-9.54E-05	23.00	0.18685	0.0001
11	ELBA [15]	0.096536059	0.09653911517	7.7857e-06	0.0965589673	-0.97712	0.002373	6.46E-05	-9.54E-05	23.00	0.18685	0.0001
12	ASSA [43]	0.097	-	-	-	-1.13	0.00244	3.57E-05	-9.54E-05	18.79	0.1817	0.000714
13	MRFO [27]	0.0966	-	0.0028	0.1062	-1.0630	0.0023641	4.32E-05	-9.54E-05	19.81	0.1829	0.000285
CASE III : Stack 250 W												
1	GCD	0.331349911	0.331395	2.90E-05	0.33146206	-1.05455	0.003426666	7.76E-05	-1.74E-04	14.41682	0.013794	0.0001
2	ZOA	0.331425933	0.333301	0.002054	0.34151317	-0.97107	2.69E-03	4.22E-05	-0.00017	14.5054	0.013791	0.000152
3	YDSE	0.3313610130	0.331435	6.04E-05	0.33160218	-1.03483	0.00353802	8.97E-05	-1.74E-04	14.44E+01	0.013797	0.0001
4	SCHO	0.333381661	0.343373	0.009879	0.37031169	-1.17338	0.003767799	7.74E-05	-1.72E-04	14.51756	0.0136	0.000194
5	SABO	0.334947748	0.439153	1.13E-01	0.82217614	-0.91253	0.002630791	5.04E-05	-1.70E-04	14.72032	0.014615	0.000386
6	RIME	0.331424271	0.334927	3.57E-03	0.34725924	-0.8532	0.002628624	6.25E-05	-1.74E-04	14.37177	0.0136	0.0001
7	HHO	0.3333541564	1.602803	0.979016	3.57583837	-1.19293	0.003353429	4.35E-05	-1.76E-04	14.84529	0.014026	0.000113
8	GWO	0.331693746	0.33706	5.65E-03	0.35188399	-1.17603	0.00359766	6.46E-05	-1.73E-04	14.42958	0.013816	0.000121
9	EDO	0.3315029560	0.332482	6.16E-04	0.33401189	-0.99712	0.00303694	6.19E-05	-1.74E-04	14.5586	0.014081	0.0001
10	IAHA [3]	0.335980	0.335980	6.91E-12	0.335980	-1.0866	0.003	5.10E-05	-1.7E-04	19.9358	0.014527	0.0001
11	COA [36]	0.61391382	0.6295520	2.0189320	0.7222777	-1.1854188	0.0030050793	9.80E-05	-1.206E-04	23.00	0.062561	0.0001
12	IAEO [24]	0.3360	0.3360	0.00000	0.3360	-0.9991	0.002825	4.47E-05	-1.70E-04	19.9358	0.0145	0.0001
13	WOA [12]	0.3372	-	0.0493	0.5029	-0.9565	0.00322221	8.23E-05	-1.75E-04	20.4470	0.0152	0.000108

case I, case II, and case III, respectively. From the tables, it is noticed that the GCD algorithm got the 1^{st} rank followed by the YDSE algorithm, and the HHO algorithm got the last rank (9^{th}) for the BCS500-W stack and the stack of 250 W

Table 5. Statistical results of comparison algorithms

S.no	Algorithm	Friedman's rank	RANK	Winner	Loser	Wilcoxon's p value
CASE I : BCS500-W						
1	GCD	1.668014289	1			
2	ZOA	5.655821506	5	465.0000	0.0000	3.01986E-11
3	YDSE	2.225386612	2	374.0000	91.0000	4.91783E-01
4	SCHO	6.639450199	6	465.0000	0.0000	3.01986E-11
5	SABO	8.639489192	8	465.0000	0.0000	3.01986E-11
6	RIME	6.694855267	7	465.0000	0.0000	3.01986E-11
7	HHO	8.96579969	9	465.0000	0.0000	3.01986E-11
8	GWO	4.862057389	4	465.0000	0.0000	3.01986E-11
9	EDO	3.331867754	3	465.0000	0.0000	3.01986E-11
CASE II : Horizon H-12						
1	GCD	2.097673596	1			
2	ZOA	4.886439712	4	465.0000	0.0000	3.01986E-11
3	YDSE	2.458899692	2	374.0000	91.0000	7.24456E-02
4	SCHO	8.827118879	9	465.0000	0.0000	3.01986E-11
5	SABO	7.886992486	8	465.0000	0.0000	3.01986E-11
6	RIME	6.082705183	5	465.0000	0.0000	5.57265E-10
7	HHO	7.526005871	7	465.0000	0.0000	3.01986E-11
8	GWO	6.114000198	6	465.0000	0.0000	6.69552E-11
9	EDO	3.301334623	3	465.0000	0.0000	2.38974E-08
CASE III : Stack 250-W						
1	GCD	1.563298011	1			
2	ZOA	4.707042578	4	465.0000	0.0000	4.5043E-11
3	YDSE	1.831268687	2	420.0000	45.0000	3.6709E-03
4	SCHO	7.766152124	7	465.0000	0.0000	3.0199E-11
5	SABO	8.48541444	8	465.0000	0.0000	3.0199E-11
6	RIME	5.427159869	5	465.0000	0.0000	4.5043E-11
7	HHO	8.87592351	9	465.0000	0.0000	3.0199E-11
8	GWO	6.013094949	6	465.0000	0.0000	3.0199E-11
9	EDO	3.820142201	3	465.0000	0.0000	3.0199E-11

PEMFC stack. For the Horizon H-12 PEMFC stack, the SCHO algorithm got the last rank (9th) and the proposed GCD algorithm obtained the 1st rank.

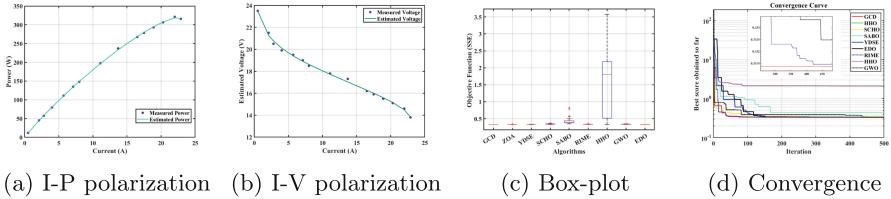


Fig. 5. Case III: Stack 250 W

5 Conclusion

In this paper, a novel hybrid algorithm named GCD is proposed to estimate the parameters in various PEMFC stacks. The proposed GCD algorithm utilizes the search equations of GWO, CS, as well as DE to enhance the exploration phase as well as exploitation phase capabilities of existing optimization techniques. The experimental and statistical results on the CEC 2017 as well as CEC 2019 benchmarks show that the GCD algorithm is highly competitive compared to jDE100, LSHADE, EBOwithCMAR, SaDE, JADE, and others. Experimental results on commercial PEMFCs: case I, case II, and case III, respectively, show that GCD can effectively optimize their parameters compared to YDSE, SCHO, RIME, and others. These outcomes highlight the performance of the GCD algorithm as well as its potential to solve computationally expensive problems. Apart from that, the GCD algorithm might suffer from slow convergence and unbalanced operation, this may be due to the use of unbalanced operations of GWO, DE and CS algorithms. A more robust approach can be preliminary study of each algorithmic component and incorporating the best of all.

As a future research direction, the GCD algorithm can be utilized to solve more complex engineering challenges, particularly in the field of power systems that include controller design, battery model identification, and optimal operation. More work can be done on the application to constrained engineering design problems, expensive optimization problems, among others.

Acknowledgement. The work of Pankaj Sharma was supported by the Council of Scientific and Industrial Research (CSIR), Government of India, for providing the CSIR Senior Research Fellowships (SRF-Direct)(File No: 09/0844(18625)/2024-EMR-I).

References

1. Abdel-Basset, M., El-Shahat, D., Jameel, M., Abouhawwash, M.: Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems. *Artificial Intelligence Review*, pp. 1–72 (2023)
2. Abdel-Basset, M., El-Shahat, D., Jameel, M., Abouhawwash, M.: Young's double-slit experiment optimizer: a novel metaheuristic optimization algorithm for global and constraint optimization problems. *Comput. Meth. Appl. Mech. Eng.* **403**, 115652 (2023)

3. Abdel-Basset, M., Mohamed, R., Abouhawwash, M.: On the facile and accurate determination of the highly accurate recent methods to optimize the parameters of different fuel cells: simulations and analysis. *Energy* **272**, 127083 (2023)
4. Abdel-Basset, M., Mohamed, R., Elhoseny, M., Chakrabortty, R.K., Ryan, M.J.: An efficient heap-based optimization algorithm for parameters identification of proton exchange membrane fuel cells model: analysis and case studies. *Int. J. Hydrogen Energy* **46**(21), 11908–11925 (2021)
5. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H.: The arithmetic optimization algorithm. *Comput. Meth. Appl. Mech. Eng.* **376**, 113609 (2021)
6. Alzoubi, S., Abualigah, L., Sharaf, M., Daoud, M.S., Khodadadi, N., Jia, H.: Synergistic swarm optimization algorithm (2024)
7. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P., Definitions, P.: Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *Technology Report* (2016)
8. Bai, J., et al.: A Sinh Cosh optimizer. *Knowl.-Based Syst.* **282**, 111081 (2023)
9. Brest, J., Maučec, M.S., Bošković, B.: The 100-digit challenge: algorithm JDE100. In: 2019 IEEE Congress on Evolutionary Computation (CEC), pp. 19–26. IEEE (2019)
10. Celikdemir, S.: A new voltage-power based approach for identifying the optimal parameters of PEM fuel cells. *Int. J. Hydrogen Energy* **75**, 592–603 (2024)
11. El-Fergany, A.A.: Extracting optimal parameters of PEM fuel cells using Salp swarm optimizer. *Renew. Energy* **119**, 641–648 (2018)
12. El-Fergany, A.A., Hasanien, H.M., Agwa, A.M.: Semi-empirical PEM fuel cells model using whale optimization algorithm. *Energy Convers. Manage.* **201**, 112197 (2019)
13. Elfar, M.H., Fawzi, M., Serry, A.S., Elsakka, M., Elgamal, M., Refaat, A.: Optimal parameters identification for PEMFC using autonomous groups particle swarm optimization algorithm. *Int. J. Hydrogen Energy* **69**, 1113–1128 (2024)
14. GUO: Manis pentadactyla optimizer (MPO). <https://in.mathworks.com/matlabcentral/fileexchange/157086-chinese-pangolin-optimizer-cpo> (2024). Accessed 13 June 2024
15. Hachana, O., El-Fergany, A.A.: Efficient PEM fuel cells parameters identification using hybrid artificial bee colony differential evolution optimizer. *Energy* **250**, 123830 (2022)
16. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872 (2019)
17. Houssein, E.H., Oliva, D., Samee, N.A., Mahmoud, N.F., Emam, M.M.: Liver cancer algorithm: a novel bio-inspired optimizer. *Comput. Biol. Med.* **165**, 107389 (2023)
18. Kandidayeni, M., Macias, A., Khalatbarisoltani, A., Boulon, L., Kelouwani, S.: Benchmark of proton exchange membrane fuel cell parameters extraction with metaheuristic optimization algorithms. *Energy* **183**, 912–925 (2019)
19. Korkmaz, S.A., Çetinkaya, S.A., Yuksel, O., Konur, O., Erginer, K.E., Colpan, C.O.: Comparison of various metaheuristic algorithms to extract the optimal PEMFC modeling parameters. *Int. J. Hydrogen Energy* **51**, 1402–1420 (2024)
20. Lian, J., et al.: Parrot optimizer: algorithm and applications to medical problems. *Comput. Biol. Med.* **172**, 108064 (2024)

21. Liang, J., Qu, B., Gong, D., Yue, C.: Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization. In: Computational Intelligence Laboratory, Zhengzhou University (2019)
22. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
23. Peraza-Vázquez, H., Peña-Delgado, A., Merino-Treviño, M., Morales-Cepeda, A.B., Sinha, N.: A novel metaheuristic inspired by horned lizard defense tactics. *Artif. Intell. Rev.* **57**(3), 59 (2024)
24. Rizk-Allah, R.M., El-Fergany, A.A., et al.: Artificial ecosystem optimizer for parameters identification of proton exchange membrane fuel cells model. *Int. J. Hydrogen Energy* **46**(75), 37612–37627 (2021)
25. Salgotra, R., Sharma, P., Raju, S., gandomi, A.H.: A contemporary systematic review on meta-heuristic optimization algorithms with their MATLAB and python code reference. *Arch. Comput. Meth. Eng.* **31**(2) 1–74 (2023)
26. Salgotra, R., Singh, U.: The naked mole-rat algorithm. *Neural Comput. Appl.* **31**(12), 8837–8857 (2019). <https://doi.org/10.1007/s00521-019-04464-7>
27. Selem, S.I., Hasanien, H.M., El-Fergany, A.A.: Parameters extraction of PEMFC's model using manta rays foraging optimizer. *Int. J. Energy Res.* **44**(6), 4629–4640 (2020)
28. Shaheen, A., El-Sehiemy, R., El-Fergany, A., Ginidi, A.: Fuel-cell parameter estimation based on improved gorilla troops technique. *Sci. Rep.* **13**(1), 8685 (2023)
29. Shami, T.M., Mirjalili, S., Al-Eryani, Y., Daoudi, K., Izadi, S., Abualigah, L.: Velocity pausing particle swarm optimization: a novel variant for global optimization. In: Neural Computing and Applications, pp. 1–31 (2023)
30. Sharma, P., Raju, S.: Metaheuristic optimization algorithms: a comprehensive overview and classification of benchmark test functions. *Soft Comput.* **28**(4), 1–64 (2023)
31. Sharma, P., Raju, S., Salgotra, R.: An evolutionary multi-algorithm based framework for the parametric estimation of proton exchange membrane fuel cell. *Knowl. Based Syst.* **283**, 111134 (2024)
32. Shehadeh, H.A.: Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization. *Neural Comput. Appl.* **35**(15), 10733–10749 (2023)
33. Sowmya, R., Premkumar, M., Jangir, P.: Newton-Raphson-based optimizer: a new population-based metaheuristic algorithm for continuous optimization problems. *Eng. Appl. Artif. Intell.* **128**, 107532 (2024)
34. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
35. Su, H., et al.: RIME: a physics-based optimization. *Neurocomputing* **532**, 183–214 (2023)
36. Sultan, H.M., Menesy, A.S., Kamel, S., Jurado, F.: Developing the coyote optimization algorithm for extracting parameters of proton-exchange membrane fuel cell models. *Electr. Eng.* **103**, 563–577 (2021)
37. Trojovská, E., Dehghani, M., Trojovský, P.: Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* **10**, 49445–49473 (2022)
38. Trojovský, P., Dehghani, M.: Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* **8**(2), 149 (2023)
39. Wang, J., Wang, W.c., Hu, X.x., Qiu, L., Zang, H.f.: Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artif. Intell. Rev.* **57**(4), 98 (2024)

40. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: 2009 World Congress on Nature and Biologically Inspired Computing (NABIC), pp. 210–214. IEEE (2009)
41. Yang, X.S., Hossein Gandomi, A.: Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29**(5), 464–483 (2012)
42. Yuan, K., Ma, Y., Zhang, H., Razmjoooy, N., Ghadimi, N.: Optimal parameters estimation of the proton exchange membrane fuel cell stacks using a combined owl search algorithm. *Energy Sources Part A Recovery Utilization Environ. Effects* **45**(4), 11712–11732 (2023)
43. Zhu, Y., Yousefi, N.: Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm. *Int. J. Hydrogen Energy* **46**(14), 9541–9552 (2021)



Spatio-Temporal Graph Neural Networks for Infant Language Acquisition Prediction

Andrew Roxburgh, Floriana Grasso, and Terry R. Payne^(✉)

University of Liverpool, Liverpool, UK
`{A.Roxburgh,F.Grasso,T.R.Payne}@liverpool.ac.uk`

Abstract. Predicting the words that a child is going to learn next can be useful for boosting language acquisition, and such predictions have been shown to be possible with both neural network techniques (looking at changes in the vocabulary state over time) and graph model (looking at data pertaining to the relationships between words). However, these models do not fully capture the complexity of the language learning process of an infant when used in isolation. In this paper, we examine how a model of language acquisition for infants and young children can be constructed and adapted for use in a *Spatio-Temporal Graph Convolutional Network (STGCN)*, taking into account the different types of linguistic relationships that occur during child language learning. We introduce a novel approach for predicting child vocabulary acquisition, and evaluate the efficacy of such a model with respect to the different types of linguistic relationships that occur during language acquisition, resulting in insightful observations on model calibration and norm selection. An evaluation of this model found that the+ mean accuracy of models for predicting new words when using sensorimotor relationships (0.733) and semantic relationships (0.729) were found to be superior to that observed with a 2-layer Feedforward neural network. Furthermore, the high recall for some relationships suggested that some relationships (e.g. visual) were superior in identifying a larger proportion of relevant words that a child should subsequently learn than others (such as auditory).

Keywords: Neural Models of Infants and Child Development · Spatial-Temporal Graph Neural Networks · Language Acquisition Prediction

1 Introduction

Developmental Language Disorder (DLD) is a condition whereby a child shows significant difficulties with language development for no clear reason. It is the most common disability in pre-schoolers, affecting somewhere between 5–7% of all UK children [21, 30, 36]. The condition is exacerbated by other factors associated with learning difficulties, or environmental or familial factors. For example,

in the UK, between 40% and 56% of children in areas of social disadvantage begin nursery school with such a language delay [19, 22]. This has been proven to cause a knock-on effect within child education; for example, DLD has an incidence rate of 50–90% on reading difficulties [35], which can have a subsequent impact on educational outcomes (in fact reading levels have been shown to be a good predictor of educational outcomes [14]). Furthermore, children show great difficulties catching up with their peers without adequate support [5, 8, 9, 37], and thus are more likely to perform worse academically, to show a lower employment rate, and to have poorer future mental health [7, 32, 42].

Language delay is also an indicator of many other developmental issues; at least 3% of all UK children have a language delay linked with hearing impairment, specific learning difficulties such as dyslexia, and general learning needs [21]. Communication difficulties also feature in Autism Spectrum Disorder, which affects 1% of the UK population [2]. Such challenges are not limited to neurologically ‘atypical’ children, as the child’s communication environment and family circumstances (e.g. poverty or limited language exposure) can have an effect on any early years language development, with a negative, cascading effect on school outcomes years later [26, 38].

With such a public health issue of this scale, identifying affected children early is extremely important. One of the most commonly-used methods for identifying such children is through the use of standardised tools such as the *MacArthur-Bates Communicative Development Inventory (CDI)* [10], a paper instrument originally developed in US English but now adapted into many languages [1]. It consists of a series of questions and checklists designed to assess how children understand and produce words and gestures, as well as their acquisition of grammar. It is typically accompanied by a Family Questionnaire, which gathers information about the family members and individuals who generally spend time with the child during a typical week. This data can be used to generate a *spatial model of language acquisition* (where the spatial component corresponds to words with specific linguistic relationships), which can then be used to identify children who may suffer from language delay.

By automating the process of acquiring this data, through mobile applications such as *Babyltalk*¹ [28], a model of a child’s language acquisition can be built over time, and used to predict those words that the child should learn in the future [3, 4]. This knowledge can be used as a potential intervention for children with DLD, by suggesting words that a child’s carer should emphasise when teaching language [16]; for example, using the *Babyltalk* application. This could also be used to boost the language acquisition of neurologically atypical children, and could have further research applications. One way to accomplish this is to refer to *Age-of-Acquisition* norms [33] that can be used to encourage caregivers to emphasise words that typical children would acquire at a similar age. However, children (as individuals) do not tend to learn the same words at the same rate or age, and so a more tailored approach is required. The accuracy of the model used to predict forthcoming words is therefore an important factor.

¹ <http://www.lucid.ac.uk/>.

Graph Neural Networks (GNN) [40] extend the concept of conventional feed-forward neural networks to capture the relationships and interactions between nodes within a graph, by learning a vector representation of each node that not only depends on its own features but also on the features of its neighbours. These *node embeddings* encode the structural and feature data of themselves, their neighbouring nodes, and ultimately extend to the overall graph structure. The resulting model can be used for edge prediction, node or graph classification, labelling, and feature prediction; or in the context of language acquisition, it could be used to determine whether the word that the node represents will be ‘known’ by the child.

In this paper we examine how a model of language acquisition for infants and young children can be constructed and adapted for use in a *Spatio-Temporal Graph Convolutional Network (STGCN)*, taking into account the different types of linguistic relationships that occur during child language learning. We introduce a novel approach for attempting to predict child vocabulary acquisition, and evaluate the efficacy of such a model with respect to the different types of linguistic relationships that occur during language acquisition. Section 2 provides an overview of existing approaches that explore how vocabulary learning can be modelled, and provides a short overview of the STGCN based approach. The way in which a child’s vocabulary can be modelled as a graph is presented together with a discussion of how it is used by the STGCN in Sect. 3. After describing the datasets collected for this study (Sect. 4), we evaluate the approach empirically (Sect. 5) and conclude in Sect. 6.

2 Background

2.1 Previous Approaches to Vocabulary Models

Vocabulary acquisition can be thought of as the evolution of a network of interconnected nodes, where each node models a word and the edges represent a variety of different relationships between them (several of these relationships are discussed in Sect. 3). As more vocabulary is acquired, the graph expands, and new connections are established. Much of the work in modelling child vocabulary growth has exploited the use of graph representations and artificial neural networks, due to the fact that the words a child learns are inherently connected with each other. Examples include the use of graphs when modelling vocabulary growth over time [17], and the use of neural networks for modelling the way that a brain acquires language [31]. Thus, graph theory can be used as a means of analysing and simulating vocabularies and their evolution.

Beckage, Mozer & Colunga [3] showed that it is possible to predict words that would be learnt in the future using the words a child already knows. By analysing CDI questionnaire data of 77 subjects over a 1-year period at 1-month intervals, and by using a network growth technique, they constructed three different models (*Additive*, *Maximal*, and *Threshold*), each calculating the probability of a word being learnt within the next month in slightly different ways, but all three relying on the probability of a new word being learned as a consequence of the set

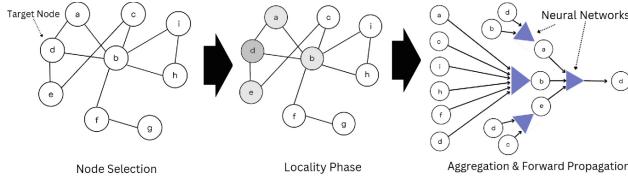


Fig. 1. General operation of a *Spatial Graph Neural Network* (GNN).

of words already learnt, and specifically the other words learnt concurrently (i.e. during that same 1-month period). They also successfully developed several neural-network based predictive models using a variety of qualitatively different sources of information as inputs, and observed that the accuracy of predictions could be enhanced by either increasing the temporal resolution of the data, or including more meaningful connections between words in the predictive model.

Subsequent work not only supported the hypothesis that individual words in a child's vocabulary are informative in predicting future vocabulary growth, but that predictions based on this knowledge reliably outperformed those based purely on demographics [4]. It was also noted that the specific composition of a child's vocabulary significantly affects a model's ability to predict the child's future language acquisition (i.e. the acquisition of new words), suggesting that an individual's vocabulary has relevant and predictive information on the type of learner - and the learning trajectory - of a particular child.

2.2 Overview of Graph Neural Network Approaches

Gori [12] and Scarselli [29] introduced *Graph Neural Networks (GNN)* [40], which extend the concept of conventional feedforward neural networks to graph-structured data. This development allowed for more effective processing of this type of data, which historically had posed challenges for many machine learning approaches. Unlike traditional neural networks, which are often designed for handling data in regular structures, such as images or data sequences, GNNs process inherently irregular data with irregular structures, e.g. potentially varying graph sizes, with no fixed number of neighbours for each nodes, and with complex connectivity patterns and attributes. They manage relationships (edges) between entities (nodes/vertices) within a graph, by learning a vector representation of each node (i.e. a *node embedding*) that not only depends on its own features but also on the features of its neighbours. For a vocabulary model, each node in the graph corresponds to a word, and is represented by a feature vector that is initialised with an attribute representing the probability that this word is known.

Spatial GNNs focus on the spatial relationships between nodes, operating directly in the graph's node and edge space. They essentially 'pass messages' between neighbouring nodes, aggregating information from a node's local neighbourhood to learn representations. Figure 1 illustrates the general operation of a Spatial GNN [27]; each node is selected and its local neighbourhood is determined (i.e. the *Locality Phase*), which for a vocabulary model, is governed by

the connection between word pairs. The graph structure is then reconfigured (i.e. the *Aggregation Phase*) prior to transforming the aggregated features through a neural network layer resulting in a new feature vector for each node, that is informed by its neighbours (i.e. the *Forward Propagation Phase*). Once all of the nodes have been processed, a final representation (i.e. a node embedding) is obtained for each of the nodes. These embeddings can then be used to determine the probability that the words they represent are ‘known’.

Graph Convolutional Networks (GCNs) [20] apply a convolution operation to graphs in GNNs in a similar way to traditional Convolutional Neural Networks (CNNs), with many filtering methods possible, typically within the Fourier domain (Spectral GCNs) or in the spatial domain (Spatial GCNs). An interesting variation was proposed by Kipf & Welling [18] that merges the two methods, with spectral graph convolutions which are simplified to reduce the overhead that comes with computing a Fourier transform of a graph.

Spatio-Temporal Graph Convolutional Networks (STGCN) further extend GCNs, by combining a time dimension in addition to the spatial dimension, and as such, can be used for handling graph-structured, time-series data. Yan *et al.* [41] first proposed the concept for recognising actions in skeleton models, and similar designs have been described as a method of predicting traffic in road networks [43, 44], with the data coming from road networks and traffic sensors represented as a graph, where edges represent direct routes between sensors, and the time dimension referring to the history of the sensor data [15].

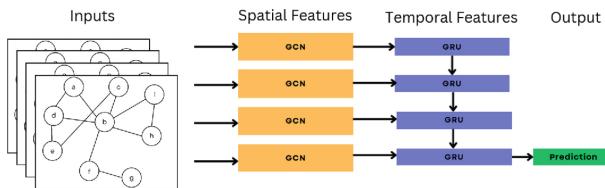


Fig. 2. Block diagram of the T-GCN algorithm [44] used for the Spatio-Temporal Graph Convolutional Network model, using *Graph Convolutional Networks* (GCNs) as spatial features and *Gated recurrent units* (GRUs) as temporal features.

STGCN algorithms, such as T-GCN [44] shown in Fig. 2, typically employ a hybrid design that combines a graph convolutional network block, with a Recurrent Neural Network block such as a *gated recurrent unit* (GRU), designed for sequential data. The graph convolutional network is used to capture the relationships between nodes, thus modelling spatial dependence, and the gated recurrent unit is used to capture the dynamics of the node features, modelling temporal dependence. STGCNs allow the modelling of evolving graph structures in which node dependencies change over time, which makes them a promising tool for attempting to predict a child’s future vocabulary on the basis of a dynamic vocabulary together with fixed relationships between words.

3 Modelling Vocabulary as a Graph

In a basic representation of the child's vocabulary as a graph, each node represents a word with an associated feature vector, incorporating information about the state of the word. We can represent the depth of the child's knowledge of each word by including a feature quantifying it. Each feature is initialised with discrete values that indicate that a child may: i) understand a word but cannot produce it; ii) produce a word but does not understand it; iii) understand and produce a word; or iv) have no knowledge of a word. Whilst this model may be a possible oversimplification of a child's word knowledge and cognition, these discrete states have the advantage of being easily observable and well understood.

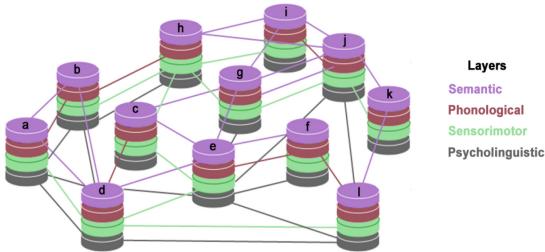


Fig. 3. Simplified example of a *Multiplex Lexical Network* with 4 relationship layers and 12 words represented by letters a - l. Edges are weighted (but not shown here).

The model can be enhanced by incorporating quantified relationships between words (thereby defining and weighting the graph edges). The various graphs can be arranged as a *multiplex* [39] graph (Fig. 3), which is a type of graph where nodes represent the same set of items on all layers. In the context of language, this has also been described as a *Multiplex Lexical Network (MLN)* [34]. A variety of data exists that could be used for this purpose, including:

Word Associations: words that have some cognitive association with each other - for example *lock/key*, or *umbrella/rain*;

Phonological Connections: data on words that have a measurable similarity due to sharing similar phonemes;

Psycholinguistic Norms: how easy a word can be mentally visualised (i.e. *Imageability*);

Semantic Features: connecting words of a similar meaning in context;

Concretedness: whether a word represents a concrete or an abstract concept;

Familiarity: how commonly used a word is; or

Other: simple measures such as how difficult a word is to remember or pronounce (e.g. word length).

Whilst a number of relationships exist that could be included, care needs to be taken not to rely too heavily on an 'adult perspective', but rather concentrate on those more in line with an infant's cognitive perspective. In this paper, we discuss the incorporation of the following set of norms in the model:

Semantic Feature Production norms measure the semantic similarity of different words. We focus on using the McRae [24] and Buchanan [6] norms. **Sensorimotor norms** measure perception and action strength of words from the perspective of the child. The *Lancaster Sensorimotor Norms* [23] are used that evaluate English words based on six perceptual modalities (*touch, taste, smell, hearing, vision, and interception*) and five action effectors (*mouth/throat, hand/arm, foot/leg, head excluding mouth/throat, and torso*).

Sensorimotor norms are named after the initial phase in Piaget’s theory of cognitive development [25], which spans from birth to approximately 2 years of age. During this period, infants learn to use their senses to construct an understanding of the world, and use motor movements (*reaching, sucking, grasping, and touching*) to interact with it. Thus, the primary objective of the sensorimotor stage is to develop an understanding of *object permanence*, i.e. the realisation that objects and events persist even when they are not directly observable by the child. From the perspective of infant language acquisition, it is especially useful to be able to connect words from the conceptual point of view of a child at the earliest development stage. Figures 4 and 5 illustrate a simplified vocabulary graph with edges connecting nodes with similar *Lancaster Norm* scores [23]: Fig. 4 illustrates nodes in an *auditory* graph; whereas Fig. 5 illustrates those in a *gustatory* one. Both graphs are simplified and focused on the word **chicken**.

To represent the strengths of connections between words in the graph for each of the similarity measures we used, in the case of the Semantic Feature Production Norms, the necessary cosine similarity matrices were used together with the data by McRae and Buchanan. Lancaster norms require additional processing for use in our model, and thus a weighted adjacency matrix was created for all possible word pairs within each category by computing the cosine similarity of each pair, which in this case is simply the normalised product of their scores. Thus, words with, for example, high scores in the *Olfactory* category (e.g. **orange** and **lemon**) would have a strong connection, whilst a word with high score and a word with low score (e.g. **orange** and **table**) would have a weak connection. Similarly, the connection between two words with low scores in the *Olfactory* category (e.g. **table** and **chair**) would also be weak. In addition, all self-loops (e.g. **head** and **head**) have a maximal weight (i.e. 1.0).

For each category, the resulting matrix comprised approximately 76,000 data points. Given that many of the pairs may not be relevant to the category and would result in an unnecessary computational burden, a threshold was used (with the inter-word connection value of 0.5) to eliminate pairs deemed unnecessary due to the implied lack of connection. For each sensorimotor category, we used the adjacency matrices to define the edges, and created a list of nodes by removing duplicates in the edge list. Finally, a feature was also added to each node representing the level of knowledge of that word.

4 Data Collection

Two types of data were collected to build a model and evaluate it; datasets of words and their relationships were collected in order to create the model itself

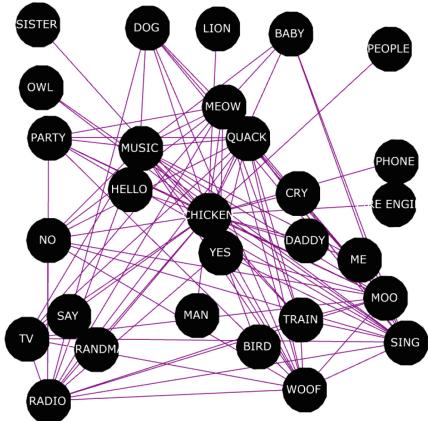


Fig. 4. Example *auditory sensorimotor* relationship graph (simplified) focused on the word *chicken* (based on Lancaster Sensorimotor Norms [23]).

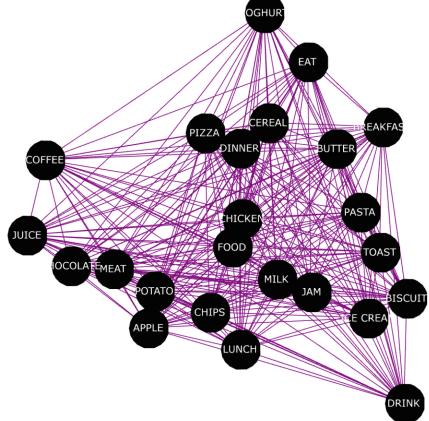


Fig. 5. Example *gustatory sensorimotor* relationship graph (simplified) focused on the word *chicken* (based on Lancaster Sensorimotor Norms [23]).

(i.e. the *Relational Datasets*), and observational data on child vocabulary was collected in order to test the resulting model (i.e. the *Observational Dataset*). For the *Relational Datasets*, we used the English language relationship datasets comprising both semantic and sensorimotor norms from [6, 23, 24]. A challenge in putting together independent datasets into one model is one of standardisation. This should be addressed by accounting for differences due to: i) synonyms (e.g. *rabbit* and *bunny*); ii) regional variations (e.g. *mommy* and *mummy*); and iii) international spelling variations (e.g. *colour* and *color*). This was addressed by pre-processing the data to identify such cases, and renaming words for consistency, thus resulting in a *standardised* vocabulary.

A second challenge was to consider *homographs*, i.e. words with multiple meanings such as *back*, *bank*, *run*, or *drink*. This was addressed by using a contextual specifier as a label for ambiguous words, thus ensuring that the vocabulary was unambiguous. For example, ambiguous words such as *drink* become *drink(beverage)* and *drink(verb)*. This is particularly challenging when considering the child perspective, as for certain words, such relationships are different than for adults. To illustrate this, the notion of *fish* being a food and *fish* being an animal are typically treated as distinct concepts for children, whereas for adults *fish* is understood as simultaneously both a food and an animal. Again these words were appended with a context-appropriate label.

The *Observational Dataset* consisted of data collected from participants using our web app as part of a larger study² as well as item-level CDI Survey responses extracted from all available forms in English acquired from Wordbank [11]. We

² The study has received ethical approval by the Faculty of Science and Engineering Research Ethics Committee, University of Liverpool, Ref. 8182.

selected only data for which there are longitudinal sequences, to enable the T-GCN algorithm’s GRU layer block to train on temporal data. Words were converted to our standardised vocabulary to allow for Irish, Australian, American and regional British dialect differences. The overall dataset included 460 observations (i.e. vocabulary inventories), in sequences of 4 contiguous observations, using a sliding window method to create smaller sequences from longer ones. The training data consisted of 115 ‘test subjects’ corresponding to individual sequences (as a child may generate more than one sequence), and the test dataset consisted of 120 observations across 30 ‘test subjects’.

Due to the nature of real-world, human-collected data regarding human behaviour, inevitably there will be errors present, and thus working with such data presents challenges. With a large dataset containing millions of observations, occasional errors will have minimal impact on the overall behaviour. However, such errors may have an adverse effect on trials with smaller datasets (such as that used for this evaluation), and thus, to address this, the data was pre-processed to identify possible errors and eliminate discrepancies. Common errors encountered in observational data include:

Observer Misinterpretation: a child may be observed at one time period as understanding but not producing a particular word; however it becomes apparent over time that the child has a different interpretation for that word.

Mistaken Observation: the parent believes that a child has said and/or understood a word, but they change their mind in the subsequent observation.

Observational Error: a sequential pair of observations that made sense to the parent at the time of each observation, but that were subsequently confused, suggesting the child has either forgotten the meaning of the word, or forgotten how to say it. In this study, we assume that children do not forget words once they have demonstrated knowledge of them.

Unobserved knowledge: the parent may be unaware that a child knows a particular word; e.g. the child may have used the word at a relatives’ house.

Articulation Error: the child may speak a certain word, but pronouncing it in such a way that it is unintelligible or ambiguous to an observer.

Dropping words from the observed vocabulary could adversely affect the ability to train a classifier on the data; therefore to deal with contradictory data, two datasets were generated, an *optimistic* one whereby it was assumed that the child continue to understand the word in subsequent observation periods, and a *pessimistic* one whereby the assumption is that words are false observations, and that the child did not understand the word during the first observation.

5 Implementation and Evaluation

A collection of thirteen ($N = 13$) individual models were used for the vocabulary relationships explored in this study, represented as graphs $G_N = (V_N, E_N)$ that were based on each of the norms described in the data model (Sect. 3), where V_i

is the set of vertices or nodes, and E_i are the corresponding set of edges for each model $i \in N$. A set of nodes was created for every observation in the data, and was populated with the corresponding observed data. These sets of nodes were combined to form a time series. The set of edges were then processed by combining each edge set with each node set in the time series, resulting in the creation of a time series of graphs for each of the vocabulary relationships. The aim of the model is to predict a child's future vocabulary based on past and existing knowledge, represented as a series of graphs representing different relationships between words in a vocabulary over a number of discrete time periods. Therefore, the nodes were embedded with feature vectors representing the child's current knowledge of the word at each time period. This comprehension attribute was assigned a starting value from one of four levels, reflecting the child's knowledge of that word at the given observation: i) no comprehension (0.0); ii) production without understanding (0.3); iii) understanding but no production (0.6); and iv) full comprehension and production (1.0).

A Spatio-temporal Graph Convolutional Network-based model was developed in Python using the *StellarGraph*³ software library, and configured using the hyperparameters in Table 1. Stellargraph is built on *Tensorflow*,⁴ and facilitates the construction of graph-based machine learning models. The Graph Neural Network algorithm used in this study is based on the Temporal-GCN (T-GCN) as described by Zhao et al. [44]. The full model consists of 13 relationship layers, each of which is a separate STGCN model that has been individually trained and executed. Some nodes in these layers may be unconnected as they have no meaningful associations with other words in the context of that relationship.

When a new prediction is required, the vector representing the child's current vocabulary is used to populate the feature vectors of each node on each relationship layer, representing the new words that the child had recently learned. The GCN classifier, in conjunction with the STGCN's spatio-temporal block, is then applied to these graphs to re-classify the unknown nodes. These newly classified nodes can then be used to determine the words that are likely to be influenced the most by its neighbours, and so may be learned next. This results in a list of candidate words from each relationship layer, from which the most likely words a child should learn can be determined.

5.1 Training and Validation

The training stage of our STGCN involves presenting the model with a time series of graphs - analogous to a spatio-temporal graph - representing observations of childrens' vocabulary changing over time. This consists of a chain of 4-consecutive-observation chunks. The Sequence Length parameter of the STGCN was set to match the consecutive observation chunk size. We split the input data into training, testing and validation sets.

³ <https://github.com/stellargraph/stellargraph>.

⁴ <https://www.tensorflow.org/>.

Table 1. Experimental HyperParameters used by all STGCN models.

Parameter	Value	Parameter	Value
Data Mode	<i>Optimistic</i>	Epochs	1000
Batch Size	4	LSTM Sequence Length	4
Prediction Length	1	Loss Optimiser	ADAM
Loss Metric	MAE	Model Metric	MSE
Dropout	<i>None</i>	Input Graph Connections Limit	2000

A 2-layer Feedforward Neural Network model was also trained using vocabulary data with no relationship element, to provide a baseline for the evaluation. Each word in the child’s vocabulary was linked to an input node in the network, which connects to a hidden layer with 500 units. Each word is represented as a node in the output layer. Other model hyperparameters include a Learning Rate of 0.8 and Momentum of 0.9, and the network was trained over 1000 epochs.

Table 2. Results for each of the 13 individual models and the ANN baseline.

Reference (baseline) Model	Precision	Recall	F1	Accuracy
2-Layer Feedforward (ANN)	0.283	0.854	0.426	0.610
Neural Network				
Semantic Relationships:				
McRae [24]	0.450	0.513	0.479	0.740
Buchanan [6]	0.403	0.606	0.484	0.715
Sensorimotor Relationships [23]:				
Lancaster (Haptic - <i>Touch</i>)	0.419	0.586	0.488	0.730
Lancaster (Gustatory - <i>Taste</i>)	0.427	0.598	0.498	0.731
Lancaster (Olfactory - <i>Smell</i>)	0.424	0.571	0.487	0.733
Lancaster (Auditory - <i>Hearing</i>)	0.465	0.395	0.427	0.750
Lancaster (Visual - <i>Vision</i>)	0.438	0.618	0.513	0.732
Lancaster (Interoceptive)	0.435	0.494	0.462	0.739
Lancaster (Mouth/Throat)	0.417	0.637	0.504	0.716
Lancaster (Hand/Arm)	0.404	0.540	0.462	0.722
Lancaster (Foot/Leg)	0.428	0.548	0.480	0.733
Lancaster (Head)	0.443	0.509	0.474	0.737
Lancaster (Torso)	0.438	0.537	0.483	0.741

5.2 Evaluation

A total of thirteen models were evaluated (2 that model semantic relationships and 11 that model sensorimotor relationships, based on the norms described in Sect. 3), and compared with the baseline 2-layer Feedforward Neural Network. The aim is to evaluate each of these models independently, and compare them with the simple neural network approach. The results are presented in Table 2, together with the following evaluation metrics:

Precision: Measures the accuracy of positive predictions; i.e. out of all the positive predictions (i.e. words that the model has predicted will have increased comprehension), how many of those were proven to be correct.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: Measures the fraction of relevant instances retrieved; i.e. out of all the words that were actually proven to be learned, how many were correctly predicted by the model.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 Score (or F1 Measure): The harmonic mean of precision and recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy: Measures the fraction of correct predictions, representing the ratio of correctly predicted words with increased comprehension, to the total number of new words actually learned by the child.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

Although the overall performance of the ANN demonstrated moderate performance, it fell short of that demonstrated by each of the STGCN models (Table 2). The results suggests that although many of the words were predicted, the percentage of those being correct was low. In comparison, the use of the different relationships with the STGCN model resulted in a greater accuracy of the words predicted, but with a drop in the overall percentage of possible predictions.

The accuracy of individual predictive models varied greatly despite achieving similar overall accuracy across all models for the entire test set. There is some variance in the performance of the different STGCN models incorporating Sensorimotor relationships, with *Visual* and *Mouth* models showing notably higher recall values at 0.618 and 0.637 respectively. These high recall values are indicative of the models' ability to identify a larger proportion of relevant words

for learning, making them suitable choices for minimising false negatives. The *Auditory* STGCN model clearly stands out with the highest accuracy (0.750) and highest F1-score (0.465) but the lowest Recall (0.395), suggesting it is more selective but might miss out on some words the child could learn. The *Haptic*, *Gustatory*, and *Olfactory* models balance between F1-scores and Recall, with F1-scores in the 0.488 to 0.498 range, indicating a reasonable compromise between the metrics. Overall, F1-scores for the Sensorimotor STGCN models are relatively high, ranging from 0.716 to 0.750, suggesting they make correct predictions about whether a word will or will not be typically learned. The *Torso*, *Foot/Leg*, and *Interoceptive* models show neither highest nor lowest performance, but could still potentially provide useful predictions for vocabulary acquisition. The performance when using the semantic relationships was comparable to those of the sensorimotor relationships, with the McRae et al. norms resulting a model that performed marginally better than that using the Buchanan et al. norms.

A comparative evaluation was also conducted to compare the two variants of the observational data: *optimistic* and *pessimistic*, with the former correcting contradictions in observations by assuming the child knew the words that appeared to be forgotten, while the latter assumed an observational error by the caregiver and that the child did not know the word. In general, the *optimistic* version outperformed the *pessimistic* version, thus indicating the significance of the observational data in determining the accuracy of predictive models.

6 Conclusions and Discussion

A Spatio-Temporal Graph Neural Network model (and the way in which it can be constructed) was presented that can be used to make predictions about a child’s upcoming vocabulary acquisition. The model is based on existing work on infant language acquisition prediction using neural networks [31] and graph models [17]; however, we have expanded this by considering the current and past vocabularies of a given child combined with multiple types of relationships between the words, including sensorimotor norms [23] and semantic feature prediction norms [6, 24].

The overall accuracies of each model were reasonably similar; however on closer inspection, individual prediction level showed several differences, suggesting there is scope for exploitation of the variation of coverage. One approach would be to combine and improve the predictability of multiple models through the use of an ensemble output stage [13] that would allow each of the models to contribute towards the overall prediction. We will expand the number of relationships we use to inform the input graphs, including word association norms, phonological relationships, and psycholinguistic vocabulary norms such as imageability and concreteness. Furthermore, given that this study addresses an important public health issue with an innovative methodology, a more detailed comparative analysis with existing state-of-the-art techniques could provide clearer insights into how well the model performs, as well as issues of scalability.

References

1. Alcock, K.J., et al.: Construction and standardisation of the UK communicative development inventory (UK-CDI), words and gestures. In: International Conference on Infant Studies (2016)
2. Baron-Cohen, S., Scott, F.J., Allison, C., et al.: Prevalence of autism-spectrum conditions: UK school-based population study. *Br. J. Psychiatry* **194**(6), 500–509 (2009)
3. Beckage, N., Mozer, M., Colunga, E.: Predicting a child's trajectory of lexical acquisition. In: Noelle, D.C., et al. (eds.) 37th Annual Meeting of the Cognitive Science Society, CogSci 2015 (2015)
4. Beckage, N.M., Mozer, M.C., Colunga, E.: Quantifying the role of vocabulary knowledge in predicting future word learning. *IEEE Trans. Cogn. Develop. Syst.* **12**(2), 148–159 (2020)
5. Bleses, D., Makransky, G., Dale, P.S., et al.: Early productive vocabulary predicts academic achievement 10 years later. *Appl. Psycholinguist.* **37**(6), 1461–1476 (2016)
6. Buchanan, E.M., Valentine, K.D., Maxwell, N.P.: English semantic feature production norms: an extended database of 4436 concepts. *Behav. Res. Methods* **51**(4), 1849–1863 (2019)
7. Clegg, J., Hollis, C., Mawhood, L., et al.: Developmental language disorders - a follow-up in later adult life. *Cognitive, language and psychosocial outcomes. J. Child Psychol. Psychiatry* **46**(2), 128–149 (2005)
8. Conti-Ramsden, N., Botting, Z., Simkin, E., et al.: Follow-up of children attending infant language units: outcomes at 11 years of age. *Int. J. Lang. Commun. Disord.* **36**(2), 207–219 (2001)
9. Feinstein, L., Duckworth, K.: Development in the early years: its importance for school performance and adult outcomes. Technical report, Centre for Research on the Wider Benefits of Learning, London (2006)
10. Fenson, L., Marchman, V., Thal, D., Dale, P., Reznick, J., Bates, E.: MacArthur-Bates Communicative Development Inventories, 2nd edn. Paul H. Brookes Publishing Company, Baltimore, MD (2007)
11. Frank, M.C., Braginsky, M., Yurovsky, D., et al.: Wordbank: an open repository for developmental vocabulary data. *J. Child Lang.* **44**(3), 677–694 (2017)
12. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings of 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 729–734. IEEE, Montreal, Que., Canada (2005)
13. Hansen, L., Salamon, P.: Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 993–1001 (1990)
14. Hulme, C., Snowling, M.J.: Developmental Disorders of Language Learning and Cognition. John Wiley & Sons, Chichester (2013)
15. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: a survey. *Expert Syst. Appl.* **207**, 117921 (2022)
16. Johnson, E.K., Jusczyk, P.W.: Word segmentation by 8-month-olds: when speech cues count more than statistics. *J. Mem. Lang.* **44**(4), 548–567 (2001)
17. Ke, J., Yao, Y.: Analysing language development from a network approach. *J. Quan. Linguist.* **15**(1), 70–99 (2008)
18. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017 (Poster) (2017)

19. Law, J., McBean, K., Rush, R.: Communication skills in a population of primary school-aged children raised in an area of pronounced social disadvantage. *Int. J. Lang. Commun. Disord.* **46**(6), 657–64 (2011)
20. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(12), 6999–7019 (2022)
21. Lindsay, G., Dockrell, J., Mackie, C., et al.: Educational provision for children with specific speech and language difficulties in England and Wales. CEDAR, University of Warwick, Technical report (2002)
22. Locke, A., Ginsborg, J., Peers, I.: Development and disadvantage: implications for the early years and beyond. *Int. J. Lang. Commun. Disorders* **37**(1), 3–15 (2002)
23. Lynott, D., Connell, L., Brysbaert, M., et al.: The Lancaster sensorimotor norms: multidimensional measures of perceptual and action strength for 40,000 english words. *Behav. Res. Methods* **52**(3), 1271–1291 (2020)
24. McRae, K., Cree, G.S., Seidenberg, M.S., et al.: Semantic feature production norms for a large set of living and nonliving things. *Behav. Res. Methods* **37**(4), 547–559 (2005)
25. Piaget, J.: *The Origins of Intelligence in Children*. Routledge, London (1953)
26. Roulstone, S., Law, J., Rush, R., et al.: Investigating the role of language in children's early educational outcomes. Technical report. DFE-RR134, Department of Education, UK (2011)
27. Roxburgh, A.: Using Graph Neural Networks to Predict Toddler Vocabulary Acquisition. Ph.D. thesis, University of Liverpool, Liverpool, UK (2024)
28. Roxburgh, A., Grasso, F., Payne, T.: Predicting word learning to boost child language acquisition. In: 7th International Conference on Digital Health, DH 2017, London, UK (2017)
29. Scarselli, F., Yong, S.L., Gori, M., et al.: Graph neural networks for ranking web pages. In: The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), pp. 666–672. IEEE (2005)
30. Scerri, T.S., et al.: DCDC2, KIAA0319 and CMIP are associated with reading-related traits. *Biol. Psychiatry* **70**(3), 237–245 (2011)
31. Sims, C., Schilling, S., Colunga, E.: Exploring the developmental feedback loop: word learning in neural networks and toddlers. In: Proceedings of Annual Meeting of the Cognitive Science Society, CogSci 2013. vol. 35, pp. 3408–3413 (2013)
32. Snowling, M.J., Adams, J.W., Bishop, D.V., et al.: Educational attainments of school leavers with a preschool history of speech-language impairments. *Int. J. Lang. Commun. Disord.* **36**(2), 173–183 (2001)
33. Stadthagen-Gonzalez, H., Davis, C.J.: The Bristol norms for age of acquisition, imageability, and familiarity. *Behav. Res. Methods* **38**(4), 598–605 (2006)
34. Stella, M., Beckage, N.M., Brede, M.: Multiplex lexical networks reveal patterns in early word acquisition in children. *Sci. Rep.* **7**(March), 1–10 (2017)
35. Stothard, S.E., Snowling, M.J., Bishop, D., et al.: Language-impaired preschoolers. *J. Speech Lang. Hear. Disord.* **41**(2), 407–418 (1998)
36. Tomblin, J.B., Records, N.L., Buckwalter, P., Zhang, X., Smith, E., O'Brien, M.: Prevalence of specific language impairment in kindergarten children. *J. Speech Lang. Hear. Disord.* **40**(6), 1245–1260 (1997)
37. Van Dulm, O., Southwood, F.: Does socioeconomic level have an effect on school-age language skills in a developed country? *Stellenbosch Papers Linguist. Plus* **49**, 59–84 (2016)

38. Walker, D., Greenwood, C., Hart, B., et al.: Prediction of school outcomes based on early language production and socioeconomic factors. *Child Dev.* **65**(2), 606 (1994)
39. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications. Structural Analysis in the Social Sciences*. Cambridge University Press, Cambridge (1994)
40. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(1), 4–24 (2021)
41. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32(1), pp. 7444–7452 (2018)
42. Young, A.R., Beitchman, J.H., Johnson, C., et al.: Young adult academic outcomes in a longitudinal sample of early identified language impaired and control children. *J. Child Psychol. Psychiatry* **43**(5), 635–645 (2002)
43. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)* (2018)
44. Zhao, L., et al.: T-GCN: a temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **21**(9), 3848–3858 (2020)



AUV Efficient Navigation Relying on Adaptive Proximal Policy Optimization

Jingzehua Xu¹, Yongming Zeng², Jintao Zhang¹, Xuanchen Li¹, Lingru Meng², Haocai Huang², Jingjing Wang^{3(✉)}, and Yong Ren⁴

¹ Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

² Ocean College, Zhejiang University, Zhoushan, China

³ School of Cyber Science and Technology, Beihang University, Beijing, China
drwangjj@buaa.edu.cn

⁴ Department of Electronic Engineering, Tsinghua University, Beijing, China

Abstract. Safe and efficient navigation is crucial for autonomous underwater vehicles (AUVs) to perform various marine monitoring tasks. Considering the complex and unknown underwater environment and limited sensing ability of AUVs, the traditional methods based on models and relying on large amounts of input information are not practical enough, and reinforcement learning (RL) has been widely discussed as one of the most promising schemes. Among many RL algorithms, the proximal policy optimization (PPO) based on trust region optimization theory not only improves sampling efficiency but also reduces deployment complexity by constraining updates of current and previous policies within an alternate trust region. Nevertheless, the performance of PPO is easily influenced by fixed clipping bounds and lacks adaptability. In order to dynamically optimize clipping bounds, we propose the adaptive PPO (APPO) algorithm for AUV navigation tasks. APPO dynamically explores and exploits clipping bounds during online training using a bandit to maximize the value of the upper confidence bound of each candidate boundary, guiding PPO to use different clipping bounds at different stages of online training to improve training efficiency and stability. Extensive simulation experiments demonstrate that APPO is more suitable for AUV navigation tasks compared to other baseline algorithms, showing superior performance in terms of robustness, stability, and adaptability.

Keywords: Autonomous underwater vehicles · Efficient navigation · Adaptive proximal policy optimization · Multi-armed bandit

1 Introduction

Autonomous underwater vehicles (AUVs) have greatly promoted the progress of marine science, playing a crucial role in the fields of seabed mapping, resource

J. Xu and Y. Zheng—These authors contributed equally to this work.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
M. Mahmud et al. (Eds.): ICONIP 2024, LNCS 15296, pp. 149–164, 2025.

https://doi.org/10.1007/978-981-96-6606-5_11

survey, information collection and so on [5]. Efficient and robust navigation is the key to ensure that AUVs can complete tasks safely and efficiently in complex and unknown environments. However, conducting efficient and safe navigation, given constraints of limited prior environmental knowledge and the AUVs' restricted sensing capabilities, presents practical challenges and holds great value [2].

In the past few decades, the navigation methods used for various unmanned vehicles can be roughly divided into graph-based methods represented by Dijkstra algorithm and A-star Algorithm, sample-based methods represented by probabilistic roadmap and rapid exploration random tree algorithm, and artificial intelligence methods represented by ant colony optimization algorithm, genetic algorithm and neural networks [2]. Considering the complexity of the underwater environment, the above methods cannot be directly transferred to the AUVs. Therefore, the performance of the above traditional navigation methods in the ocean current environment is compared and analyzed in [12]. To solve the problem of slow convergence speed and poor effect of heuristic algorithms, Wen et al. proposed a fusion heuristic algorithm for AUV navigation under ocean current interference by integrating genetic algorithm, simulated annealing algorithm and ant colony optimization algorithm [7]. Gong et al. established a multi-trajectory planning model based on ant colony optimization and comprehensively considered constraints such as underwater environment and motion efficiency to deduce multiple alternative trajectories in order to select the best scheme [3]. Unfortunately, such model-based control methods not only rely on prior environmental information but also need to perform parameter tuning. In the face of unknown and dynamic environments, they suffer from high time complexity and computational complexity, lack generalization and learning ability, and have limited application scenarios.

Reinforcement learning (RL) is widely used in AUV navigation because it can optimize the decision-making of the agent by interacting with the environment, so as to cope with the dynamic changing environment without the need for an exact environment model [1, 10]. However, with the increase of task complexity and environment state dimension, traditional RL algorithms will face dimension disaster due to insufficient memory. Deep reinforcement learning (DRL) combined with deep neural network is used to solve these problems and has achieved satisfactory results in various fields. In [4], the author proposed an obstacle avoidance algorithm based on DRL based on the obstacles detected by sonar to ensure the safe navigation of AUVs in complex environments. In [8], the author designed an actor-critic structure optimal adaptive distributed controller based on DRL for end-to-end AUV motion planning and formation control. Other studies have applied DRL to AUV trajectory tracking, autonomous positioning, target hunting and other applications, all of which demonstrate the superior performance of DRL [6, 9]. Although remarkable progress has been made in AUV navigation, DRL algorithms still face problems such as slow convergence, unstable training, and low learning efficiency [2, 11]. Among DRL algorithms, trust region policy optimization (TRPO) improves stability and ensures monotony convergence by limiting the update of new policies to one trust region. Proximal Policy Opti-

mization (PPO), derived from TRPO, simplifies deployment by constraining the updates between old and new policies within a modified trust region. However, the fixed clipping bound limits the performance of PPO, and PPO cannot adjust the conservative degree of its strategy update according to the current learning situation. Therefore, it is very beneficial to explore and study dynamic clipping bounds to improve PPO performance.

Based on the above analysis, this paper proposes a novel DRL method named Adaptive PPO (APPO) for AUV navigation. APPO features an adaptive clipped trust region mechanism that dynamically adjusts the clipping bounds via bandit during online training. By optimizing the upper confidence bound (UCB) value for each candidate bound, PPO is directed to employ different cut bounds at various stages of online training. This approach helps in selecting the most effective cut bounds at each stage, thereby enhancing the algorithm's performance. Simulation results show that our proposed APPO can not only adapt to navigation tasks in different scenarios, but also has higher learning rate and stability compared with other baseline algorithms.

The remainder of this paper is structured as follows: Sect. 2 introduces the system model of the AUV navigation task. Section 3 introduces the problem formulation, while Sect. 4 introduces the methodology, mainly including the APPO algorithm design and its principle modules. Section 5 presents experiments and comparisons, followed by the conclusions drawn in Sect. 6.

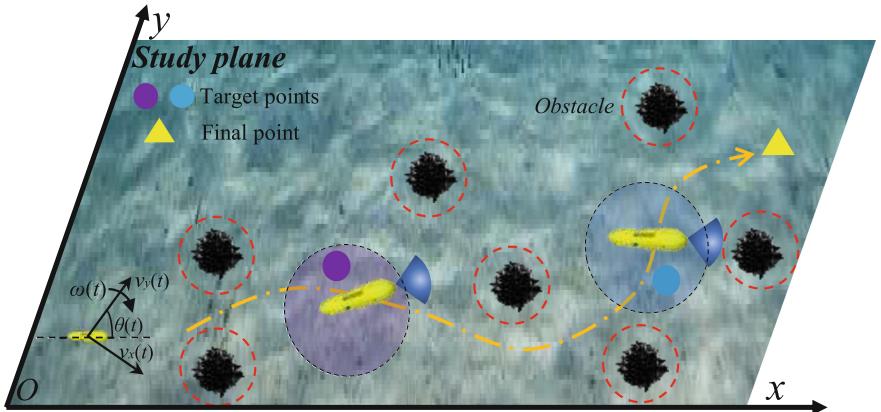


Fig. 1. Illustration of the AUV navigation system model in the obstacle environment, which consists of two main models, AUV dynamics model, and underwater detection model. The AUV is working to complete the navigation task while avoiding obstacles.

2 System Model

The AUV navigation model we considered is shown in Fig. 1, and the AUV conducts the navigation task in a two-dimensional plane with a fixed depth.

During the process, the AUV can obtain the location of the target points through various underwater sensors, and detect the environment through sonar to avoid obstacles in order to reach the target points safely. The task ends when the AUV reaches the final point. The AUV motion model and the underwater detection model are described in detail in Sect. 2.1 and Sect. 2.2, respectively.

2.1 AUV Motion Model

Without loss of generality, we consider a three-degree-of-freedom motion model for AUV navigation tasks. At time t , the AUV's body frame reference is represented by $\mathbf{v} = [v_x(t), v_y(t), \omega(t)]^T$, where $v_x(t)$, $v_y(t)$, $\omega(t)$ are surge velocity, sway velocity and yaw angular velocity, respectively. And the world reference frame at time t can be represented by $\boldsymbol{\eta} = [x(t), y(t), \theta(t)]^T$, where $x(t)$ and $y(t)$ indicate the position of the AUV, while $\theta(t)$ denotes the yaw angle. According to Fossen's motion [4], the motion model of AUV considering hydrodynamics and hydrostatic forces is

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{J}(\boldsymbol{\eta}(t)) \cdot \mathbf{v}(t), \quad (1)$$

$$\mathbf{M}_A \dot{\mathbf{v}}(t) + \mathbf{C}_A(\mathbf{v}(t)) \cdot \mathbf{v}(t) + \mathbf{D}_A(\mathbf{v}(t)) \cdot \mathbf{v}(t) + \mathbf{G}_A(\boldsymbol{\eta}(t)) = \boldsymbol{\tau}(t), \quad (2)$$

where \mathbf{M}_A , \mathbf{C}_A , \mathbf{D}_A and \mathbf{G}_A represent the inertia matrix with the added mass of the AUV, the Coriolis centripetal force matrix, the damping matrix for the viscous fluid force and the composite matrix of gravity and buoyancy, respectively. Moreover, $\boldsymbol{\tau}(t)$ is the control input, while $\mathbf{J}(\boldsymbol{\eta}(t))$ stands for the transformation matrix, which can be defined as

$$\mathbf{J}(\boldsymbol{\eta}(t)) = \begin{bmatrix} \cos \theta(t) & -\sin \theta(t) & 0 \\ \sin \theta(t) & \cos \theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

According to practical application, above equations need to be discretized as

$$\boldsymbol{\eta}(t+1) = \boldsymbol{\eta}(t) + \Delta T \cdot \mathbf{J}(\boldsymbol{\eta}(t+1)) \cdot \mathbf{v}(t), \quad (4)$$

$$\mathbf{v}(t+1) = \mathbf{v}(t) + \Delta T \cdot \mathbf{M}_A^{-1} F(\boldsymbol{\eta}(t), \mathbf{v}(t)), \quad (5)$$

where ΔT is the time interval, and $F(\boldsymbol{\eta}(t), \mathbf{v}(t))$ can be calculated by

$$F(\boldsymbol{\eta}(t), \mathbf{v}(t)) = \boldsymbol{\tau}(t) - \mathbf{C}_A(\mathbf{v}(t)) \cdot \mathbf{v}(t) - \mathbf{D}_A(\mathbf{v}(t)) \cdot \mathbf{v}(t) - \mathbf{G}_A(\boldsymbol{\eta}(t)). \quad (6)$$

2.2 Underwater Detection Model

During the navigation process, the AUV uses the sonar to detect the environment, including obstacles and target points, which can be modeled using the active sonar equation [6]

$$EM = SL - 2TL(f, d) + TS - NL(f) + DI - DT, \quad (7)$$

where SL , TL , TS , NL and DI are the emission sound strength, transmission loss, target strength, environmental noise level and directionality index of target, respectively [2]. Additionally, DT and EM represent the detection threshold and echo margin, respectively. Furthermore, TL is related to the detection distance d and the center acoustic frequency f , which can be expressed as

$$TL = 20 \log(d) + d \times a(f) \times 10^{-3}, \quad (8a)$$

$$a(f) = 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f^2} + 2.75 \times 10^{-4} f^2 + 0.003, \quad (8b)$$

where $\alpha(f)$ is the attenuation coefficient of sound wave in water, and the maximum detection radius r_m of the AUV can be determined by considering the monotonically decreasing relationship between EM and the detection distance d , and we have

$$r_m = \arg \max_d \{EM(d) \geq 0\}. \quad (9)$$

3 Problem Formulation

We describe the AUV navigation problem as a Markov decision process (MDP), which can be defined by a quintuple, i.e.

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}(\cdot | s(t), a(t)), \mathcal{R}, \gamma), \quad (10)$$

where \mathcal{S} and \mathcal{A} denotes the state and action space of the AUV, respectively. Moreover, γ is the discount factor, while $\mathcal{P}(\cdot | s(t), a(t))$ represents the state transition probability function. To be intuitive, at time t , AUV selects the action $a(t) \in \mathcal{A}$ according to its policy π_θ by observing the current state $s(t) \in \mathcal{S}$, and transitions to the next state $s(t+1) \sim \mathcal{P}(\cdot | s(t), a(t))$ and gets the reward $r(t) \in \mathcal{R}$. The details are as follows:

State Space: In the navigation task, the observation space of AUV at time t is $s(t)$, which can be defined as

$$s(t) = [\mathbf{l}(t), l_{A \leftrightarrow T}(t), \theta(t), \phi_{A \leftrightarrow T}(t), \chi(t)], \quad (11)$$

where $\mathbf{l}(t)$ contains the distances detected by sonar between the AUV and various obstacles, while $l_{A \leftrightarrow T}(t)$ represents the distance between the AUV and the target point. $\theta(t)$ and $\phi_{A \leftrightarrow T}(t)$ respectively indicate the orientation angle (yaw angle) of the AUV and the angle between the AUV and the target point. Furthermore, $\chi(t) \in \{0, 1\}$, and $\chi(t) = 1$ indicates the current training episode has concluded, while vice versa.

Action Space: In the process of navigation task, the AUV makes action $a(t)$ at time t by observing the state $s(t)$ and action $a(t)$, which can be given by

$$a(t) = [v(t), \omega(t)], \quad (12)$$

where $\|v(t)\| = \sqrt{v_x(t)^2 + v_y(t)^2}$ and $\|\omega(t)\|$ indicate the linear and angular velocity of the AUV, respectively. And the AUV can adjust its own motion state by changing its linear and angular velocity.

Reward Function: We need to design the corresponding reward function to guide the AUV to make reasonable decisions in the complex environment to optimize the navigation trajectory to safely complete the navigation task. The rewards received by the AUV at time t consist of the following parts

$$r_c(t) = -500 \text{ceil}(l_{\text{safe}} / \min(l(t))), \quad (13)$$

$$r_g(t) = 1000 \text{ceil}(l_{A \leftrightarrow T}^{\max} / l_{A \leftrightarrow T}(t)), \quad (14)$$

$$r_e(t) = -0.2 + 5(l_{A \leftrightarrow T}(t-1) - l_{A \leftrightarrow T}(t)) + 2(\phi_{A \leftrightarrow T}(t-1) - \phi_{A \leftrightarrow T}(t)), \quad (15)$$

where $r_c(t)$ is a penalty term used to prevent the AUV from colliding with the obstacles, while l_{safe} is the safe distance between the AUV and the obstacles, and $\text{ceil}(\cdot)$ is the integer up function. Additionally, when the AUV detects the target point for the first time, it receives a reward $r_g(t)$. In addition, we use the reward item $r_e(t)$ to encourage the AUV to get closer to the target point. Therefore, the total reward available for AUV at time t can be weighted by

$$r(t) = \delta_c r_c(t) + \delta_g r_g(t) + \delta_e r_e(t), \quad (16)$$

where δ_c , δ_g and δ_e are the weights of each reward or penalty item, respectively, which can be adjusted according to the application needs.

Based on the above analysis, we summarize several engineering constraints that need to be considered during the actual navigation process, and formulate a constraint optimization problem whose goal is to optimize the policy of the AUV to maximize the total expected return. The constrained optimization problem can be expressed as

$$\max_{\pi_\theta} J(\theta) = \max_{\pi_\theta} E \left[\sum_{t'=t}^{T=\infty} \gamma^{t'-t} r_{t'}(s(t), \pi_\theta(a(t) | s(t))) \right], \quad (17a)$$

$$\text{s.t. } \min(l(t)) \geq l_{\text{safe}}, \quad (17b)$$

$$\text{s.t. } l_{A \leftrightarrow T}(t) \leq l_{A \leftrightarrow T}^{\max}, \quad (17c)$$

$$v_{\min} \leq \|v(t)\| \leq v_{\max}, \omega_{\min} \leq \|\omega(t)\| \leq \omega_{\max}, \quad (17d)$$

where Eq. (17a) denotes the optimization objective, and Ineq. (17b) represents the constraint that prevents the AUV from colliding with obstacles. Moreover, Ineq. (17c) stands for the constraint that ensures the AUV to get to the target point, while Ineq. (17d) restricts the velocity and angular velocity range of the AUV.

4 Methodology

In this section, we mainly introduce the principals for APPO, which consists of three main modules, trust region optimization, multi-armed bandit and upper

confidence bound, and sampling clipping bound with alternate uncertainty term. Based on the modules, we finally present the pseudo-code of the APPO algorithm in detail.

4.1 Trust Region Optimization

Importance Sampling. Algorithms like TRPO and PPO utilize importance sampling to convert on-policy algorithms into approximations of off-policy algorithms. This modification enables the use of collected data to train the current policy by applying the policy gradient method, i.e.

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}} \left[\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}} \right], \quad (18)$$

where τ is the collected data, π_{old} represents the old policy, while π_{cur} denotes the current policy. In addition, $A^{\pi_{\text{old}}}$ stands for the advantage function, whose value is determined by state, action and π_{old} of the AUV.

KL Divergence and Trust Region Optimization. The expression in Eq. (18) could result in the deviation of the new policy from old policy, complicating the attainment of the optimal solution. Consequently, it becomes essential to reduce the Kullback-Leibler (KL) divergence between the current and previous policies, denoted as $D_{\text{KL}}(\pi_{\text{cur}} \parallel \pi_{\text{old}})$, thereby limiting the updates to the policy to remain within a designated trust region

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}} \left[\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}} - D_{\text{KL}}(\pi_{\text{cur}} \parallel \pi_{\text{old}}) \right]. \quad (19)$$

Additionally, we can directly limit the updates between the current and previous policies to a fixed trust region, thus Eq. (19) can be translated into

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}} \left[\min \left(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}}, \text{clip} \left(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}}, 1 - \epsilon, 1 + \epsilon \right) A^{\pi_{\text{old}}} \right) \right], \quad (20)$$

where ϵ is the clipping bound controlling the range for policy updating.

4.2 Multi-Armed Bandit and Upper Confidence Bound

Considering n independent variables $\mathcal{T} = \{\epsilon_0, \epsilon_1, \dots, \epsilon_i, \dots, \epsilon_n\}$ that are identically distributed, we model the process of sampling from these clipping bounds as a multi-armed bandit game. Upon sampling the i -th clipping bound ϵ_i , an immediate reward $r_{t=N_{\epsilon_i}}$ is obtained, where N_{ϵ_i} represents the cumulative number of times ϵ_i has been accessed. Subsequently, we can compute the expected return as $\mathbb{E}[R_i | \epsilon_i]$, and we have

$$U(\epsilon_i) = \mathbb{E}[R_i | \epsilon_i] = \sum_{t=0}^{t=N_{\epsilon_i}} \gamma^t r_t(\epsilon_i). \quad (21)$$

In the procedure of sampling from \mathcal{T} and updating $\mathbb{E}[R_i|\epsilon_i]$, opting for the most promising clipping bound based on the highest expected return during sampling is known as exploitation. In contrast, it is known as exploration. Notably, solely relying on exploitation for updating the expected return estimations without incorporating exploration could hinder our ability to determine the optimal clipping bound, which might result in overestimating the confidence of sub-optimal clipping bounds. To mitigate the risks of balance between exploitation and exploration, necessitates the integration of uncertainty estimation into the process.

The UCB is a decision-making algorithm, which serves to maintain an equilibrium between the exploration and exploitation of options by integrating an uncertainty estimation $\hat{U}(\epsilon_i)$

$$U^{UCB}(\epsilon_i) = U(\epsilon_i) + \hat{U}(\epsilon_i). \quad (22)$$

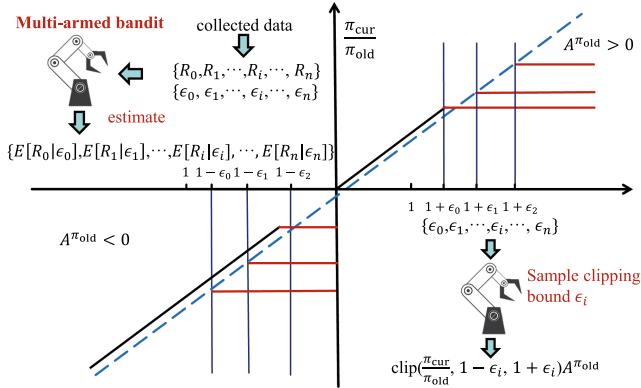


Fig. 2. The schematic diagram of proposed APPO algorithm.

Besides, the uncertainty estimation of i -th clipping bound ϵ_i can be formulated as

$$\hat{U}(\epsilon_i) = \lambda \sqrt{\frac{N_{\text{bandit}}}{N_{\epsilon_i}^{\text{bandit}} + \text{eps}}}, \quad (23)$$

where λ is a coefficient to control the value of uncertainty estimation, while eps is a very small float number set to prevent value overflow.

Specifically, given the sampling times $N_{\epsilon_i}^{\text{bandit}}$ of ϵ_i and total sampling times $N^{\text{bandit}} = \sum_{\epsilon_i \in \mathcal{T}} N_{\epsilon_i}^{\text{bandit}}$, if a certain clipping bound is sampled infrequently, resulting in a lower N_{ϵ_i} , it will correspondingly yield a higher $\frac{N_{\text{bandit}}}{N_{\epsilon_i}^{\text{bandit}}}$, leading to a larger U^{UCB} . This encourages the exploitation of such clipping bounds.

Algorithm 1. APPO Algorithm

1: Initialize the parameters, including the multi-armed bandit with clipping bounds $\mathcal{T} = \{\epsilon_0, \epsilon_1, \dots, \epsilon_i, \dots, \epsilon_n\}$, the total sampling times N , the sampling times of each bandit $\{N_{\epsilon_0}^{\text{bandit}}, N_{\epsilon_1}^{\text{bandit}}, \dots, N_{\epsilon_i}^{\text{bandit}}, \dots, N_{\epsilon_n}^{\text{bandit}}\}$, online replay buffer $\mathcal{D}_{\text{online}}$, the critic network of PPO, old and new policies $\pi_{\text{old}}, \pi_{\text{cur}}$ of the AUV.

2: **for** each episode k **do**

3: Reset the training environment, total reward, and visitation counters.

4: **for** each time step t **do**

5: Sample an action according to the policy:
6: $a_t \sim \pi_{\text{old}}(a_t | s_t)$;

7: Collect the next state from environment:
8: $s_{t+1} \sim \mathcal{P}(s_{t+1} | s_t, a_t)$;

9: Calculate reward r_t by Eq. (13) ~ Eq. (16);

10: Store sampling tuple (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}_{\text{online}}$.

11: Computing UCB values $U^{UCB}(\epsilon_i)$ via Eq. (21)~(23);

12: Sample a clipping bound according to the maximum of UCB values:
13: $\epsilon^* \leftarrow \arg \max_{\epsilon_i} \{U^{UCB}(\epsilon_i) | \epsilon_i \in \mathcal{T}\}$

14: **end for**

15: Update the policy π_{cur} with Eq. (20)

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}} [\min(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}}, \text{clip}(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}}, 1 - \epsilon^*, 1 + \epsilon^*) A^{\pi_{\text{old}}})]$$
.

16: Update the evaluated return of the bandit:

$$R^{\text{bandit}} \leftarrow R^{\text{bandit}} + R_{\epsilon^*}^{\text{bandit}}$$

17: Update the total visitation counter and bandit visitation counter by

$$N^{\text{bandit}} \leftarrow N^{\text{bandit}} + 1$$

$$N_{\epsilon^*}^{\text{bandit}} \leftarrow N_{\epsilon^*}^{\text{bandit}} + 1$$
.

18: Update the old policy with the new policy:

$$\pi_{\text{old}} \leftarrow \pi_{\text{cur}}$$

19: Update the critic network in PPO via mean-squared error L_{MSE} .

20: **end for**

4.3 Sampling Clipping Bound with Alternate Uncertainty Term

Therefore, we can sample the optimal clipping bound with highest UCB value to efficiently balance exploration with exploitation of candidate clipping bounds

$$\epsilon^* \leftarrow \arg \max_{\epsilon_i} \{U^{UCB}(\epsilon_i) | \epsilon_i \in \mathcal{T}\}. \quad (24)$$

Next, we detail the updating process for the expected return of clipping bound ϵ_i , namely, updating $\mathbb{E}[R_i | \epsilon_i]$. For each sampled clipping bound ϵ_i , we first update the policy π_{old} . Subsequently, we evaluate this updated policy π_{cur} , obtaining the average evaluated return $R_{\epsilon_i}^{\text{bandit}}$ as the reward $r_{N_{\epsilon_i}}$ for arm ϵ_i . Consequently, we can compute the expected return of sampling ϵ_i as $\mathbb{E}[R_i | \epsilon_i]$.

Based on the above modules, we can integrate them together with PPO to compose the design of APPO algorithm, whose pseudo-code is detailed in Algorithm 1.

5 Experiments

In this section, we aim to validate the proposed APPO through simulation experiments of training a single AUV for the navigation task. First we present the experimental setup, followed by a detailed description of the entire experiment process. Subsequently, we analyze and discuss the results of the experiments, focusing on the performance of APPO.

5.1 Experimental Setup

During the simulation, we employ two distinct sets of parameters: the simulation environment and algorithm parameters. These sets of parameters are considered comprehensively to ensure an effective evaluation.

Table 1. Parameters of Simulation Experiment

Parameters	Values
Max velocity v_{\max}	1.0 m/s
Max angular velocity ω_{\max}	1.6 rad/s
Experimental site size	40 m × 40 m
Safe distance $l_{\min}^{i \leftrightarrow j}$	1.6 m
Target distance $l_{\max}^{i \leftrightarrow T}$	2.5 m
discount factor γ	0.99
Maximum steps per episode T	2000
Time step per episode Δt	0.25
Training episodes ε	1000
Hidden layer size	256

Simulation Environment Parameters. The simulation is carried out on a 40 m × 40 m area with a water depth of −200 m, on which the obstacles are randomly distributed. At the beginning, the position of the AUV is randomly distributed, and the AUV knows its own position. The area boundaries act as obstacles to restrict the AUVs in the specified area.

Algorithm Parameters. The implementation of APPO incorporates various parameters and settings. The discount factor γ is assigned a value of 0.99. During each episode, a maximum of 2,000 steps T are allowed, with a simulation time step Δt of 0.25 s. A hidden layer size of 256 is utilized. All the parameters are detailed in Table 1 for a summary.

5.2 Experimental Results and Analysis

Algorithm Comparison Experiments. We first conducted simulation experiments on a $40\text{ m} \times 40\text{ m}$ square, employing the APPO algorithm to train the AUV for navigation and obstacle avoidance. Each training episode commenced with the AUV positioned at a specified coordinate (x, y, z) , with a training reset condition, denoted as $\chi(t)$, triggered upon colliding obstacle or upon reaching the maximum step count in an episode, thereby initializing the AUV's position to (x', y', z') for the subsequent episode. At each episode, the AUV constantly interacted with the surrounding environment, making real-time decision and obtaining corresponding reward based on the AUV's current state. However, the preliminary training phase, characterized by AUV's suboptimal policy, resulted in frequent collisions with obstacles. While the collected interaction experience served as the valuable data for training the policy and networks after each episode. It is also notable that each bandit arm had very high uncertainty estimation at the beginning of the training, which necessitated the APPO algorithm to engage in exploring each arm.

As training progressed, the APPO adeptly balanced exploration and exploitation of clipping bounds across different arms, leading to select the optimal arm with corresponding clipping bound for AUV to obtain highest expected reward. And throughout the training process, the AUV simultaneously assimilated valuable insights from its interactions, facilitating policy improvement. This iterative learning process culminated in the AUV's proficiency to navigate towards the target point while adeptly avoiding obstacles, thereby marking a transition from initial suboptimal policy to final expert policy. After 1000 training episodes, the AUV has mastered an expert policy, thereby fulfilling the navigation task. The evolution of the AUV's policy, as evidenced by the total reward curve during RL training, is depicted in Fig. 3(a).

Furthermore, in an endeavor to evaluate the superiority of APPO, comparative experiments were conducted utilizing both the Proximal Policy Optimization (PPO) and the Soft Actor-Critic (SAC) algorithms under identical experimental setup. The experiment results are delineated in Fig. 3(a). The curves in Fig. 3(a) revealed that, in preliminary stages, both PPO and SAC exhibited significant policy improvement, achieving reward of 1400 and 1800 after 500 episodes' training, surpassing the Adaptive PPO's reward of 1000.

Nevertheless, the Adaptive PPO demonstrates a superior advantage in policy improvement in the last 500 episodes, evidenced by a consistent increment in reward, ultimately converging to a value of 2941. In contrast, PPO and SAC's reward oscillated around the 2000 and 2400, respectively. These observations underscore APPO's superiority in facilitating navigation and obstacle avoidance for AUV, while showcasing its enhanced training stability.

Ablation Experiments. To assess the impact of different bandit arm numbers on the performance of the APPO algorithm, ablation experiments were conducted with the arm numbers varying from 6 and 12, respectively. Corresponding clipping bounds were defined as [0.005, 0.05, 0.12, 0.16, 0.20, 0.24] for

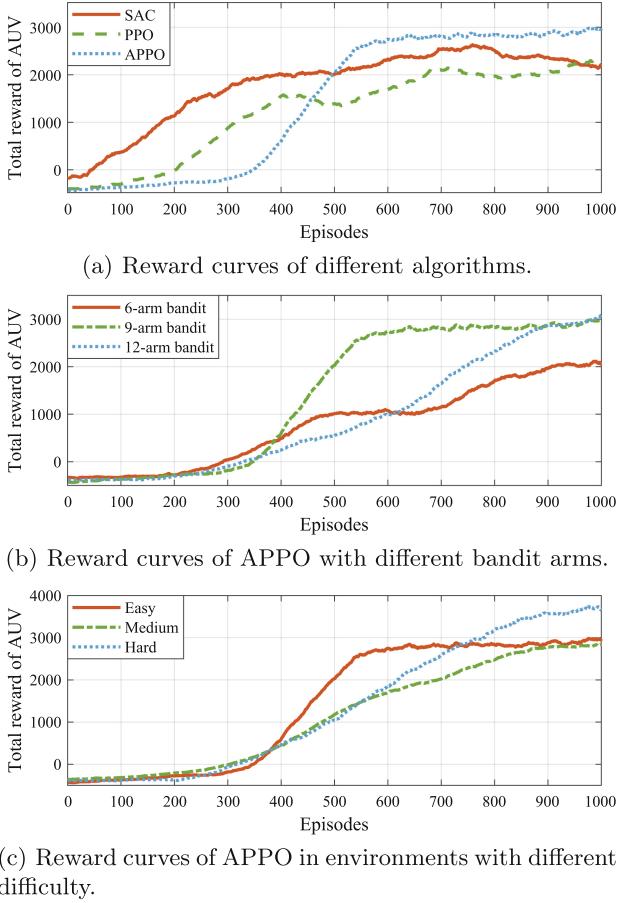


Fig. 3. (a) Reward curves of different algorithms (SAC, PPO and APPO). (b) Reward curves of APPO with different bandit arms (6-arm, 9-arm and 12-arm). (c) Reward curves of APPO in environments with different difficulty (Easy, Medium, Hard).

the 6-arm bandit, while [0.005, 0.01, 0.03, 0.05, 0.08, 0.12, 0.16, 0.17, 0.20, 0.24, 0.28, 0.32] for the 12-arm bandit. These experiments were performed under identical other parameter settings, with the outcome of the reward curves illustrated in Fig. 3(b). By observing Fig. 3(b), it can be found that in the preliminary training stage, the increase speed of reward decreased first and then increased as the number of arms rose. While in the last 500 training episodes, the final reward value varied from 1986, to 2941 and to 2763 when arm number ranged from 6 to 12, respectively, demonstrating an initial increase and subsequent decrease in reward values with rising arm numbers.

This phenomenon suggests that the number of arms influences the balance between exploration and exploitation of the clipping bound in the APPO algo-

rithm. With fewer arms, the algorithm tends to expedite the exploration of clipping bounds, thereby accelerating preliminary-stage policy improvement but potentially leading to premature convergence to local optima, as reflected by lower reward values in the end. Conversely, a bandit with more arm extends the exploration phase, decelerating initial policy improvement but mitigating the risk of suboptimal convergence, hence achieving higher reward values in the final episode. This analysis underscores the significance of selecting optimal number of arms with corresponding clipping bounds to maximize algorithm performance.

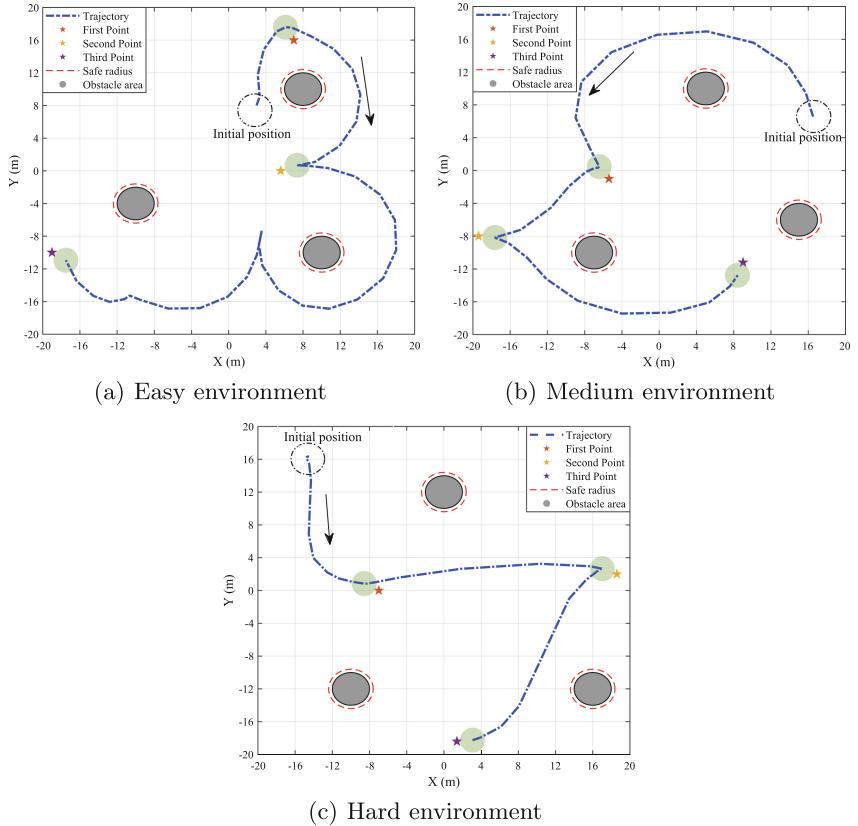


Fig. 4. Trajectories of the AUV in the environment with different difficulty. (a) Trajectories of the AUV in the easy environment. (b) Trajectories of the AUV in the medium environment. (c) Trajectories of the AUV in the hard environment. (The green circles denote the target distance $l_{\max}^{i \leftrightarrow T}$.) (Color figure online)

Environment Generalization Experiments. Furthermore, to verify the generalization capabilities of the APPO algorithm, we changed the environmental

parameters of obstacles within the simulation. Specifically, we designed three environments of different difficulty varying from easy to hard. In the easy environment, the location of each obstacle is fixed, which can be randomly initialized at set intervals in the medium environment. While in the hard environment, the frequency of random initialization increases. Relying on these environments, we conducted simulation experiments employing APPO to train the AUV for navigation and obstacles avoidance, and utilized the trained model to complete the navigation task. The reward curves of training and visualization of trajectories in three different environment are depicted in Fig. 3(c) and Fig. 4, respectively.

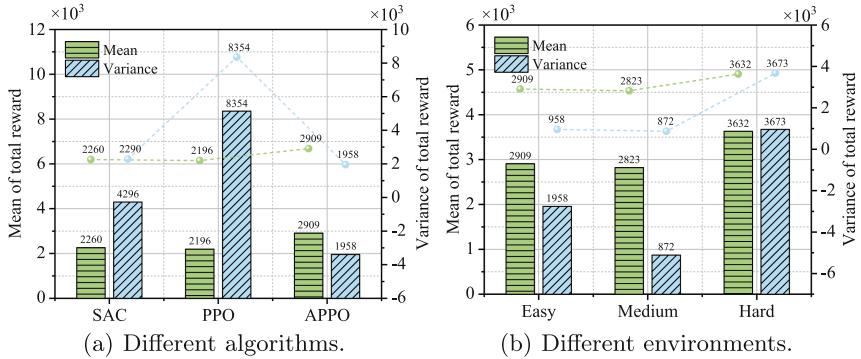


Fig. 5. (a) The mean and variance of total reward using three different algorithms for training (SAC, PPO and APPO). (b) The mean and variance of total reward using APPO for training in environments with different difficulty (Easy, Medium, Hard).

Observations from these results revealed that AUV can obtain the expert policy via APPO for training in three different environments. Based on the trained model via APPO, AUV can adeptly navigate through various environments, consistently achieving high reward outcomes. This performance is indicative of APPO's robust generalization capability, demonstrating effectiveness in adapting to diverse challenges presented by fluctuating environmental conditions.

Finally, we calculated the mean and variance of the total reward from the last 100 episodes of algorithm comparison and environment generalization experiments, respectively, and visualized the results in Fig. 5. The APPO algorithm achieved higher mean while maintained a lower variance of total reward in Fig. 5(a). On the other hand, the AUV trained by APPO all got satisfactory outcomes in three different environments in Fig. 5(b). Through analyzing above observations, we can conclude that the proposed APPO outperforms the baseline algorithms, showcasing superior adaptability, robustness, and generalization. Future work will consider further improving the realism of the simulation, and conduct both simulation and real-world experiments in even more complex tasks.

6 Conclusion

This study investigates the AUV robust navigation problem in complex environments, modeling it as a Markov decision process and solving it using the proposed APPO algorithm. APPO optimizes the selection of clipping bounds in PPO by dynamically exploring and exploiting clipping bounds during online training using a bandit, guiding PPO to make optimal choices at different stages of online training by maximizing the value of the upper confidence bound of each candidate bound. Compared to methods using fixed trust regions, APPO can dynamically respond to the requirements of training tasks with lower computational complexity. In simulation experiments, by gradually changing the environment of AUV navigation tasks from easy to hard, conducting ablation experiments, and comparing with different algorithms, it is found that APPO can efficiently complete tasks in different navigation scenarios, demonstrating superior adaptability, robustness, and generalization.

Acknowledgement. This work of Jingjing Wang was partly supported by the National Natural Science Foundation of China under Grant No. 62071268 and No. 62222101, partly supported by the Young Elite Scientist Sponsorship Program by the China Association for Science and Technology under Grant No. 2020QNRC001, and partly supported by the Fundamental Research Funds for the Central Universities. This work of Yong Ren was partly supported by the National Natural Science Foundation of China under Grant 62127801, partly supported by the National Key Research and Development Program of China under Grant 2020YFD0901000.

References

- Chen, C., Chen, X.Q., Ma, F., Zeng, X.J., Wang, J.: A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Eng.* **189**, 106299 (2019)
- Chu, Z., Wang, F., Lei, T., Luo, C.: Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. *IEEE Trans. Intell. Veh.* **8**(1), 108–120 (2023)
- Gong, Y.J., Huang, T., Ma, Y.N., Jeon, S.W., Zhang, J.: Mtrajplanner: a multiple-trajectory planning algorithm for autonomous underwater vehicles. *IEEE Trans. Intell. Transp. Syst.* **24**(4), 3714–3727 (2023)
- Jiang, P., Song, S., Huang, G.: Attention-based meta-reinforcement learning for tracking control of AUV with time-varying dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(11), 6388–6401 (2022)
- Wang, Z., Zhang, Z., Wang, J., Jiang, C., Wei, W., Ren, Y.: AUV-assisted node repair for IoUT relying on multiagent reinforcement learning. *IEEE Internet Things J.* **11**(3), 4139–4151 (2024)
- Wei, W., Wang, J., Du, J., Fang, Z., Ren, Y., Chen, C.L.P.: Differential game-based deep reinforcement learning in underwater target hunting task. *IEEE Trans. Neural Netw. Learn. Syst.* 1–13 (2023). (early access)
- Wen, J., Yang, J., Wang, T.: Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm. *IEEE Trans. Veh. Technol.* **70**(9), 8529–8544 (2021)

8. Yan, J., Gong, Y., Chen, C., Luo, X., Guan, X.: AUV-aided localization for internet of underwater things: a reinforcement-learning-based method. *IEEE Internet Things J.* **7**(10), 9728–9746 (2020)
9. Yao, J., Li, C., Sun, K., Cai, Y., Li, H., Ouyang, W., Li, H.: NDC-scene: boost monocular 3d semantic scene completion in normalized device coordinates space. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9421–9431. IEEE Computer Society (2023)
10. Yao, J., Qian, Q., Hu, J.: Multi-modal proxy learning towards personalized visual multiple clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14066–14075 (2024)
11. Yao, J., Wu, T., Zhang, X.: Improving depth gradient continuity in transformers: a comparative study on monocular depth estimation with CNN. arXiv preprint [arXiv:2308.08333](https://arxiv.org/abs/2308.08333) (2023)
12. Zeng, Z., Sammut, K., Lian, L., He, F., Lammas, A., Tang, Y.: A comparison of optimization techniques for AUV path planning in environments with ocean currents. *Robot. Auton. Syst.* **82**(C), 61–72 (2016)



MedSiML: A Multilingual Approach for Simplifying Medical Texts

Hardik A. Jain¹(✉) , Chirayu Patel¹, Riyasatali Umatiya¹, Sajib Mistry¹(✉) , Aneesh Krishna¹ , and Amin Beheshti²

¹ School of Electrical Engineering, Computing and Mathematical Sciences,
Curtin University, Perth, Australia

{h.jain2,c.patel28,r.umatiya}@student.curtin.edu.au,
{sajib.mistry,a.krishna}@curtin.edu.au

² School of Computing, Macquarie University, Sydney, Australia
amin.beheshti@mq.edu.au

Abstract. Health literacy is crucial yet often hampered by complex medical terminology. Existing simplification approaches are limited by small, sentence-level, and monolingual datasets. To address this, we introduce MedSiML, a large-scale dataset designed to simplify and translate medical texts into the ten most spoken languages, improving global health literacy. MedSiML includes over 64,000 paragraphs from PubMed, Wikipedia, and Cochrane reviews, simplified into English, Mandarin, Spanish, Arabic, Hindi, Bengali, Portuguese, Russian, Japanese, and Punjabi, with an additional super-simplified English version for those with learning disabilities. We detail MedSiML’s creation, including data sourcing, cleaning, and annotation using the Gemini Flash-1.5 model. We fine-tuned the Text-To-Text Transfer Transformer (T5) base model on this paragraph-level, multilingual data, achieving significant improvements: 10.61% in Recall-Oriented Understudy for Gisting Evaluation 1 (ROUGE1), 11.01% in the Sentence-level Quality Estimation (SARI) score, and 49.1% in semantic similarity over previous state-of-the-art models. Experimental results show that the Flesch-Kincaid (FK) and Automated Readability Index (ARI) readability scores are improved by 0.38 and 1.06, respectively, with no significant changes in the Bilingual Evaluation Understudy (BLEU) score.

Keywords: Multilingual Dataset · Medical Text Simplification · Gemini Flash-1.5 model · Text-To-Text Transfer Transformer (T5)

1 Introduction

Access to high-quality healthcare is a fundamental necessity, yet health literacy—the ability to access, understand, and use health information effectively—remains a significant challenge for many individuals. *Low health literacy* is associated with poorer health outcomes and behaviors¹, and medical texts often contain complex

¹ <https://www.aihw.gov.au/reports/australias-health/health-literacy>.

terminology that requires professional interpretation. Despite the availability of healthcare services, financial constraints and geographical barriers prevent many from receiving necessary medical attention. For instance, high consultation and treatment costs have led to significant percentages of the population abandoning needed healthcare services, particularly among young adults (aged 25 to 34)². Real-time access issues and the socioeconomic disadvantages of living in regional areas further exacerbate this problem.

The internet has emerged as a valuable resource for health information, and in many countries, a significant percentage of the population, especially younger adults, accesses medical advice online. For example, in 2021, 82% of people aged 18 to 34 years in Australia accessed medical advice online³. However, the availability of online medical information does not inherently translate to its comprehensibility, as such information is not always tailored for easy interpretation by laypersons. This creates a gap between the availability of health information and its accessibility to the general population. In multicultural nations with diverse linguistic heritages, the challenge is further compounded by the need for accurate and comprehensible communication in healthcare. For example, in Australia, approximately 22.3% of the population speaks a language other than English at home⁴, underscoring the critical importance of effective translation and interpretation services. While community organizations and hospitals do provide interpretation services, these are often limited to major urban centers and more commonly spoken languages, leaving many without adequate support.

Given these challenges, the development of an automated model for simplifying medical text is of paramount importance. However, this task is fraught with difficulties, primarily due to the lack of large, publicly available datasets that are both multilingual and extend beyond sentence-level simplifications. Existing datasets, such as CLEAR [9], MyTomorrows-Wiki [4], AutoMeTS [20], and MSD Manuals [5], are primarily *sentence-level* and *monolingual* text simplification datasets. These sentence-level datasets present significant limitations when training comprehensive models intended for practical applications that require the simplification of extended texts. While more recent datasets such as Cochrane [7] have introduced paragraph-level simplifications, they still suffer from being noisy, smaller in size with only 4.5k records, and monolingual. Additionally, the introduction of MultiCochrane [10], a multilingual version of the Cochrane dataset supporting four languages, addresses some of these issues but remains limited to sentence-level simplifications. These limitations highlight a critical gap in the resources available for medical text simplification.

To address this gap, our study introduces *MedSiML* (*Medical Simplified Multilingual Text*), a novel and large-scale dataset specifically designed to meet the

² <https://www.abs.gov.au/statistics/health/health-services/patient-experiences/latest-release>.

³ <https://www.statista.com/statistics/1282620/australia-share-of-people-who-accessed-health-services-online-by-age/>.

⁴ <https://www.abs.gov.au/statistics/people/people-and-communities/cultural-diversity-census/latest-release>.

demands of multilingual and paragraph-level medical text simplification. MedSiML comprises over 64,000 paragraphs sourced from PubMed, Wikipedia, and Cochrane clinical reviews, each simplified into the top 10 most spoken languages worldwide: English, Mandarin, Spanish, Arabic, Hindi, Bengali, Portuguese, Russian, Japanese, and Punjabi. Additionally, a super-simplified English version is available to improve comprehension for younger audiences and individuals with learning disabilities. The dataset creation process involved rigorous data cleaning to remove noise and ensure quality, followed by automated annotation using the Flash-1.5 variant of the Gemini model to simplify and translate the texts. This model's capability to handle large inputs and generate accurate simplified outputs ensures the retention of critical information while reducing complexity.

The motivation behind the development of MedSiML lies in the need for a robust, multilingual dataset that can effectively support the training of models aimed at simplifying extended medical texts. Unlike existing sentence-level and monolingual datasets, MedSiML offers a significant improvement in scale, linguistic diversity, and the depth of simplification by operating at the paragraph level. This provides a robust foundation for training models that can address the complexities of real-world medical text simplification for a diverse global audience, thereby significantly enhancing health literacy and accessibility.

To fully leverage the comprehensive capabilities of MedSiML, we fine-tune the Text-To-Text Transfer Transformer (T5) base model on this paragraph-level and multilingual data. The T5 model is known for its versatility and strong performance across various natural language processing tasks. By optimizing it to process and simplify multilingual medical texts, we achieve substantial improvements. The fine-tuning process involves adjusting the model to handle the complex requirements of medical text simplification while ensuring that the simplified outputs retain the essential information from the original texts. By evaluating our fine-tuned T5 model against existing state-of-the-art models, such as those described in Devaraj et al. [7] and Lu et al. [15], we demonstrate significant performance gains across various metrics. Specifically, we observe improvements in metrics such as ROUGE, SARI, and semantic similarity, which are critical for assessing the quality of text simplification. Our model shows a notable increase in accuracy and effectiveness, underscoring the superior quality and utility of the MedSiML dataset. The contribution of the paper is summarized as follows:

- MedSiML, a large-scale dataset with over 64,000 paragraphs from PubMed, Wikipedia, and Cochrane clinical reviews, simplified into the top 10 most spoken languages and a super-simplified English version.
- A fine-tuned T5 base model achieving significant performance improvements over existing models in metrics such as readability, ROUGE, SARI, and semantic similarity.

Table 1. A comparison of Mono and Multilingual Medical Datasets

Datasets	Year	Type	#Records	#Languages
CLEAR	2018	Sentences	663	1
MyTomorrows-Wiki	2019	Sentences	9033	1
AutoMeTS	2020	Sentences	3259	1
MSD-Manuals	2020	Sentences	2551	1
Med-EASi	2023	Sentences	1979	1
Cochrane	2021	Paragraphs	4,459	1
MultiCochrane Clean	2023	Sentences	1,632	4
MultiCochrane Noisy	2023	Sentences	60,058	4
MedSiML	2024	Paragraphs	64,493	10

2 Related Work

2.1 Text Simplification

The two most popular works on text simplification include the Simple Wikipedia sentence-aligned corpus [6, 26] and the Newsela news corpus published by Xu et al. in 2015 [23]. In the medical domain specifically, early text simplification models are based on statistical approaches like WordNet and embeddings [11]. More recently, a paragraph-level unlikelihood-trained model is demonstrated by Devaraj et al. in 2021 [7]. They modify the unlikelihood loss algorithm [22] to explicitly penalize the generation of technical terms by adding a weighted term to the log loss, thereby decreasing the probability assigned to a set of complex tokens derived from a logistic regression model trained to classify document simplicity as shown in Eq. 1. The model is trained on the Cochrane corpus obtained by scraping the Cochrane Reviews website which provides a plain language summary for many scientific articles published in the clinical domain. The same algorithm is subsequently reused in other studies [15] and we present our findings with and without using unlikelihood loss. For evaluating our dataset, we fine-tune the T5-base model (220 million parameters) which allows the same model to be used for multiple tasks by prefixing the task name before the input text [17].

$$-\sum_{t=1}^{|y|} \sum_{j=1}^{|\mathcal{S}|} \mathbb{1}_{s_j, t} w'_j \log(1 - p_\theta(s_j | y_{<t})) \quad (1)$$

2.2 Machine Translation and Multilingual Datasets

The first neural machine translation (NMT) sequence to sequence (Seq2Seq) approach was demonstrated in 2014 [19], which is further improved with the introduction of the attention mechanism [1]. Gehring et al. later demonstrate that convolution-based approaches can perform equivalent or better compared

to RNN-based approaches [8]. These approaches are later simplified by the Transformer architecture, which proposes eliminating all sequential blocks like CNN and RNN and only retaining the attention blocks, leading to improved performance and scalability [21]. Machine translation and text simplification are traditionally solved as separate problems, with only recent efforts being made towards creating multilingual datasets to train a single model that achieves better performance than a pipeline of two separate models. These datasets enable the training of models that can handle complex medical terminologies across different languages, thus making medical information more accessible to diverse populations. The MultiCochrane dataset [10] is a prominent example in this domain. Compiled from Cochrane clinical reviews, it includes plain language summaries translated into four languages, making it easier for non-experts to understand scientific texts. However, MultiCochrane is limited to Cochrane reviews and lacks the diversity in source material and linguistic representation necessary for broader application.

Existing datasets often suffer from limited diversity in sources, insufficient simplification annotations, and restricted language coverage (see Table 1). Our curated MedSiML dataset aims to address these gaps by offering a large-scale, diverse corpus with detailed simplification annotations and translations into the top 10 most spoken languages, thus enhancing the development of multilingual text simplification models. Our dataset is several times larger than the corresponding datasets for medical text simplification.

3 A Multilingual Approach for Simplifying Medical Texts

3.1 The Corpus: Medical Simplified Multilingual Text (MedSiML)

The data was compiled from three different sources to cover a broader domain of knowledge. We sourced 50 thousand abstracts of articles from PubMed using NCBI APIs. The articles are distributed across 5 years from 2020 to 2024 with each year contributing 10 thousand articles. A combination of keywords was used to select the 10 thousand most relevant articles to the biomedical domain. For Wikipedia, the data was sourced from the Hugging Face 2022 Wikipedia corpus. A combination of biomedical keywords was again used to extract the most relevant articles. The Cochrane dataset, derived from Devaraj et al.’s work, added 22,000 records. In total, 110,000 paragraphs were collected. Table 2 highlights the major sources covered.

Table 2. Distribution of records in MedSiML based on the sources.

Source	#Records	Percentage
PubMed (abstracts)	30,445	47.2
Wikipedia (biomedical articles)	21,340	33.1
Cochrane (clinical reviews)	12,708	19.7
Total	64,493	100

Algorithm 1. Processing markdown records for MedSiML

```

procedure PROCESS-DATA(recs, abs, langs)
    procs ← empty list
    fails ← empty list
    for each idx in recs do
        md ← GETREC(idx)
        secs ← SPLIT(md)
        json ← {'orig' : abs[idx]}
        i ← 1
        while i < LEN(secs) – 1 do
            lang ← EXTLANG(secs, i)
            if lang not in langs then
                break
            ni ← FINDNEXT(secs, i, langs)
            if lang not in json then
                text ← CONCAT(secs, i, ni)
                json[lang] ← text
            i ← ni
            l_count ← COUNTLANG(json)
            if l_count  $\geq$  0 then
                APPEND(procs, json)
            else
                APPEND(fails, idx)
    return procs, fails

```

The collected data underwent rigorous cleaning to remove HTML, CSS, non-ASCII characters, and duplicate records. The cleaning process aimed to eliminate noise and ensure the dataset's quality and usability for training models. The removal of HTML and CSS elements was crucial to avoid introducing irrelevant information that could confuse the simplification models. Non-ASCII characters were either removed or transliterated to maintain consistency. Duplicate records were filtered out to prevent redundancy, and records with inappropriate lengths were excluded to focus on meaningful paragraph-level simplification.

3.2 Automated Annotation Process

Existing medical text simplification datasets often rely on sentence alignment techniques such as cosine similarity, which are robust but often introduce noise and inaccuracies. To overcome these limitations, we employed the Gemini Flash-1.5 large language model, capable of processing up to a million tokens with an 8k output limit. This model was used to simplify the corpus into ten languages, with a super simplified English version as shown in Algorithm 1. We observed that the simplified records generated by the model were highly accurate and much more reliable than records in existing datasets, representing a significant advancement in the field. We exclude simplifications that have a very low cosine

original	english simplified	english super simplified	mandarin simplified	spanish simplified	arabic simplified	hindi simplified	bengali simplified	portuguese simplified	russian simplified	Japanese simplified	punjabi simplified
Public trust in physicians has declined over ...	People are losing trust in doctors. To fix this...	Doctors are finding it harder to get people to... ... between patients and doctors. For...	人们对医生的信任度在过去50年里一直在下降。未来医生需要修复患者与医生之间的信任关系。	La confianza pública en los médicos ha disminuido...	لُغَةُ اَنْجَفَتْ ٥٠ مِنْ اَعْمَالِ الْمُؤْمِنِينَ اَلْأَطْهَارِ عَلَى الْحَسَنِ...	पिछले 50 वर्षों में चिकित्सकों का अनुसरण घट गया है। जनसाधारण के लोगों का भरोसा... आशा करते...	A confiança pública nos médicos diminuiu nos últimos 50 anos... A confiança pública nos médicos diminuiu nos últimos 50 anos...	Доверие к врачам в обществе снизилось за последние...	過去50年で、医師に対する国民の信頼は下落しました。未来の医師は、患者と医師の信頼関係を修復...	ਪਿੱਛਲੇ 50 ਸਾਲਾਂ ਵਿਚ ਡਾਕਤਾਂ ਦੇ ਲੋਟਾਂ ਦਾ ਭਰੋਸਾ...	
This study evaluates changes in practice patter...	This research looks at how facelift surgery pr...	This study looked at how facelift surgery has...	手术的做法是如何变化的。 这项研究考察了15年来拉皮手术的做法是如何变化的。 该研究使用了美国整形外科委员会从20...	Este estudio evalúa cómo han cambiado las prácticas...	هَذِهِ الْمُرْسَلَاتُ تَعْرِفُ بِهِنْدِيفِ الْمُهَاجِرِينَ إِلَى الْقِبَلَةِ الْعَلَيِّينَ الَّذِينَ طَارُوا ...	यह अध्ययन 15 सालों में फेसलिफ्ट प्रैक्टिस के लिए बदलाव की विशेषता का प्रध...	Este estudio avalia a mudança nos padrões de pr...	исследование оценивает изменения в практике...	本研究は、アメリカ整形外科協会が実施した15年間の調査結果をもとに、整形外科手術の実践がどのように変化したかを評価するものです。	ਜਾਪਾਨ ਵਿਚ ਮਹਾਨ 15 ਸਾਲਾਂ ਵਿਚ ਫੇਸਲਿਫਟ ਸਾਹਮਣੀ ਦੇ ਅਧਿਕਾਰਾਂ...	
Current research on prostate cancer is heavily...	Doctors are focused on finding prostate cancer...	Doctors are working on finding prostate cancer...	医生们专注于早期发现前列腺癌和开发新的治疗方法。 但他们没有足够地研究前列腺癌。	Los médicos se centran en la detección temprana...	مُكَثِّفُونَ عَلَى الْكِسْبِ الْمُبْكِرِ مِنْ سَرْطَانِ الْمُرْسَلَاتِ...	ڈاکٹਰز کسرا پاکیستانی پاپیلੋਨ... آئرلند...	ডাক্তাররা প্রেরণে কাঞ্চাগুরু প্রক্রিয়া নির্দেশ করছে...	Врачи сосредоточены на ранней диагностике рака...	医師は、前立腺がんの早期発見と新しい治療法の開発に注力しています。しかし、彼らは...	ਡਾਕਟਰ ਪ੍ਰਾਤਿਸ਼ਠਾ ਕੇਸ਼ ਵਿਖੇ ਸਾਡੇ ਪੜਾ ਲੁਕਾਉਣ ਵਾਲੇ...	

Fig. 1. Snapshot of Dataset produced

similarity with the original text. Some records were omitted due to safety reasons as explained in Subsect. 4.4.

We observed the Gemini Flash 1.5 Large Language Model often provides syntactically incorrect JSON outputs that could not be parsed and thus used its markdown output. We passed multiple inputs to the model in a single request and retrieved corresponding outputs in a single response. Algorithm 1, *Process-Data*, processes simplified markdown records generated by the Gemini model. It initializes two lists: procs for successfully processed records and fails for those that fail. For each record, the markdown data is retrieved and split into sections, and a JSON object is initialized with the original abstract. The algorithm iterates through the sections, extracting the language and finding the next section with the same language. If the language is valid, the sections are concatenated to the JSON object. Figure 1 depicts a snapshot of the dataset produced based on the above annotation process.

3.3 Paragraph Level Text Simplification Using T5-Base Model

We apply the fine-tuned T5-base model using our MedSiML dataset for text simplification. The MedSiML dataset is split into training, validation, and testing sets in a 99:0.5:0.5 ratio to ensure ample training data and reliable evaluation metrics. The records are shuffled to ensure a comparable distribution, and each input record is prefixed with a task-specific label, such as “simplify medical text:”, which guides the model in performing the intended task. The records are

then tokenized using a fast tokenizer from Hugging Face, converting the text into a sequence of tokens that the model can process efficiently.

The fine-tuning process begins with initializing the pre-trained T5-base model, which has a robust foundational understanding of language from its initial training. The fine-tuning is configured with optimal hyperparameters, such as learning rate, batch size, and the number of epochs. During training, the input tokens are passed through the encoder to generate contextualized embeddings, and the decoder generates the simplified text based on these embeddings. The loss is calculated by comparing the generated text with the actual simplified text, guiding the model in minimizing errors through backpropagation.

4 Experiments and Results

We compare the efficiency and the proposed approach with two state-of-art paragraph level text simplification approaches: a) BART trained using unlikelihood loss on the Cochrane Clinical Reviews Corpus [7] and b) NaPSS which is a multi-model pipeline based approach involving summarization followed by simplification [15]. We provide a comparative analysis in terms of readability, lexical accuracy, and quality of simplification relative to baseline models. The evaluation incorporated both automated metrics and detailed human assessment. All models in the experiment are trained on 2x RTX Quadro 6000 (24 GB each) using the torch data parallel module and took four days to finish leading to an estimated 10.37 kgCO₂eq emission. Estimations were obtained using the Machine Learning Impact calculator [13].

Hyperparameters: the batch size used is 16, precision used is FP16 (or half precision). We retain the optimizer AdamW and learning rate 2e-5 with a weight decay of 0.01 same as used for T5 pre-training.

Although we have curated a multilingual dataset in 10 languages, we were only able to finish training the models in the English language due to time and computational constraints. The test dataset consists of 297 records which are sourced from Cochrane to ensure consistency in the comparison. The code can be accessed on GitHub at <https://github.com/nepython/MedSiML>.

4.1 Performance Analysis on Readability Metrics

The readability of text simplification models is crucial for ensuring accessibility. Readability metrics assess how easy or difficult it is to read a sentence based on its structure and vocabulary. Readability metrics give higher scores for text which are more difficult to read and lower scores for text which are easier to read. Thus, while measuring readability improvements, a lower score is better whereas for other metrics a higher score is better. We employ the Flesch-Kincaid (FK) [12] and Automated Readability Index (ARI) [18] metrics for this purpose which were also employed in the baseline studies. Algorithm 2 depicts the steps for calculating the FK score. The FK score is influenced by the average sentence length and the average number of syllables per word. Longer paragraphs tend

Algorithm 2. FK Score calculation

```

procedure FK-READ(para)
    words  $\leftarrow$  WORDCOUNT(para)
    sents  $\leftarrow$  SENTENCECOUNT(para)
    sylls  $\leftarrow$  SYLLABLECOUNT(para)
    asl  $\leftarrow$  words / sents
    asw  $\leftarrow$  sylls / words
    fk  $\leftarrow$   $0.39 \times asl + 11.8 \times asw - 15.59$ 
    return fk

```

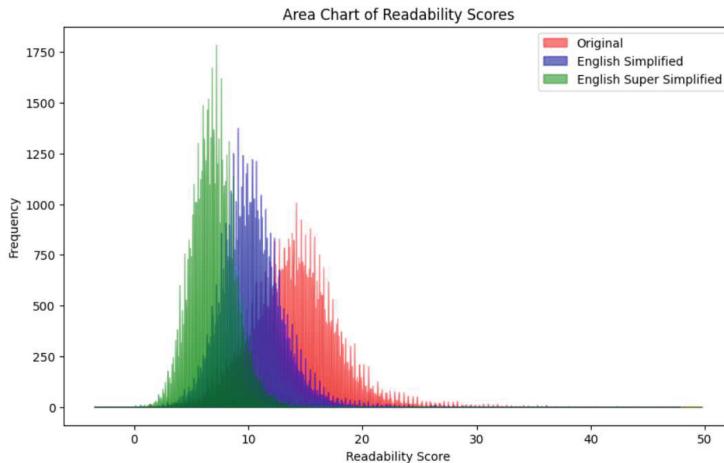


Fig. 2. Readability of the training dataset between the original text and the simplified, super simplified records measured by FK index as shown in Algorithm 2.

to have more sentences and words, leading to a higher average sentence length, which affects the FK score calculation.

The Gemini model's ability to handle large input and output sizes allowed the proposed approach to provide detailed instructions for retaining the original text's voice, numbers, and other essential details while simplifying the content. Existing datasets are unable to retain these important factors in the simplified text. We observe in the dataset, readability of the text from the three sources slightly varies with PubMed biomedical abstracts having the highest median readability at 14.83, followed by Cochrane clinical reviews at 13.2 and Wikipedia biomedical articles at 12.9 as shown in Table 3. As shown in Fig. 2, the overall FK readability in the simplified records is reduced from 14.24 to 10.35, that is an improvement of 27.3% (Table 4).

A very small fraction of records were observed to contain excessive repetitions with minor noticeable repetitions observed for almost 10% of the records as shown in Table 5. We also observe that there were only a few records that are longer in comparison to the original text. Coincidentally, these are the same records which have excessive repetition in them. There was no noticeable reduc-

Table 3. Comparison of readability across original, simplified, and super simplified text for different sources in the dataset.

Datasets	original				simplified				super simplified			
	mean	median	max	min	mean	median	max	min	mean	median	max	min
Cochrane	13.34	13.20	72.80	1.70	10.10	9.80	36.40	2.90	6.72	6.50	28.60	-2.70
PubMed	14.83	14.80	42.70	-3.50	10.66	10.50	27.20	0.50	7.39	7.30	20.40	-3.50
Wikipedia	13.95	12.90	295.40	0.30	10.07	9.50	77.40	-3.50	6.76	6.40	86.00	-3.50

Table 4. We observe that FK readability reported for simplified text is higher than complex text although the simplified text is easier to read.

Text	Score
Complex abstract: Four small randomised controlled trials involving 728 allocated/224 analysed participants met our inclusion criteria. These trials had a high risk of bias. Drug companies sponsored two of the trials. We were unable to pool the data due to the heterogeneity in outcome definitions and the different antibiotics used. The included trials compared the following antibiotic schedules. The first trial compared quinolone (levofloxacin) plus standard treatment (anti-staphylococcal penicillin (cloxacillin or dicloxacillin), aminoglycoside (tobramycin or netilmicin) and rifampicin) versus standard treatment alone reporting uncertain effects on all-cause mortality (8/31 (26%) with levofloxacin plus standard treatment versus 9/39 (23%) with standard treatment alone; RR 1.12, 95% CI 0.49 to 2.56, very low quality evidence). The second trial compared daptomycin versus low-dose gentamicin plus an anti-staphylococcal penicillin (nafcillin, oxacillin or flucloxacillin) or vancomycin. This showed uncertain effects in terms of cure rates (9/28 (32.1%) with daptomycin versus 9/25 (36%) with low-dose gentamicin plus anti-staphylococcal penicillin or vancomycin, RR 0.89 95% CI 0.42 to 1.89; very low quality evidence). The third trial compared cloxacillin plus gentamicin with a glycopeptide (vancomycin or teicoplanin) plus gentamicin. In participants receiving gentamycin plus glycopeptide only 13/23 (56%) were cured versus 11/11 (100%) receiving cloxacillin plus gentamicin (RR 0.59, 95% CI 0.40 to 0.85; very low quality evidence). The fourth trial compared ceftriaxone plus gentamicin versus ceftriaxone alone and found no conclusive differences in terms of cure (15/34 (44%) with ceftriaxone plus gentamicin versus 21/33 (64%) with ceftriaxone alone, RR 0.69, 95% CI 0.44 to 1.10; very low quality evidence). The trials reported adverse events, need for cardiac surgical interventions, uncontrolled infection and relapse of endocarditis and found no conclusive differences between comparison groups (very low quality evidence). No trials assessed septic emboli or quality of life.	8.8
Simplified text: We looked at four small studies that tested different antibiotics for treating a serious heart infection called infective endocarditis. The studies had some serious problems, and two of them were funded by drug companies. The studies used different antibiotics and measured different outcomes, so we couldn't combine their results. Overall, the studies didn't show any clear benefit of one antibiotic over another for treating this infection. They also didn't find any clear differences in side effects, the need for heart surgery, or infection coming back.	11.1

tion in length for about a quarter of the records and the rest were slightly or much shorter than the original. Every 1 in 5 records had some essential information missing in the generated simplified text compared to the original text.

4.2 Performance Analysis on Lexical, Content and Semantic Similarity

Lexical similarity, which measures the overlap of n-grams between the candidate and reference texts, is evaluated using the Bilingual Evaluation Understudy (BLEU) metric [16]. Additionally, content similarity is assessed through Recall-Oriented Understudy for Gisting Evaluation (ROUGE), where ROUGE-L represents the Longest Common Subsequence and ROUGE-N represents n-gram overlap [14].

Algorithm 3. calculating cosine similarity for sentence alignment in MedSiML

```

procedure COSINE-SIM(para1, para2)
    tokens1  $\leftarrow$  TOKENIZE(para1)
    tokens2  $\leftarrow$  TOKENIZE(para2)
    embed1  $\leftarrow$  EMBED(tokens1)
    embed2  $\leftarrow$  EMBED(tokens2)
    dot_prod  $\leftarrow$  DOT(embed1, embed2)
    norm1  $\leftarrow$  NORM(embed1)
    norm2  $\leftarrow$  NORM(embed2)
    cosine_sim  $\leftarrow$   $\frac{\text{dot\_prod}}{\text{norm1} \times \text{norm2}}$ 
return cosine_sim

```

We utilize the Sentence-level Quality Estimation (SARI) metric [24], which was specially created to score simplification tasks and considers lexical, syntactic, and semantic factors. Semantic similarity examines how closely the meaning of the simplified text matches that of the original. These metrics often use cosine similarity. Measuring the similarity between sentences or paragraphs is often used for producing sentence aligned text simplification dataset as it ensures that corresponding texts are matched correctly. Algorithm 3 calculates the cosine similarity between two paragraphs. Cosine similarity measures the cosine of the angle between two non-zero vectors in an inner product space. The paragraphs are first tokenized, then converted to embeddings, and finally, the cosine similarity is computed. We use BertScore [25] for evaluating semantic similarity by comparing the contextual SciBERT embedding [3] of candidate and reference texts.

In Table 6, we present the performance of our model across these metrics compared to other models trained on similar corpora. This evaluation provides insights into our fine-tuned model's efficacy in simplifying medical text while maintaining semantic accuracy and natural language expression. Our models perform exceptionally well compared to the previously published models across a wide variety of metrics. Noticeably, the T5-base (UL) improves the readability to 8.69 which is significantly better than the training dataset's 10.35, an almost 20 % improvement over Gemini. Similar to these studies we also train our models with unlikelihood loss in addition to standard cross-entropy loss. The unlikelihood loss

Table 5. The last sentence has the same meaning as its predecessor in the generated text for this record.

Simplified Text: This study looked at whether a test called procalcitonin, C-reactive protein (CRP), and erythrocyte sedimentation rate (ESR) could accurately tell if a child has a urinary tract infection (UTI) compared to a scan done within 30 days of the UTI. The researchers wanted to see if these tests could accurately distinguish children with cystitis from children with pyelonephritis. They also wanted to see if the characteristics of the child and the study affected the accuracy of these tests. They found that procalcitonin, CRP, and ESR could be compared to a scan done within 30 days of the UTI in children with pyelonephritis. **They also found that procalcitonin, CRP, and ESR could be used instead of a scan done within 30 days of the UTI.**

Table 6. Performance comparison of the fine-tuned models against existing approaches.

Models	Readability		Lexical				Simplification	Semantic
	FK	ARI	Rouge1	Rouge2	Rouge-L	BLEU	SARI	BertScore
UL-BART [7]	11.97	13.73	38.00	14.00	36.00	39.0	40.00	N/A
UL-BART [15]	9.30	12.40	43.25	16.36	40.22	7.9	40.08	24.64
NapSS [15]	10.97	14.27	48.05	19.94	44.76	12.3	40.37	25.73
NapSS (+UL) [15]	8.67	11.80	45.39	16.77	42.53	9.1	41.12	23.13
NapSS (-Prompt) [15]	9.86	13.06	45.62	20.01	44.83	12.1	39.68	25.57
T5-base	11.33	13.76	58.66	33.75	45.47	27.7	52.13	74.83
T5-base (super)	10.01	12.1	31.68	10.68	22.66	7.59	43.58	63.51
T5-base (UL)	8.29	10.74	35.34	14.17	26.31	9.46	43.92	62.33

is helpful in significantly improving the readability score for the model. In the manual evaluation, we observe that 96% of the records have correct grammar and syntax with no spelling or punctuation errors. We also observe that almost 84% of the generated simplified records had consistency with the original complex records for the medical terms.

4.3 Evaluating Hallucination

While most studies including ours have increasingly started relying on automated metrics to evaluate the performance of trained models, these metrics have limitations on evaluating hallucination in comparison to human evaluation. Hallucination refers to the event of generated text introducing facts or details that aren't present within the original text. We observe that almost 26% of predictions have some form of hallucination ranging from a few words to sentences. Around 8% of the predictions contain major inaccuracies which changed the overall meaning, whereas 18% contain minor inaccuracies which retained the overall meaning.

Table 7 shows an example record with hallucination where the two ultrasound methods have been interchanged. Ultrasound scan is a form of diagnostic ultrasound that focuses on imaging and diagnostics, while therapeutic ultrasound is aimed at treatment and rehabilitation. So, they are not the same, but both utilise sound waves for different purposes⁵.

4.4 Limitations

The primary limitation of this study is the exclusion of many records from the original 110k set which were blocked by Gemini as they discussed topics that are deemed unsafe or by our filters for being incomplete. Gemini has a filtering process in place that will only block text containing any content on harassment,

⁵ <https://www.iuslondon.co.uk/blog/understanding-the-differences-between-therapeutic-and-diagnostic-ultrasound/>.

Table 7. Hallucination: Therapeutic ultrasound is replaced by an Ultrasound scan in the model’s prediction.

Original Text: The evidence from this systematic review is uncertain regarding the effect of [therapeutic ultrasound](#) on pain in individuals with chronic non-specific LBP. Whilst there is some evidence that therapeutic ultrasound may have a small effect on improving low back function in the short term compared to placebo, the certainty of evidence is very low. The true effect is likely to be substantially different. There are few high-quality randomised trials, and the available trials were very small. The current evidence does not support the use of [therapeutic ultrasound](#) in the management of chronic LBP.

Simplified Text: This study looked at whether [ultrasound scans](#) can help people with chronic low back pain. The study found that ultrasound scans might have a small effect on improving low back function in the short term compared to a fake scan, but the evidence is very weak. The true effect is likely to be much different. There aren’t many high-quality studies, and the studies we found were very small. The current evidence doesn’t support using [ultrasound scans](#) to treat chronic low back pain.

Table 8. Top words in the records blocked by Gemini due to safety reasons as filtered using the Hurtlex dictionary.

Top Words in Blocked Records										
con	rat	red	ass	imp	low	ring	die	cur	ire	
kin	lie	mean	different	nag	rot	roach	loo	scribe	rag	
rip	cop	bed	disorder	hot	quest	gula	rage	black	problem	
tool	mad	blood	cad	usual	face	race	severe	poor	story	
sly	pirate	score	mark	rout	power	firm	force	sod	kill	
rape	tail	infant	family	agent	bus	member	nut	beg	sept	
blind	degree	minister	white	pass	lot	head	infants	moth	faction	
lust	fix	imitation	tart	hyl	fib	stain	nance	neonatal	hole	

hate speech, sexually explicit or dangerous. We have filtered the top words used in the blocked records using a words dictionary of hate speech [2]. We have explored the topics that are being discussed in the records not processed by Gemini and present the top concerning words in Table 8. Although on reading these blocked records, it was concluded that the content in them was not harmful and the studies were exploring or describing the topic.

5 Conclusion

The MedSiML dataset represents a significant advancement in the field of multilingual medical text simplification, addressing a critical gap in healthcare communication. The dataset’s larger size and reduced noise improve the performance of models trained on it, as demonstrated through various evaluations. Models such as the Text-To-Text Transfer Transformer (T5) base model, fine-tuned on the MedSiML dataset, show substantial performance improvements, outperforming previous state-of-the-art models like UL-BART and various NapSS models.

Notable improvements include 10.61% in ROUGE1, 11.01% in SARI score, and 49.1% in semantic similarity. The model trained with unlikelihood loss also shows readability enhancements in FK and ARI scores.

Despite its advancements, MedSiML has limitations, including the exclusion of some records due to safety concerns and computational constraints limiting the evaluation to English. Future work should focus on including these records, training models in all represented languages, and further refining models to reduce hallucinations and improve text accuracy. Expanding the dataset to cover more specialized medical domains could also enhance its applicability. This research bridges disparities in health literacy and paves the way for innovations in healthcare communication, contributing to better health outcomes for diverse and multilingual communities worldwide.

Disclosure of Interests. The authors declare they have no competing interests.

References

- Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (2014)
- Bassignana, E., Basile, V., Patti, V.: Hurtlex: A Multilingual Lexicon of Words to Hurt, pp. 51–56. Accademia University Press, Torino (2018)
- 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). Association for Computational Linguistics, Stroudsburg, PA, USA (2019)
- van den Bercken, L., Sips, R.J., Lofi, C.: Evaluating neural text simplification in the medical domain. In: The World Wide Web Conference. ACM, New York, NY, USA (2019)
- Cao, Y., Shui, R., Pan, L., Kan, M.Y., Liu, Z., Chua, T.S.: Expertise style transfer: a new task towards better communication between experts and laymen. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA (2020)
- Coster, W., Kauchak, D.: Simple English Wikipedia: a new text simplification task. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 665–669. Association for Computational Linguistics, Portland, Oregon, USA (2011)
- Devaraj, A., Marshall, I., Wallace, B., Li, J.J.: Paragraph-level simplification of medical texts. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Stroudsburg, PA, USA (2021)
- Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional Sequence to Sequence Learning (2017). <https://proceedings.mlr.press/v70/gehring17a.html>
- Grabar, N., Cardon, R.: Clear – simple corpus for medical French. In: Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA). Association for Computational Linguistics, Stroudsburg, PA, USA (2018)

10. Joseph, S., et al.: Multilingual simplification of medical texts. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA (2023)
11. Kauchak, D., Leroy, G.: A web-based medical text simplification tool. In: Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences (2020)
12. Kincaid, J.P.: Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel. Institute for Simulation and Training (1975)
13. Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the Carbon Emissions of Machine Learning (2019). <https://arxiv.org/abs/1910.09700>
14. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (2004)
15. Lu, J., Li, J., Wallace, B., He, Y., Pergola, G.: NaPSS: paragraph-level medical text simplification via narrative prompting and sentence-matching summarization. In: Findings of the Association for Computational Linguistics: EACL 2023. Association for Computational Linguistics, Stroudsburg, PA, USA (2023)
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 2002. Association for Computational Linguistics, Morristown, NJ, USA (2002)
17. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2019)
18. Smith, E.A., Senter, R.J.: Automated readability index. AMRL-TR. Aerospace Medical Research Laboratories (U.S.), pp. 1–14 (1967)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014)
20. Van, H., Kauchak, D., Leroy, G.: AutoMeTS: the autocomplete for medical text simplification. In: Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Stroudsburg, PA, USA (2020)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
22. Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., Weston, J.: Neural text generation with unlikelihood training. arXiv preprint [arXiv:1908.04319](https://arxiv.org/abs/1908.04319) (2019)
23. Xu, W., Callison-Burch, C., Napoles, C.: Problems in current text simplification research: new data can help. *Trans. Assoc. Comput. Linguist.* **3**, 283–297 (2015)
24. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. *Trans. Assoc. Comput. Linguist.* **4**, 401–415 (2016)
25. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating Text Generation with BERT (2019). <https://arxiv.org/abs/1904.09675>
26. Zhu, Z., Bernhard, D., Gurevych, I.: A monolingual tree-based translation model for sentence simplification. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1353–1361. Association for Computational Linguistics, USA (2010)



A Study on Time-Resilient Features for Detecting TLS Encrypted Malware Traffic

Kaisei Fujiwara¹, Akira Yamada¹ , Seiichi Ozawa¹ , and Chanho Park²

¹ Kobe University, Kobe, Japan

ozawasei@kobe-u.ac.jp

² LY Corporation, Tokyo, Japan

Abstract. With TLS encryption becoming commonplace in today’s Internet, attackers can easily conceal their malicious activities; that is, they can easily hide malware’s Command and Control (C2) communication or Remote Access Trojan (RAT) traffic. While conventional works using machine learning attempt to detect malicious TLS-encrypted traffic mainly based on flow statistics and features of TLS metadata, they present a limitation in time resiliency. Thus, they lack robustness against time changes in both malicious and benign traffic, resulting in not long-lasting high accuracy in detection. This paper explores new time-resilient features that sustain high accuracy in the detection of TLS-encrypted malware traffic. Our proposed features utilize domain and URL reputation services as references and employ the internal structure of TLS certificates to extract characteristics of encrypted malware. In addition, a multi-view approach is introduced to extract features on sequence of packet lengths and time (SPLT). The evaluation of proposed time-resilient features is carried out using the five-year-long malware datasets. The experimental results reveal that these features are robust against the time evolution of malware activities at least for five years.

Keywords: Cybersecurity · Machine Learning · Cyberattack Detection · Encrypted Malware Traffic · Transport Layer Security

1 Introduction

The Internet has become essential in our daily lives, as we engage in various activities such as communication and shopping. Ensuring the confidentiality of the information we handle has become a crucial issue. In response to such concerns, the SSL (Secure Socket Layer) protocol and its successor, the TLS (Transport Layer Security) protocol, have been developed. One of the most common instances we encounter is HTTPS communication, which uses TLS encryption on top of the application layer protocol, HTTP (Hyper Text Transfer Protocol). The use of HTTPS can be confirmed through the address bar, with some

browsers displaying indicators to signify that the connection is secure. Encryption has provided benefits for privacy protection to companies, governments, and individuals, and the use of TLS is expected to continue to increase.

However, the benefits of encryption also apply to attackers who carry out cyberattacks. It has been reported that they exploit TLS to communicate with Command and Control (C2) servers, hide their activities by receiving command instructions, and steal sensitive data from infected devices [1]. One method to mitigate such attacks might be to perform decryption and inspect the content. Obviously, however, this approach requires heavy computations, and further brings up privacy issues. Due to these reasons, there is a demand for detecting malicious traffic, such as malware, without decrypting packets.

In response to such challenges, various features and detection models utilizing machine learning have been proposed to distinguish between benign and malicious traffic. However, under the circumstances that new malwares are emerged frequently and attackers intentionally attempt to evade detection, it is important that adopted features must be difficult for attackers to evade their malicious activities. The objective of this paper is to share insights for building robust detection models by analyzing the trends in features of malicious traffic, including the proposal of additional effective features. More concretely, we attempt to propose long-lasting effective features by considering additional information about certificates, information obtained from third parties, and features related to certificate costs. In addition, a multi-view machine learning approach is introduced to extract features on sequence of packet lengths and time (SPLT).

This paper consists of the following sections. In Sect. 2, we provide an overview of related work and discuss our perspectives on aspects that have not been deeply addressed in recent studies. In Sect. 3, we describe the proposed features and the structure of the detection model. Section 4 presents the experimental results to validate the effectiveness of the proposed features and models. We analyze and discuss the significant features for classifying malicious traffic based on temporal trends. Finally, in Sect. 5, we summarize the entire research and outline the current challenges and future directions.

2 Related Work

Anderson et al. [1] conducted extensive research on TLS malware traffic and proposed a machine learning approach to adopting the following features for detecting malicious TLS traffic without decryption.

TLS Metadata Flow features are defined from the metadata of packets such as ClientHello, ServerHello, and Certificate, including CipherSuite, TLS Extensions, and EllipticCurve, as well as the validity period of certificates and whether the server certificate is self-signed.

Flow Metadata Another flow features are defined based on the traffic statistics of a flow that are unrelated to protocols. The traffic statistics (e.g., packet length and counts) can be measured through NetFlow, and considered as a traffic feature.

Sequence of Packet Lengths and Time (SPLT) This feature is obtained through the frequency analysis of packets within a flow. Let us consider the two histograms of packet lengths and packet arrival time, each of which is composed of k -bins. Then, a matrix A is constructed to represent the transitional frequency from the i th to the j th bins, where each entry $A[i, j]$ corresponds to the frequency count of packet length or packet arrival time. When making a feature vector from SPLT, a matrix A is flattened by rows.

Along with the above feature engineering approaches, various end-to-end deep learning models [2–4] such as Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been invented to explore high-performance in the malicious encrypted traffic detection. Gomez et al. [4] proposed a method to detect and cluster similar TLS flows using selected features from TLS clients, TLS servers, TLS certificates, and encrypted payloads through unsupervised models. Lee et al. [5] proposed new TLS features for detecting malicious traffic, showing that most malware uses a limited number of TLS extensions, cipher suites, and self-signed certificates based on their analysis. Afzal et al. [6] demonstrated high accuracy and recall rates with a small number of features for detecting malicious TLS traffic and discussed reducing false detection rates. Liu et al. [7] proposed DeepTLS, a feature extraction system that rapidly extracts comprehensive features using machine learning to analyze malicious TLS traffic. Zheng et al. [8] proposed a malicious encrypted traffic identification scheme MET-FMF (Malicious Encrypted Traffic identification based on Fine-grained Multi-feature Fusion) integrated by multi-granularity network (MGN). In this method, packet header and data are processed as an image with CNN model.

There have been proposed several works on the detection of unknown malicious traffic due to the appearance of new malware families. He et al. [9] proposed a method to detect only one sample, considering the difficulty of obtaining a large number of annotated samples. Zhou et al. [10] demonstrated the potential of unsupervised learning neural network methods for feature extraction and clustering of new unknown malicious traffic. Furthermore, Kim et al. [11] proposed a method to visualize TLS-encrypted flows, and SVM and CNN were used to achieve high accuracy in classifying malware families.

As mentioned above, many studies on malicious cryptographic traffic detection use static features defined from TLS metadata and packet flow traffic statistics. The former TLS metadata of malicious cryptographic traffic differs from that of normal traffic because of incomplete TLS configuration by the malware creator or incomplete attack tools. In addition, there may be other limitations due to the cost and configuration effort required for the attack. However, these limitations from the economic point of view and inadequate TLS settings are no longer effective features to distinguish malicious cryptographic communications from benign ones, due to the improvement of attack tools and the spread of inexpensive TLS certificates. Considering that the effective features of malicious communications change depending on the countermeasures taken by attackers to evade detection, it is necessary to explore and exploit highly invariant features

that cannot be hidden in order to achieve the attackers' goals (i.e., vulnerability search, infection spreading, etc.). For this purpose, Anderson et al. proposed SPLT and demonstrated its effectiveness. However, since the sensitivity matrix is flattened by rows, the features of temporal variation in packet length and packet arrival time are not well represented.

In this paper, we first select features from the existing TSL metadata and Flow metadata that are still valid at this point in time. We also define new features that take into account economical efficiency from the attacker's point of view and changes in the trend of cryptographic schemes. Furthermore, in order to capture the characteristics of temporal variation in the SPLT features defined by Anderson et al. in more detail, we incorporate a multi-view approach in which the sensitivity matrix can be treated as an image for feature extraction. By doing this, we aim to mitigate the temporal reduction in detection performance due to countermeasures taken by attackers.

3 Robust Detection for Encrypted Malicious Traffic

3.1 Time-Resilient Features

As discussed in Sect. 2, this work is aiming for the robust detection of encrypted malicious traffic which can be long-lasting in effectiveness without the decryption of payloads. First, let us select effective features from the Anderson's work [1], which are relatively difficult for attackers to conceal their malicious activities, so that time-resilient property in features can be lasted over time. Table 1 shows our selection of time-resilient features.

As seen in Table 1(a), there are three categories of time-resilient features mentioned in 2: TLS Metadata, Flow Metadata, and SPLT. The TLS Metadata consists of the four features on cipher suite and certificate information. The first feature in Table 1 shows the cipher suite selected from 145 suites with a 16-bit binary vector, and if there is no cipher suite matched with 145 suites, it is represented with the code 'FFFF'. The second to fifth features in the TLS metadata comes from the certificate information: the number of SAN, validity period of certificates, and the presence of self-signed server certificate. The second feature type, Flow Metadata, consists of 7 features. The first feature on source port is represented with a boolean flag that is '1' if a source port number falls within the designated range of ephemeral ports [12] and '0' otherwise. The second feature on destination port is also defined as a boolean flag, representing if it matches one of the IANA ten port numbers related to TLS [13]. The third to seventh features corresponds to the packet flow statistics. The third feature type, SPLT, was explained in Sect. 2. A feature of SPLT/SPT is represented with a 100-dimensional integer vector that is reshaped from a 10×10 sensitive matrix.

In order to make the malicious encrypted traffic more time-resilient, we propose additional features that can be retrieved from the following sources: Virus-Total [14], TLS certificates, and certificate costs. Table 1(b) shows the 15 additional features that are not easy for attackers to change themselves, considering

Table 1. Time-resilient features: (a) features adopted from [1] and (b) proposed features.

(a)		
Feature Type	Feature Descriptions	Data Type
TLS Metadata	Selected cipher suite	16-bit binary vector
	#Subject Alternative Name (SAN)	Integer
	Certificate validity (in days)	Integer
	Is self-signed certificate	Boolean
Flow Metadata	Is designated range of source port	Boolean
	Is designated range of destination port	Boolean
	#Inbound bytes	Integer
	#Outbound bytes	Integer
	#Inbound packets	Integer
	#Outbound packets	Integer
	Duration of packet flow	Integer
	SPLT	Sequence of Packet Lengths (SPL) Sequence of Packet Times (SPT)
(b)	SPLT	100-dim Integer vector 100-dim Integer vector
	Data Source	Feature Descriptions
	VirusTotal	#TSL Certificates registered with IP. Average #SANs for last 10 certificates. Average validity period for last 10 certificates. #Malicious files detected at IP #Detection counts as malicious at IP #Malicious files detected with specified IP #Detection counts as malicious at specified IP
	TLS Certificates	Is JA3 blacklist. Is TLS version 1.2 or 1.3. Average validity period of server and intermediate certificates. Valiance validity period of server and intermediate certificates. #Certificate hierarchies.
	Certificate Costs	Is EV or OV certificate. Is cheap certificate. Is wildcards used in Common Name (CN) or SAN.

their malicious intentions. VirusTotal is a service that performs malware inspections on files and other elements. Users can submit their queries on URL, IP and/or files for analysis and obtain the analytical results on their maliciousness. Since VirusTotal store the users' queries and responses in the database, the users can easily retrieve the historical records on compromised events. In this study, we adopt the following information retrieved from VirusTotal as additional features shown in Table 1(b).

Furthermore, we adopt the JA3 blacklist published in [15]. JA3 is a hashed value obtained by referring to Client Hello packet of SSL handshake and by extracting the values of the following five fields: SSLVersion, Cipher, TSLExtension, EllipticCurve, and EllipticCurvePointFormat. Utilizing the JA3 fingerprint,

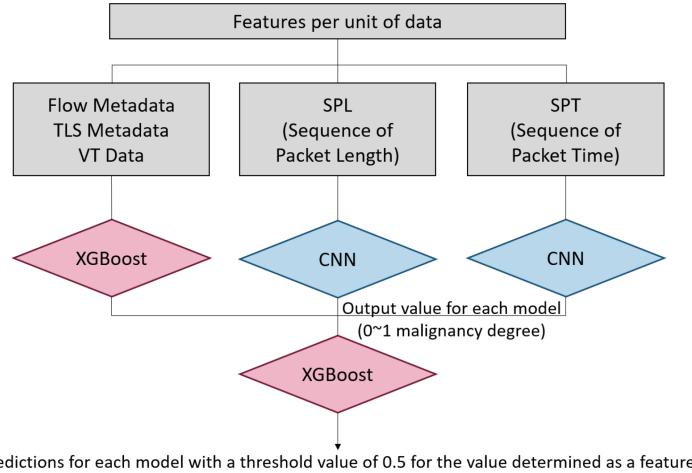


Fig. 1. Proposed XGBoost/CNN hybrid model for encrypted malicious traffic.

it is possible to identify specific traffic, and we can easily check if the traffic comes from a device published in the blacklist. However, as described in [15], relying solely on these fields, determined by the supported fields on the client side, may lead to high potential for false detection. Therefore, the JA3 fingerprint are used as features for the malicious traffic detection.

Considering that the costs of TSL certificates can affect the attack motivation, we can see if a TSL certificate is issued by major authorities or the 17 known authorities that offer free or low-cost certificates [16, 17]. We also take into consideration factors such as the scrutiny level of certificates [18] by seeing if a wildcard option is used in Common Name (CN) or SAN.

3.2 Hybrid Machine Learning Model for Malicious Traffic Detection

Using the 28 features shown in Table 1, we can construct a machine learning model for high-performance malicious traffic detection. However, to ensure more time-resilient detection, we propose an hybrid machine learning model in Fig. 1 where the prediction by XGBoost with the 28 features are combined with the outputs of the two Convolutional Neural Networks (CNN). In Fig. 1, in the left column, XGBoost learns the combined features, including flow metadata, TLS metadata, and proposed features. In the middle and right columns, CNN learns two of 10×10 matrices of SPT and SPL features. In the initial stage, the output is trained to compute a continuous malicious score ranging from 0 to 1 for each traffic. The malicious scores obtained from each model are used as three variables and input into the second-stage XGBoost model. The final determination of maliciousness is made by classifying the output as either 0 (benign) or 1 (malicious) using a threshold of 0.5.

The reason for dividing the features into three parts and training them separately is to examine the importance of the features. SPL and SPT features do not hold much significance in terms of the importance of individual elements since they represent a mere range of numerical values. Instead, we believe that the overall variability and patterns when considering the entire matrix are more crucial. On the contrary, flow metadata and similar features have meaningful interpretations based on their magnitude. To confirm the robustness, which is one of the objectives of this study, it is necessary to observe the importance and changes in these features. Therefore, flow metadata is trained with XGBoost, while SPL and SPT, being conceptually different, are trained with CNN. Additionally, SPLT features are input to CNN in their matrix representation rather than treating each entry as an individual feature. This is because, as mentioned earlier, each entry corresponds to a probability value within a certain range, and when the value slightly increases, it belongs to the adjacent entry. In other words, we assumed that neighboring entries in the matrix share similar meanings, and there is some form of relationship or concentration of distribution around a certain entry. Hence, instead of treating each entry as a separate feature, we employ convolution on the matrix.

4 Performance Evaluation

4.1 Dataset

Here, we explain the dataset used in this study. To minimize environmental dependencies, we collected data from several sources. Additionally, to verify the effectiveness of the features and the robustness of the detection accuracy, we collected new data for the malicious dataset, including data up until 2022, in addition to the data used for training. To observe the changes in the features of malicious traffic at different periods, the additional data was used separately for each year. Below, we provide a brief description of the sources for both benign and malicious datasets.

– Benign Dataset

- The data used in this study is extracted from the normal traffic log database of enterprise networks. A total of 196,376 log data entries obtained from October to December 2021 were prepared as training data. Only logs related to traffic using SSL/TLS were used in this study.
- CTU-13 Normal Dataset [19]: This dataset, acquired by the Czech Technical University, contains capture data of 13 different types of malware executed in an actual network environment. The publicly available capture data includes malware, normal, and background traffic. For comparison purposes, this dataset was used as benign traffic, considering the features related to TLS Extensions that were previously used in related work to observe malware feature transitions. Regarding the normal data, it was captured using Windows running on a virtual machine, and the data used in this study is from 2017.

- Malicious Dataset
 - Lastline: A collection of very short captures of malware detected by network sensors from 2016 to 2019, publicly available at [20].
 - Stratosphere: Similar to the Lastline dataset, this dataset contains long-term captures of malware using TLS for communication from 2016 to 2018, publicly available at [20].
 - Malware-Traffic-Analysis.net (MTA.net) [21]: This website provides pcap files and malware samples related to malicious network traffic. For this study, unused malicious pcap files from 2018 to 2022 were obtained from this site and used for evaluating the detection accuracy of unknown malicious traffic and investigating feature transitions.

To ensure that malicious pcap files do not contain benign traffic, filtering was performed by removing data matching the three major top 1 million sites, namely Alexa [22], Majestic [23], and Cisco Umbrella [24]. The data collected from these sources was processed using Zeek [25], to create the dataset used in this study. Table 2 summarizes the number of data entries after filtering.

Table 2. Evaluation Datasets.

Name	Label	#Pcap Files	#Data
Corporation A	benign	-	193,823
CTU-13	benign	8	25,013
Lastline	malicious	183	418
Stratosphere	malicious	30	32,028
MTA.net 2018	malicious	11	6,129
MTA.net 2019	malicious	20	982
MTA.net 2020	malicious	29	1,944
MTA.net 2021	malicious	33	3,669
MTA.net 2022	malicious	14	532

4.2 Consideration of Detection Rates and Feature Transitions for Unknown Malicious Traffic

In this subsection, let us study the distributions of a specific feature through the comparison between benign and malicious traffic. Due to evasive attempts by attackers and for some reasons, some features may become indistinguishable between benign and malicious traffic, and this would cause the performance drop in detection by a machine learning model. Thus, the purpose of this subsection is to understand the trends associated with each feature, aiming to construct more robust detection models.

The malicious dataset consists of the observations from 2016 to 2018. In this section, we verified how well the trained XGBoost model could detect unknown

malicious traffic from different years. We obtained malicious pcap files from “Malware-Traffic-Analysis.net” spanning from 2018 to 2021 and extracted features using Zeek Network Security Monitor.

The malicious traffic used for training the XGBoost model in this study was collected from 2016 to 2018, and as a result, there were more missed detection for the malicious traffic from years not used in training, especially for the 2021 data. We examined the types of malware that were missed in the 2021 data but did not observe any bias in terms of specific malware types being detected or missed. This suggests that overall, it has become more challenging to distinguish between normal traffic and malicious traffic. Figures 2 and 3 show the comparison of various features between benign traffic (normal), the malicious traffic used during training (2016–18), and the features from 2018 and 2021. Furthermore, regarding other features related to TLS configurations, we examined the changes in malicious data from 2018 to 2021 using training data from 2016 to 2018. For each feature, we set a threshold and visualized the percentage of data exceeding the threshold in Fig. 4.

Figure 2 illustrates an example of a significant change in a feature. As observed in Fig. 2, the proportion of self-signed certificates in malicious traffic has decreased significantly from 2018 to 2021. Additionally, Fig. 2c reveals an increase in the usage of low-cost certificates in the 2021 malicious traffic. Many adversaries have adapted their tactics to blend in with normal traffic by evading obvious features and employing changes or disguises. Regarding the TLS version, which was included as a feature in our study, Fig. 2b shows that 79% of the 2021 malicious traffic used TLS 1.2 or 1.3. Since the IETF deprecated TLS 1.0 and 1.1 in 2021 for security reasons [26], it is expected that attackers will gradually eliminate the difference between normal and malicious traffic on this front as well.

Conversely, for the features that did not show significant differences, we provide some examples in Fig. 3. Regarding JA3, which is a hash value of client-side extensions, it can also be present in benign traffic, but we did not observe significant changes in malicious traffic. Additionally, other features such as those shown in Fig. 3c and Fig. 3d rely on historical information that has been accumulated in benign traffic, making it difficult for attackers to intentionally modify them.

Although not included as features in this study, previous work utilized the number of TLS extensions and the position of the server’s chosen cipher suite in the client’s list. To compare the transitions, we used the CTU-13 dataset as benign traffic and performed a similar analysis. We set thresholds for each metric and compared the percentage of data points with a value of 1 or above to those with a value below 1, graphically represented in Fig. 4. To provide a simultaneous comparison, we also included data obtained from the CTU-13 dataset for benign traffic. From Fig. 4, it is evident that for the indicated metrics, the proportion of malicious traffic exhibiting settings similar to normal traffic increases as the data becomes more recent. TLS libraries like OpenSSL allow customization of cipher suites and the use of pre-defined options, enabling attackers to evade

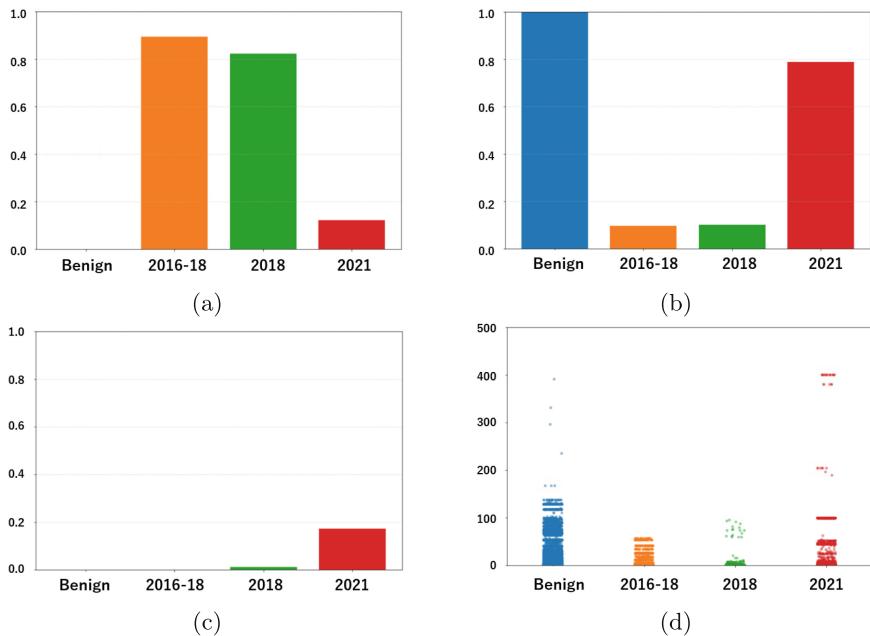


Fig. 2. Feature difference between benign and malicious traffic and distinctive time-change in malicious features on (a) rate of self-signed certificates, (b) rate of TLS 1.2 or 1.3, (c) rate of low-cost certificates, and (d) the number of SANs.

detection by avoiding rarely used encryption methods that might be associated with these features. Attackers may intentionally modify their choices by selecting stronger cryptographic algorithms commonly used in normal traffic, avoiding less frequently used cryptographic algorithms, and adjusting settings related to extensions and elliptic curves, among others. We anticipate that an increasing number of malware strains are adopting these tactics to evade detection. For this section's validation, we utilized pcap files obtained from MALWARE-TRAFFIC-ANALYSIS.NET. Table 2 presents the breakdown of malware types for the malicious traffic used in training and for each year. Different types of malware are expected to exhibit varying degrees of obfuscation or modification to evade detection, so it is important to consider the bias in the data and the tendencies of the features in our analysis.

4.3 Robustness of the Proposed Model to Trend Changes in Malicious Traffic

Finally, we compare the detection performance and robustness of the proposed model structure. As described in Sect. 1, we made adjustments to the usage of

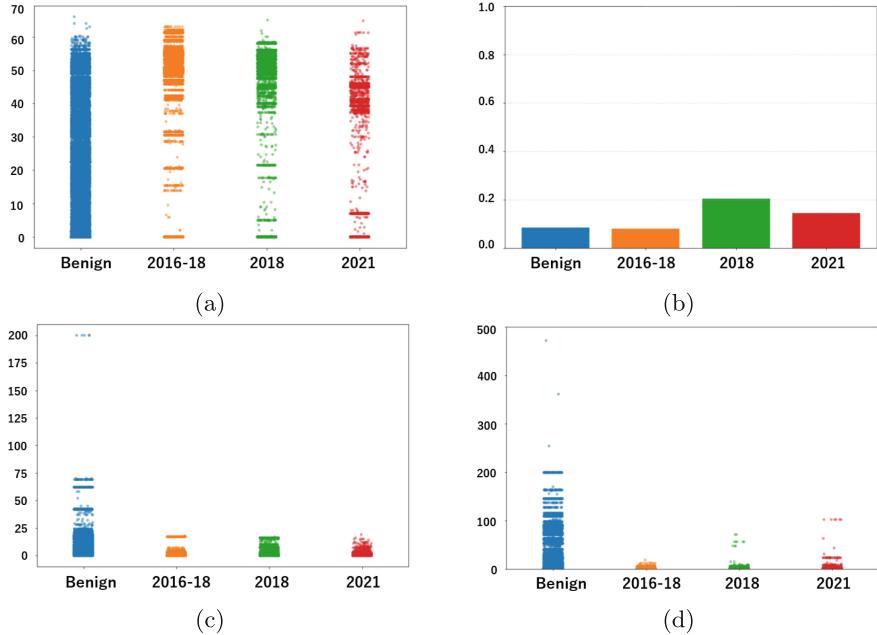


Fig. 3. Feature difference between benign and malicious traffic and time-resilient properties in malicious features on (a) #malignant files detected by VirusTotal, (b) matching rate with JA3 blacklist, (c) the number of TSL certificates registered with IP, and (d) the average number of SANs for last 10 certificates.

SPLT features and prepared the following two models for comparison, as shown in Fig. 1.

XGBoost trained with SPLT entries as features, alongside other flow metadata features, including SPT 0–99 and SPL 0–99 matrices (proposed model). XGBoost trained without SPLT features, only using flow metadata features. We evaluated the detection accuracy of these three models using validation data consisting of TLS malicious traffic from MALWARE-TRAFFIC-ANALYSIS.NET [21] spanning from 2018 to 2022. The results are shown in Fig. 5.

In the model without considering SPLT, although it exhibits relatively high performance around the time of the training data (approximately 2018), the detection accuracy significantly decreases as time passes and unknown data, i.e., data not present in the training data, increases. This makes it challenging to adapt to the emergence of malware subtypes.

Conversely, in the other two models that consider SPLT, although there is a decline in performance over time, it is relatively mitigated compared to the

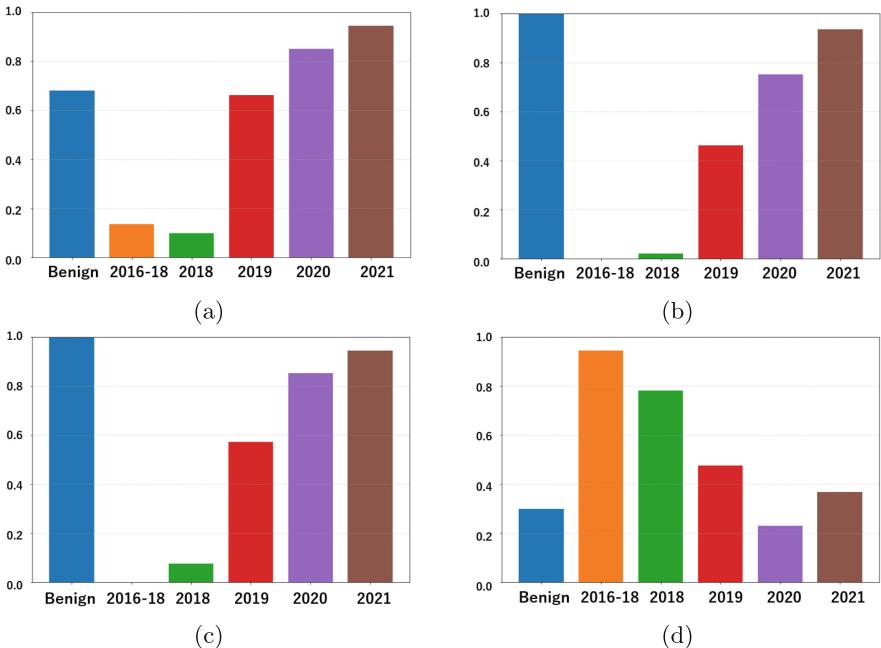


Fig. 4. Feature difference between benign and malicious traffic on (a) the average number of cipher suites used in clients, (b) the number of certificates using Elliptic Curves, (c) the number of certificates using TLS extensions, and (d) the client-side ranking of selected cipher suite. Note that these features are not used in the detection model because it is clear that the difference between benign and malicious traffic is diminishing over time.

model that does not consider SPLT. Furthermore, the proposed model demonstrates some advantages in terms of robustness. While there is some variation in the ranking across years, this is likely due to the overlapping data composition of malware types between the training and validation datasets and subtle differences in the tendencies of SPLT. It is necessary to conduct additional investigations to explore the biases in malware types and the specific features of SPL and SPT to gain a better understanding.

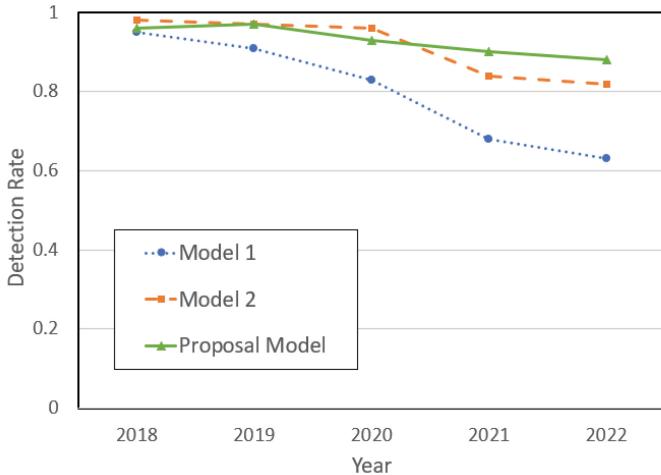


Fig. 5. Effectiveness of SPLT features in the proposed XGBoost/CNN hybrid model (see Fig. 3.2). ‘Model 1’ is a detection model using XGBoost whose input features are only Flow/TLS metadata and VT data, while ‘Model 2’ considers SPLT features as well as Flow/TLS metadata and VT data.

5 Conclusions and Future Work

In this paper, we presented a new approach to actualizing the robust detection for malicious encrypted traffic where time-resilient features and hybrid structure of machine learning models are introduced so that attackers cannot evade their malicious activities from various aspects. In addition to the flow metadata and TLS metadata previously used in [1], new features focusing on versions and costs of TLS server certificates, certificate paths, and past records on specific TSL certificates are taken into account. Machine learning techniques were employed to detect malicious traffic based on these features. Furthermore, we evaluated the detection rates using the trained models obtained in this study against malicious traffic data from 2018 to 2021. We also discussed the changing trends and robust features in the characteristics of malicious traffic.

The test data results showed that while the precision was high, the recall, which is an evaluation metric for missed detections, was slightly lower. It is necessary to build a detection system that combines detection models focusing on features like metadata and traffic behavior, with the primary goal of reducing missed detections. In the experiments evaluating the detection rate for time-series malicious datasets, a decrease in detection accuracy was particularly observed for the data obtained from 2021. Based on the results of the temporal analysis of each feature and the importance of the features using SHAP values, it is believed that the presence of more benign-like trend data in elements with higher importance is the cause. As attackers are expected to continue adapting to blend in with normal traffic, further investigation from the attacker’s perspective is necessary.

Furthermore, the results of this study suggest that the decrease in detection rate is not gradually worsening over time. Other factors such as biases in the acquired malicious traffic and the types of malware may also contribute. While this study focused on the binary classification of benign and malicious traffic, there is a need for multi-class classification for different malware families. By preparing datasets obtained from the same environment for benign and malicious traffic and extracting data specific to each malware type, it is expected to gain insights into the trends of each family and develop more detailed observations and robust detection systems.

Moreover, we created and verified effective features in this study, but counter-measures are likely being taken against obvious distinguishing features. Therefore, as suggested in related work, future detection systems will require feature extraction systems that actively and automatically extract features for distinguishing between benign and malicious traffic. We aim to build machine learning systems that can detect changes and are difficult for attackers to modify or mimic by interpreting changes in trends from large amounts of data and during the process of attackers constructing and executing attacks.

The current detection system for TLS encrypted malware traffic has some limitations that should be solved for the actual deployment. First, the proposed detection system requires somehow rich computational resources both for training and deployment because it adopts a flow-based approach to discovering malicious intents in encrypted communications. Second, in the training phase, our system requires new malicious TLS flow data and the updated information of the third-party services like VirusTotal. Collecting reliable data and adapting to the evolution of attack methods might cause some time delay before deploying the updated system. To shorten the delay, an automatic data collection and update system can be introduced, while it still requires a regular monitoring and maintenance system to ensure the high-performance in a real deploy environment. On the other hand, the detection itself works almost real time in our experimental setting. However, we should prove the real-time detection under real environments.

Acknowledgment. The results of this research were obtained through a collaborative study with LY Corporation. We would like to express our sincere gratitude to other staff in LY Corporation for their invaluable guidance and support throughout this study. A part of this study was conducted as the project of Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 23K21670 and 21KK0178.

References

1. Anderson, B., Paul, S., McGrew, D.: Deciphering malware's use of TLS (without decryption). *J. Comput. Virol. Hack. Techn.* **14**(3), 195–211 (2018). <https://doi.org/10.1007/s11416-017-0306-6>
2. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: an overview. *IEEE Commun. Mag.* **57**(5), 76–81 (2019). <https://doi.org/10.1109/MCOM.2019.1800819>

3. Barut, O., Grohotolski, B., DiLeo, C., Luo, Li, P., Zhang, T.: Machine learning based malware detection on encrypted traffic: a comprehensive performance study. In: Proceedings of 7th International Conference on Networking, Systems and Security (2020). <https://doi.org/10.1145/3428363.3428365>
4. Gomez, G., Kotzias, P., Dell'Amico, M., Bilge, L., Caballero, D.: Unsupervised Detection and Clustering of Malicious TLS Flows. [arXiv:2109.03878](https://arxiv.org/abs/2109.03878) (2022)
5. Lee, W., Jin, S.: Encrypted malware traffic detection using TLS features and random forest. In: Atluri, S.N., Vušanović, I. (eds.) ICCES 2021. MMS, vol. 98, pp. 85–100. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67090-0_8
6. Afzal, Z., Brunstrom, A., Lindskog, S.: Using Features of Encrypted Network Traffic to Detect Malware. Secure IT Systems. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-70852-8_3
7. Liu, Z.: DeepTLS: Comprehensive and High-performance Feature Extraction for Encrypted Traffic, vol. 1 (2022). <https://doi.org/10.48550/ARXIV.2208.03862>
8. Zheng, X., Li, H.: Identification of malicious encrypted traffic through feature fusion. IEEE Access **11**, 80072–80080 (2023). <https://doi.org/10.1109/ACCESS.2023.3279120>
9. He, G., Wei, Q., Wang, J., Zhu, H., Xu, B.: One-shot detection of malicious TLS traffic. In: 2022 IEEE International Conference on Computer Supported Cooperative Work in Design (2022). <https://doi.org/10.1109/CSCWD54268.2022.9776236>
10. Zhou, Z., et al.: Malicious encrypted traffic features extraction model based on unsupervised feature adaptive learning. J. Comput. Virol. Hack. Techn. **18**, 453–463 (2022). <https://doi.org/10.1007/s11416-022-00429-y>
11. Kim, D., Han, J., Lee, J., Roh, H., Lee, W.: Poster: feasibility of malware traffic analysis through TLS-encrypted flow visualization. In: 2020 IEEE International Conference on Network Protocols (ICNP) (2020). <https://ieeexplore.ieee.org/document/9259387/>
12. Ephemeral Port (2024). https://en.wikipedia.org/wiki/Ephemeral_port. Accessed 23 Sep 2024
13. List of TCP and UDP Port Numbers (2024). https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers. Accessed 23 Sep 2024
14. Virus Total (2024). <https://www.virustotal.com/>. Accessed 23 Sep 2024
15. abuse.ch.: SSL Blacklist (2024). <https://sslbl.abuse.ch/>. Accessed 23 Sep 2024
16. Let's Encrypt (2024). <https://letsencrypt.org/docs/>. Accessed 23 Sep 2024
17. Zero SSL (2024). <https://zerossl.com/pricing/>. Accessed 23 Sep 2024
18. DV, OV, IV, and EV Certificates (2022). <https://www.ssl.com/article/dv-ov-and-ev-certificates/>. Accessed 09 May 2023
19. Stratosphere: Stratosphere Laboratory Datasets (2015). <https://www.stratosphereips.org/datasets-overview>. Accessed 23 Sep 2024
20. Roques, O.: Detecting Malware in TLS Traffic (2019). <https://api.semanticscholar.org/CorpusID:208194045>
21. MALWARE-TRAFFIC-ANALYSIS.NET (2024). <https://www.malware-traffic-analysis.net/>. Accessed 23 Sep 2024
22. Alexa: The Top 500 Site on The Web (2024). <https://gist.github.com/elimistev/69077a93d21b8bf8a02a362c830fbcb1>. Accessed 23 Sep 2024
23. Majestic Million (2024). <https://majestic.com/reports/majestic-million>. Accessed 23 Sep 2024
24. Cisco: Cisco Umbrella 1 Million (2024). <https://umbrella.cisco.com/blog/cisco-umbrella-1-million>. Accessed 23 Sep 2024

25. Zeek Project Team: Zeek (2024). <https://github.com/zeek/zeek>. Accessed 23 Sep 2024
26. Cimpanu, C.: IETF Officially Deprecates TLS 1.0 and TLS 1.1 (2024). <https://therecord.media/ietf-officially-deprecates-tls-1-0-and-tls-1-1>. Accessed 23 Sep 2024



Enhancing Semantic Segmentation in Open Compound Domain Adaptation Through Mixed Image and Epistemic Uncertainty

Yiqun Ma[✉], Wenrui Wang[✉], Siyuan Wang[✉], Xi Yang[✉],
and Yuyao Yan^(✉)[✉]

Xi'an Jiaotong-Liverpool University, Suzhou, China
{Yiqun.Ma22,Wenrui.Wang22,Siyuan.Wang21}@student.xjtlu.edu.cn,
{Xi.Yang01,Yuyao.Yan}@xjtlu.edu.cn

Abstract. Open Compound Domain Adaptation (OCDA) presents a novel challenge in semantic segmentation, where the target domain combines multiple domains with blurry boundaries and unseen categories. While UDA-based semantic segmentation achieves high accuracy on unseen domain data, it struggles to maintain accuracy in open compound domains. Specifically, data augmentation for accurate predictions is challenging, and uncertainty in prediction probabilities often goes overlooked when encountering unknown categories from new domains. In this paper, we propose an uncertainty quantification method to measure the epistemic uncertainty of the model, thereby improving the reliability of its generated predictions. We also propose a novel data augmentation approach that combines paired images from different domains, employing Global Luminous Alignment (GLA) to generate new augmented samples, thereby reducing the domain variance between the target and source domain data. Experiments on GTA5, BDD100K, Synthia, and Cityscapes datasets demonstrate the effectiveness of our methods.

Keywords: Open Compound Domain Adaptation · Epistemic Uncertainty · Data Augmentation

1 Introduction

Supervised learning has achieved great success in the field of Semantic Segmentation [20]. However, supervised learning methods struggle with domain shift issues. To address that issue, Unsupervised Domain Adaptation (UDA) [1, 35, 38] tasks exploit a large amount of unlabeled target-domain data to generate pseudo labels to transfer the knowledge from the labeled source domain to the unlabeled target domain. Albeit the usefulness, they do not work well in real-world domain adaptation problems. In particular, the target data consists of multiple

Y. Ma and W. Wang—Equal contribution.

similar domains, where the distinction between those domains is hard to define. For example, in automatic driving under cloudy and rainy weather, the target data distributions become difficult to distinguish. Therefore, predicting precise semantic segmentation maps in real-world scenarios presents novel challenges.

A more realistic domain adaptive task, named as Open Compound Domain Adaptation (OCDA) [19], reflects a new data collection situation in which the target domain is composed of multiple homogeneous domains without class labels of a few samples. However, the majority of previous attempts at OCDA tasks [12, 27] involve straightforward applications of existing UDA techniques, wherein the composite target domain is treated as an unimodal distribution [29]. This approach neglects the alignment of multiple domains within the target, thereby generating inter-domain discrepancies. This may directly lead to lower reliability of pseudo labels generated in self-training.

In efforts to mitigate the distribution variance between the source and target domains, experts employ data augmentation methods, such as image fusion and linear difference data augmentation [40, 42]. Specifically, introducing generative models such as the Wasserstein Auto Encoder and Generative Adversarial Network for the continuous simulation of the target domain distribution [30] is also a widely adopted method to generate augmented data. However, prevalent data augmentation strategies operate under the assumption of comprehensive knowledge of all source domains, neglecting the real-world scenarios of existing unknown distributions. Consequently, when confronted with an open compound target domain, including multi-target domains and unseen domains that are significantly divergent from the source domain, performance gains may be severely impeded [6, 17].

Based on the above issues, a more realistic continuous domain adaptive setting, OCDA [19], has been explored, which allows the model to achieve higher performance on test data within a nuanced and compounded target domain. The SF-OCDA model [46], promotes a cross-patch style exchange method to enrich samples with diverse patch styles at the feature level. Furthermore, a multi-teacher framework is proposed for domain adaptation tasks, integrating bidirectional photometric mixing to enhance model adaptation to varied target subdomains [26]. While these strategies do improve the adaptation of the model, they do not adequately address the uncertainty of the prediction probabilities of the model in the face of unknown classes in the new domain. As a consequence, high uncertainty prediction results are erroneously categorized as “correct” outputs, potentially leading to inaccurate predictions in ambiguous areas.

In this paper, we explore the epistemic uncertainty of open compound domain adaptive semantic segmentation, which quantifies the uncertainty and enhances the reliability of generated pseudo labels. Meanwhile, we propose a novel data augmentation method under the OCDA setting [19], Global Luminous Alignment, termed GLA, to mix the source domain data and target domain data. It performs a certain degree of difference elimination operation for the basic features of the images between the mixed domains. We conduct comprehensive experiments on publicly available benchmark datasets (GTA5 [31],

BDD100K [39], Synthia [32], and Cityscapes [4] datasets). Our experimental results show a significant improvement in mean Intersection over Union (mIoU), surpassing the state-of-the-art performance in the DACS task [34] by approximately 7% points and the OCDA task [19] by 3% points, demonstrating the effectiveness of our model.

In conclusion, our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to explore epistemic uncertainty in open compound domain adaptive semantic segmentation, improving the reliability of pseudo labels of the compound target domain data.
- We introduce a novel data augmentation approach, Global luminous alignment (GLA) to open compound domain adaptive semantic segmentation, generating augmented samples through luminosity mixing of the source domain images and the target domain images to enhance model performance on test data within a nuanced and compounded target domain.
- We conduct comprehensive qualitative and quantitative experiments on public benchmark datasets to compare our methodology with state-of-the-art methods in the OCDA setting [19]. The results demonstrate that our model outperforms existing methods.

The rest of the paper is organized into the following sections. Section 2 summarizes the important related work to contextualise our approach. Section 3 presents the GLA and uncertainty module used in open compound domain adaptive semantic segmentation. Section 4 outlines the experimental setup and experimental results. Section 5 concludes our work and future work direction.

2 Related Work

2.1 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) aims to transfer the learned knowledge from the labelled source domain to the target domain. The UDA methods can be classified into adversarial training methods [11, 21, 28, 33] and self-training methods [8, 9, 24, 44, 45]. The purpose of adversarial training methods is to align the distributions of source and target domains in a GAN framework, where the use of multiple scales or categories of information for the discriminator can optimize the comparison. In self-training methods, the network is trained using pseudo labels of the target domain. In other cases, pseudo labels can also be computed online during training, and to mitigate training instability.

2.2 Open Compound Domain Adaptation

Open Compound Domain Adaptation (OCDA) [19] is a domain adaptation setting in which the target domain is modelled as a compound of multiple unknown homogeneous domains, which facilitates model generalization to unknown domains (unseen domains). OCDA [19] can handle unlabeled compound heterogeneous target domains and unseen open domains. It proposes a

course learning strategy but does not fully use the specific information of each target subdomain. Based on this, OCDA researchers [12, 27] proposed to separate the composite target domain into multiple subdomains to better deal with intra-domain gaps. MOCDA [12] has devised a novel meta-learning-based approach for continuous modelling of separated unlabeled target sub-domains, which improves the generalization of the model DHA [27] utilizes image translation and the target-to-source alignment domain-wise to exploit domain invariant features from multiple subdomains. ML-BPM [26] introduces a multi-teacher framework to adapt to each target subdomain separately.

2.3 Data Augmentation

The performance of deep learning models is limited by the quality of training data. Applying data augmentation techniques can increase the diversity of training data by performing various transformations on the training data, thus improving the generalization ability and robustness of the model. This technique has been widely used in tasks such as image classification and image segmentation. In image classification tasks, random flips, rotations and crops are commonly used [13]. More sophisticated techniques such as Cutout [7] (generating random occlusions), CutMix [40] (replacing part of an image with another) and MixUp [41] (linearly interpolating between two images) have shown very impressive results. Data augmentation can simulate the appearance of targets at different locations and angles, increasing the diversity of training data and improving the robustness of the detection model. In image segmentation tasks, data augmentation can change the brightness and contrast of an image by transforming the scale and shape of the image [25], which helps to improve the generalization ability of the model. Alternatively, adversarial generative models [16, 43] have also been widely used in data augmentation. However, this adversarial approach produces output with detailed feature distributions, thereby reducing the discrepancy in feature distributions at the image level within the same domain. To address this limitation, we propose a novel data augmentation strategy, the Global Local Alignment module, to project pixel-level features from both target and source domains into the source domain feature manifold. Our data augmentation method enables the effective reduction of the domain gap between the target and source domains while preserving feature diversity.

2.4 Epistemic Uncertainty

Uncertainty estimation is used to assign a level of confidence to the output of a model [14]. Epistemic uncertainty is also known as model uncertainty. It states that the model's own estimates of the input data may be inaccurate due to poor training and insufficient training data, independent of an individual piece of data [17]. Thus, epistemic uncertainty measures the uncertainty of the model parameters estimated by the training process itself. This uncertainty can be mitigated or even resolved by targeted adjustments (e.g., by adding training data). Recently, the deterministic uncertainty approach [10, 17, 25, 41] has led the way

in quantifying prediction uncertainty. It is specifically designed to quantify cognitive uncertainty from the distribution of potential representations in a computationally efficient way. This state-of-the-art approach shows superiority in several computer vision tasks.

3 Methodology

Current methodologies in OCDA typically encompass three main components: data preprocessing, feature alignment, and pseudo-labelling. This work primarily focuses on enhancements in data preprocessing and pseudo-labeling while completely following the feature alignment method outlined in OCDA [19]. In the data preprocessing phase, we introduce rare class sampling to improve the learning of rare classes and employ Global Luminous Alignment with distinct processing methods for the luminance and colour channels. This approach ensures that source and target images have similar luminosity properties, ultimately enhancing the quality of mixed images for better model generalization and performance. In the pseudo-labelling phase, we introduce uncertainty estimation into OCDA tasks, aiming to quantify epistemic uncertainty to improve the reliability of pseudo labels. Our framework is illustrated in Fig. 1, and the specifics of each component will be detailed in the following sections.

3.1 Preliminary

Segformer is a common framework for semantic segmentation [36]. We utilize this framework as the source domain network g_θ . The source domain data is represented by x_S and its corresponding labels are y_S . Same with Segformer, our source model is trained via a categorical cross-entropy loss L_S :

$$L_S = - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_S^{(i,c)} \log g_\theta(x_S)^{(i,c)}, \quad (1)$$

where $H \times W$ represents the dimensions of the source image in terms of height and width, and C denotes the number of categories.

Additionally, we utilize a self-training approach to address the domain gap by leveraging a source-trained teacher network g_θ . This network generates pseudo labels for the target domain data x_T based on the class with the highest predicted probability. Here, c' represents each possible category in the target domain. The pseudo label \hat{y}_T is determined as follows:

$$\hat{y}_T^{(i,c)} = \begin{cases} 1, & \text{if } c = \arg \max_{c'} g_\theta(x_T)^{(i,c')} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

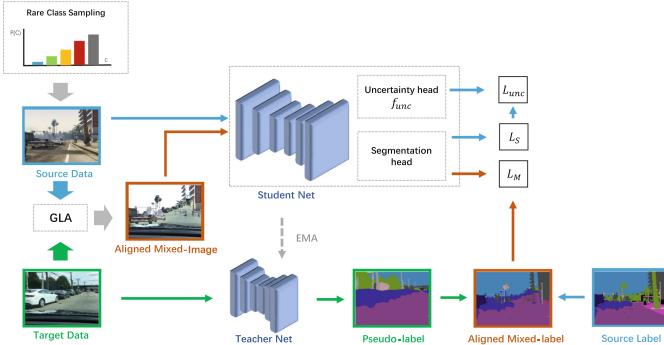


Fig. 1. Overview of proposed framework. GLA is used to generate mixed images and utilize a source-trained model to produce target pseudo labels. Mixed labels are generated by combining the source domain labels with target pseudo labels. During training, self-training model leverages both source and mixed domain data. Additionally, epistemic uncertainty is estimated on predicted results to correct pseudo labels through iterative refinement of the student model. Furthermore, exponential moving average method is employed to update the teacher model.

Exponentially Moving Average. In this work, we update the teacher net g_θ with an Exponentially Moving Average method [18], shown in Eq. 3. The source domain teacher network weights $\mu_{g\theta}$ are updated using the student network weights μ_g after each training step t .

$$\mu_{g\theta}^{t+1} \leftarrow \alpha \mu_{g\theta}^t + (1 - \alpha) \mu_g , \quad (3)$$

where α is the momentum to temporally ensemble the student network.

Rare Class Sampling. In order to improve the learning of rare classes, we use the Rare Class Sampling (RCS) [15]. RCS samples images with rare classes from the source domain and calculates the frequency of the class by the number of pixels class of the rare class:

$$f_c = \frac{\sum_{i=1}^{N_s} \sum_{j=1}^{H \times W} [y_s^{(i,j,c)}]}{N_s \times H \times W} , \quad (4)$$

where c is the class order number, N_s is the number of images in the source domain. Then the sampling probability of each class can be obtained:

$$P(c) = \frac{e^{1-f_c}/T}{\sum_{c'=1}^C e^{1-f_{c'}/T}} , \quad (5)$$

where T is the temperature parameter that balances the distribution. Higher T values yield a more balanced distribution, while lower values enhance the sampling probability of rare classes.

Notations. To further stabilize the training process, we follow DACS [34] to generate pseudo labels on non-augmented images and train the student network with domain-mixed images. In each iteration, a subset paired source and target images are sampled, we denote x_S^n and y_S^n as a pair of source images and their corresponding labels, and x_T^m and y_T^m as target images and their pseudo labels, where the n and m represent the index of the selected source and target image, respectively. Subsequently, a random subset of classes is selected from the source labels y_S , which is then used to generate a binary mask $M \in \{0, 1\}^{H \times W}$, where pixels corresponding to the selected classes are assigned a value of 1, and all others are set to 0. The mixed images with their mixed labels are defined as:

$$x_{mix} = x_S^n \odot M + x_T^m \odot (1 - M), \quad (6)$$

$$y_{mix} = y_S^n \odot M + \hat{y}_T^m \odot (1 - M), \quad (7)$$

where \odot denotes the Hadamard product. The student model is then trained with a mixed image and its mixed label with the mixed cross-entropy loss:

$$L_M = - \sum_{i=1}^{H \times W} \sum_{c=1}^C w_{mix}^{(i)} y_{mix}^{(i,c)} \log g(x_{mix})^{(i,c)}. \quad (8)$$

Here, $w_{mix} = 1 \odot M + w_T \odot (1 - M)$ is the weight map to decrease the potential error of pseudo labels.

$$w_T = \text{Sigmoid}(f_{unc} \cdot \exp(-z_{sc})/\tau), \quad (9)$$

where τ is the temperature to control the sensitivity of uncertainty estimation in target domain reweighting, z_{sc} is generated based on a predefined confidence threshold upon the maximum class probability, as detailed in Eq. 14.

3.2 Image Alignment

Global Luminous Alignment (GLA) is a data alignment operation performed by the source domain dataset images and the target domain dataset images on the Lab channel, which makes the source domain data and the target domain data closer to the level of the underlying image elements, thus reducing the inter-domain gap. As shown in Fig. 2, we employed distinct processing methods for the luminance and colour channels.

Classical histogram matching is performed on colour channels (a and b) for the source domain image and the target domain image to prevent the occurrence of vignettes and colour errors in the histogram alignment results [23].

Gamma correction is used for the L channel, which is different from that of the colour channel. Gamma correction is realized by a power law function $C(I) = I^\gamma$ to constrain the luminosity magnitude of the source and target domain images, where $C(I)$ is the coded value of the image, I is the luminosity value range from 0 to 1 at each pixel of the image, and γ is the correct parameter. This power



Fig. 2. The Left image represents the source domain image, while the center image is the target domain image. The right image is a mixed image, which has undergone global luminance alignment to make the luminance properties of the source and target images more similar. zh

law function makes the average value of the L channel in the source domain image equal to the average value in the target domain data. When γ is 1, the photometric values between domains are equal. The mean value of the histogram is indicated:

$$\sum_I C(I)h_m^p(I) = \sum_I I^\gamma h_m^p(I) = \sum_I Ih_n^q(I), \quad (10)$$

where h_m^p and h_n^q present the histograms of luminosity in the source and target domains, respectively.

To adapt the alignment parameters to elemental attributes of different images, our approach employs photometric alignment to automatically estimate the gamma correction parameter γ for each selected source domain image and target domain image. The optimal γ values for both domains are then determined through gradient descent. Additionally, a regularization parameter β is incorporated to refine the alignment process, with its formula presented below:

$$\gamma = \arg \min_{\gamma} \left(\sum_I I^\gamma h_m^p(I) - \sum_I Ih_n^q(I) \right)^2 + \beta(\gamma - 1)^2 \quad (11)$$

3.3 Uncertainty Module

To enhance the reliability of prediction results, we introduce uncertainty estimation [37] into the model f_{seg} , enabling the model to quantify the uncertainty of its predicted outcomes. Additionally, We define a set of learnable prototype vectors $P = \{p_j\}_{j=1}^n$, where n is the number of prototypes. For each class, by optimizing the prototypes using the prototype dissimilarity loss L_{Dis} , shown in Eq. 12, we can identify an embedding center for the intra-class data.

$$L_{Dis} = - \sum_{j < k} \|p_j - p_k\|, \quad (12)$$

where j and k are the indexes of intra-class samples.

To enhance the quality of uncertainty prediction, we introduce the epistemic uncertainty model f_{unc} to estimate the uncertainty of the model's prediction results for each pixel point. For each sample, we compute the cosine similarity

of the feature embeds z from the feature extraction model f_e with their corresponding prototypes, as shown below:

$$DM(z, P) = [S_c(z, p_1), S_c(z, p_2), \dots S_c(z, p_n)], \quad (13)$$

where DM is the distinction maximization layer, $z = f_e(X_S)$, and $S_c(-, -)$ represents the cosine similarity. Furthermore, we compute the predicted uncertainty \hat{u}^S of predicted results on the student model as shown in the following:

$$\hat{u}^S = \text{Sigmoid}(f_{unc} \cdot \exp(-z_{sc})), \quad (14)$$

where $z_{sc} = DM(z, P)$, and \exp represents the activation function.

The uncertainty loss L_{unc} optimizes the uncertainty estimation model f_{unc} as shown in Eq. 15. This optimization enables the suppression of high uncertainty in the predicted results, thereby mitigating the adverse impact on the accuracy of the final segmentation outcomes.

$$L_{unc} = BCE(\hat{u}^S, Norm(L_S)), \quad (15)$$

where L_S represents source domain cross-entropy loss as detailed in Eq. 1, $Norm$ denote the error is min-max normalized into the range of $[0, 1]$, and the BCE represents the binary cross entropy loss.

Overall Objective. In summary, to mitigate distribution variance between source and target domains, Global Luminous Alignment is employed for data augmentation. Source-domain images are mixed with target-domain images, generating mixed images and corresponding mixed labels combining source-domain labels with pseudo labels from the target domain. Unlike traditional self-supervised training, which only uses source domain data, the proposed approach trains the semantic segmentation model on mixed domain data with mixed images and pseudo labels. To improve the reliability of pseudo labels, an uncertainty loss term, L_{unc} , is integrated. The prototype dissimilarity loss L_{Dis} is used to learn prototypes. The segmentation models are updated using the EMA method, based on the total loss L_{all} , which also updates the teacher model using Eq. 3:

$$L_{all} = L_S + L_M + \lambda(L_{Dis} + L_{unc}) \quad (16)$$

where L_S and L_M represent the training losses on the source and mixed domain data, respectively, and λ is the hyperparameter balancing segmentation and uncertainty estimation.

4 Experiments

4.1 Datasets and Implementation Details

Dataset. Our evaluation method employs a widely used scenario that transfers data from a virtual source domain to a real open compound domain. For the

source domain, we use the Grand Theft Auto 5 (GTA5) dataset [31] (with 24,966 images at 1914×1052 resolution). In line with the OCDA study [19], we utilize the C-Driving dataset [19] as the target domain, derived from the BDD100K dataset [39], comprising 22,840 training images and 1,430 test images with a resolution of 1280×720 pixels. In the training process, both datasets are resized to maintain uniformity, C-Driving images [19] adjusted to 1024×512 pixels, and GTA5 images [31] to 1280×720 pixels.

Implementation Details. To be consistent with other competitors, we use the Deeplabv2 [2] framework with a ResNet101 [13] backbone as our image encoder. In the training process, the training data is divided into 4×4 patch size. We initialize our backbone with the pre-trained model on ImageNet-1k [5] datasets. Additionally, we use AdamW [22] optimizer and set the weight decay into 0.01. The initialized learning rate is set to 6×10^{-5} for the encoder and 6×10^{-4} for the decoder. In order to improve the stability of the training process, we use the linear warm-up decay strategy for the first 1500 iterations. We train the model for a total of $40k$ iterations with a batch size of 2. This batch size and number of iterations were chosen to balance the trade-off between computational efficiency and model performance. Following DACS [34], we set the teacher momentum α to 0.99. In the experiment, we used one Quadro RTX 8000 GPU running for 14.5 h to train our model.

Table 1. Comparison of GTA5 \rightarrow C-Driving adaptation in terms of mIoU(%). The best result is highlighted in **bold**.

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	mIoU
No Adaptation	73.4	12.5	62.8	6.0	15.8	19.4	10.9	21.1	54.6	13.9	76.7	34.5	12.4	68.1	31.0	12.8	0.0	10.1	1.9	28.3
DACS [34]	79.1	9.4	67.2	12.3	15.0	20.1	14.8	23.8	65.0	22.9	82.6	40.4	7.2	73.0	27.1	18.3	0.00	16.1	1.5	31.4
DHA [27]	79.9	14.5	71.4	13.1	32.0	27.1	20.7	35.3	70.5	27.5	86.4	47.3	23.3	77.6	44.0	18.0	0.1	13.7	2.5	37.1
ML-BPM [26]	85.3	26.2	72.8	10.6	33.1	26.9	24.6	39.4	70.8	32.5	87.9	47.6	29.2	84.8	46.0	22.8	0.2	16.7	5.8	40.2
Ours	87.5	44.3	76.1	35.6	38.3	27.7	20.9	39.2	75.8	37.4	90.0	50.0	26.7	82.6	68.7	67.3	0.0	22.5	3.0	47.0

4.2 Comparisons with State-of-the-Art Methods

In the experiment, we evaluate our methods on the key metric of mean Intersection over Union (mIoU) as shown in Table 1. Additionally, we conduct a comparison against four baselines: 1) **No Adaptation** means we do not use any domain adaptation methods. 2) **DHA** [27] utilizes adversarial learning for domain alignment. 3) **DACS** [34] investigates a novel cross-domain mixing technique for data augmentation. 4) **ML-BPM** [26] used a self-training framework.

Our proposed method achieves 47.0% mIoU of 19 categories, showing notable improvements over competitors. In particular, our results improve on the mIoU metric by about 7% points compared to the ML-BPM model [26], which demonstrated that our proposed luminosity alignment method could learn a better representation of some detailed features between similar source and target domains.

For some classes with fewer samples, such as TL, TS, Rider and Car, our method also maintains a comparable result compared with previous studies. Additionally, in Table 2, we compare two backbones (Segformer [36] and DeeplabV2 [3]) on the DACS [34] approach, respectively. The results show that the combination of our method with Segformer [36] quantifies the uncertainty in the training process, which can reduce the accumulated error. Therefore, our method also selects Segformer as the backbone for open compound domain semantic segmentation. Overall, all results demonstrate the effectiveness of our method under the setting of OCDA.

Table 2. Comparison of different network architectures: the comparison presents the evaluation performance on the compound domains (C), open domain (O) and open compound domain (C+O) of the C-Driving dataset.

Method	Compound(C)	Open(O)	C+O
DACS(DeeplabV2) [34]	36.6	39.7	38.2
ML-BPM(DeeplabV2) [26]	40.2	40.8	40.5
DACS(Segformer) [34]	42.1	46.3	44.2
Ours	44.8	49.2	46.2

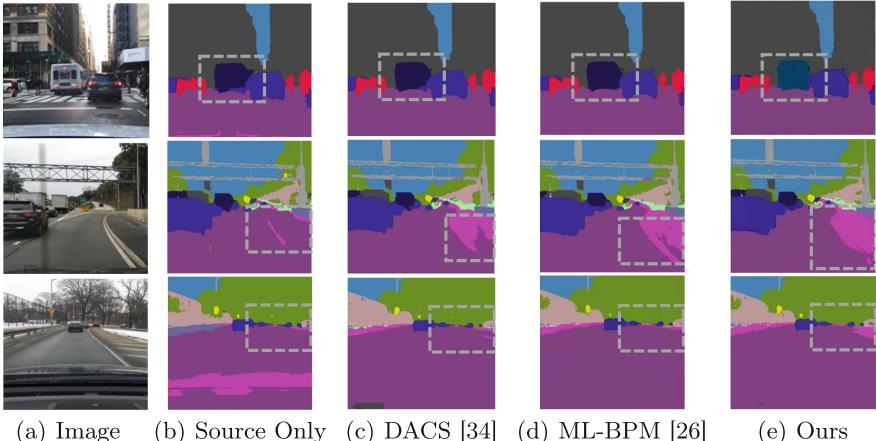


Fig. 3. Qualitative results of our method compared with baseline methods on the domain adaptation setting of GTA5 → C-Driving.

The qualitative segmentation results of our methods are given in Fig. 3. As shown in the white dashed box, our methods demonstrate improved performance in highly uncertain regions such as blurred backgrounds, unclear object boundaries, or distant objects. The qualitative segmentation results demonstrated that

our proposed uncertainty estimation method can improve the accuracy of the uncertain regions on the open compound domain.

4.3 Ablation Study

To assess the effectiveness of our model’s components, we conducted ablation studies and evaluated performance across various target domain scenarios. Specifically, we focused on the performance of the COMPOUND and OPEN domains in the experiments, which represent test scenarios with compound (C), open (O) and open compound (C+O) domains, respectively. The detailed experimental results are shown in Table 3.

Table 3. Ablation study of the efficiency of each component within our methods. GLA represents the Global Luminous Alignment, and Unc represents the uncertainty module.

Method	GLA	Unc	GTA5→C	GTA5→O	GTA5→C+O
Baseline	—	—	42.1	46.3	44.2
+GLA	✓	—	43.4	46.5	45.0
+Unc	—	✓	43.6	46.7	45.2
Ours	✓	✓	44.8	49.2	47.0

We first removed the GLA module individually and subsequently removed the uncertainty (Unc) module independently to explore the specific contribution of each module to the overall performance of the model. From the results presented in Table 3, it is clear that the Baseline model exhibits a certain level of base performance without the integration of our proposed modules. However, with the introduction of the GLA module, we observed a significant performance improvement in all the tested domains, a result that suggests that the GLA module is effective in enhancing the model’s awareness of the global and local consistency of the data structure, thus improving the generalization capability. Similarly, the inclusion of the Unc module also brings performance gains to the model, which confirms the importance of the Unc module in the identification of uncertain regions, strengthening our model to cope with various data distributions. Particularly, the combination of GLA and Unc modules in our model yields a synergistic effect, as the experimental results reveal that these two modules are functionally complementary and can mutually reinforce each other, which optimizes the adaptability of our model performance.

5 Conclusion

In this paper, we explore the uncertainty of open compound domain adaptation to correct ambiguous regions with high uncertainty in the target domain, which

can significantly improve the accuracy of generated pseudo labels. At the same time, we introduce novel data argumentation methods, which utilize the GLA image enhancement technique to mix the source and target images. Our approach achieves superior performance in quantitative and qualitative experiments, demonstrating its significance and effectiveness in open compound domain adaptation tasks. In the future, we will focus on exploring the potential of feature alignment in other image features, as well as studying how to preserve the diversity of pseudo labels when introducing the uncertainty module.

Acknowledgement. The work was partially supported by the following: National Natural Science Foundation of China under No. 92370119 and No. 62206225; Research Development Fund in XJTLU under no. RDF-23-02-044.

References

1. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3722–3731 (2017)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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)
3. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* (2018)
4. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
5. 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)
6. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? Does it matter? *Struct. Saf.* **31**(2), 105–112 (2009)
7. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint [arXiv:1708.04552](https://arxiv.org/abs/1708.04552) (2017)
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015)
9. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **27** (2014)
10. Franchi, G., Yu, X., Bursuc, A., Aldea, E., Dubuisson, S., Filliat, D.: Latent discriminant deterministic uncertainty. In: European Conference on Computer Vision, pp. 243–260. Springer (2022)
11. Ganin, Y., et al.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(59), 1–35 (2016)

12. Gong, R., et al.: Cluster, split, fuse, and update: meta-learning for open compound domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp. 8344–8354 (2021)
13. 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)
14. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint [arXiv:1610.02136](https://arxiv.org/abs/1610.02136) (2016)
15. Hoyer, L., Dai, D., Gool, L.V.: DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. CoRR abs/2111.14887 (2021). <https://arxiv.org/abs/2111.14887>
16. Ieracitano, C., Mammone, N., Paviglianiti, A., Morabito, F.C.: A conditional generative adversarial network and transfer learning-oriented anomaly classification system for electrospun nanofibers. Int. J. Neural Syst. **32**(12), 2250054 (2022)
17. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. **30** (2017)
18. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations, ICLR 2017 (2017)
19. Liu, Z., et al.: Open compound domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12406–12415 (2020)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
21. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. Adv. Neural Inf. Process. Syst. **31** (2018)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
23. Ma, H., Lin, X., Yu, Y.: I2F: a unified image-to-feature approach for domain adaptive semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
24. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision, pp. 69–84. Springer (2016)
25. Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: ClassMix: segmentation-based data augmentation for semi-supervised learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1369–1378 (2021)
26. Pan, F., Hur, S., Lee, S., Kim, J., Kweon, I.S.: ML-BPM: multi-teacher learning with bidirectional photometric mixing for open compound domain adaptation in semantic segmentation. In: European Conference on Computer Vision, pp. 236–251. Springer (2022)
27. Park, K., Woo, S., Shin, I., Kweon, I.S.: Discover, hallucinate, and adapt: open compound domain adaptation for semantic segmentation. Adv. Neural. Inf. Process. Syst. **33**, 10869–10880 (2020)
28. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
29. Peng, D., Lei, Y., Hayat, M., Guo, Y., Li, W.: Semantic-aware domain generalized segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2594–2605 (2022)
30. Qiao, F., Zhao, L., Peng, X.: Learning to learn single domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12556–12565 (2020)

31. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 102–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_7
32. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243 (2016)
33. 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)
34. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: DACS: domain adaptation via cross-domain mixed sampling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1379–1389 (2021)
35. Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M., Yang, Q.: Transfer learning with dynamic distribution adaptation. ACM Trans. Intell. Syst. Technol. (TIST) **11**(1), 1–25 (2020)
36. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: simple and efficient design for semantic segmentation with transformers. Adv. Neural. Inf. Process. Syst. **34**, 12077–12090 (2021)
37. Yao, K., Su, Z., Yang, X., Sun, J., Huang, K.: Explore epistemic uncertainty in domain adaptive semantic segmentation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 2990–2998 (2023)
38. Yu, C., Wang, J., Chen, Y., Huang, M.: Transfer learning with dynamic adversarial adaptation network. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 778–786. IEEE (2019)
39. Yu, F., et al.: BDD100K: a diverse driving dataset for heterogeneous multitask learning. In: Conference on Computer Vision and Pattern Recognition (2020)
40. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023–6032 (2019)
41. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Commun. ACM **64**(3), 107–115 (2021)
42. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint [arXiv:1710.09412](https://arxiv.org/abs/1710.09412) (2017)
43. Zhang, J., et al.: Transformer based conditional GAN for multimodal image fusion. IEEE Trans. Multimedia **25**, 8988–9001 (2023)
44. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
45. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: unsupervised learning by cross-channel prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1058–1067 (2017)
46. Zhao, Y., Zhong, Z., Luo, Z., Lee, G.H., Sebe, N.: Source-free open compound domain adaptation in semantic segmentation. IEEE Trans. Circuits Syst. Video Technol. **32**(10), 7019–7032 (2022)



What Should Insect Brains Forget?

Koichiro Yamauchi^(✉) and Takahiro Hirate

Department of Computer Science, Chubu University, 1200 Matsumoto-cho,
Kasugai-shi, Aichi 487-8501, Japan
k.yamauchi@fsc.chubu.ac.jp

Abstract. This study proposes learning methods that work with limited resources as a model of the learning behavior of the fly brain, namely, the mushroom body. Recent research on the fly's mushroom body has shown that some of its output neurons (MBONs) are activated by unknown odors. However, these effects were quickly suppressed by repeated exposure to the same odor. It appears that such MBON behaviors reflect learning of odors. We were interested in how flies could continue to learn about odors throughout their lives with their small brains. It has been suggested that learning about new odors can help the fly to forget its existing memories. Considering this, we hypothesized that the main reason for continual learning was that it serves as a strategy to forget. To test the validity of this hypothesis, we created three models using kernel perceptron. This is suitable for estimating the ongoing learning capacity within a budget. Through computer simulation and theoretical analysis, the model demonstrated the importance of having a forgetting mechanism for two reasons. One is prepare for the next new learning, and the other, is to reduce the negative effects of deleting memories.

Keywords: Insect brain · Kernel perceptron · Forgetting · Incremental learning on a budget · Drosophila mushroom body · Mushroom body output neuron (MBON)

1 Introduction

To develop technological breakthroughs, the imitation of biological systems is sometimes valuable. In the present study, we focused on the Drosophila brain, which contains only a small number of neurons. With its limited resources, performing incremental learning is difficult. In this study, we aimed to gain insights into how small brains learn by considering the Drosophila brain.

The Drosophila nerve center is a mushroom body that mainly consists of Kenyon cells [3], mushroom-output neurons (MBONs), and dopaminergic neurons (DANs). It is well known that some of the DANs encode the predicted reward of the current situation and their output signal affects the connection between Kenyon-cells (KCs) and MBONs. Moreover, DANs appear to control

This work was supported by JSPS KAKENHI Grant Number 22K12176.

the learning of MBONs, which in turn, send their output signals to other areas of the insect brain. MBONs are thought to play roles in associative learning, including olfactory and spatial memory, as well as the integration of multisensory information. There are 15 compartments in MBONs, each with its own function. In this study, we focus on the $\alpha'3$ compartment, which is related to alerting behaviors.

In the $\alpha'3$ compartment, MBONs and DANs are highly activated by novel odor stimuli. However, their activities are quickly suppressed by repeated stimulations from the same odor. This phenomenon suggests that MBONs and DANs undergo incremental learning. The capacity of this area depends on the number of orthogonal vectors in the KCs. One odor can activate 5% – 10% of KCs. This translates to 10–20 orthogonal feature vectors and implies that MBONs can store 10–20 memory units. However, 10–20 memory units may be too few for the Drosophila to survive for a month. This prompted us to ask, "How does the fly continue learning with such limited resources?" To answer this question, we develop a learning model for MBONs and DANs using a kernel perceptron. The class of kernel perceptron learning method is suitable for mistake-bound analysis. From this analysis, we aimed to determine Drosophila's learning properties via the mistake-bound mode.

We chose 'Forgetron' [5] as the base model for this analysis. Forgetron reduces its weight parameters during incremental learning such that the model fits the learning strategy of the MBON. However, the mistake-bound model assumes that the learning machine forgets the oldest memory in each round. In this case, the purpose of forgetting is to decrease the dependency on each memory and reduce the adverse effects of forgetting the oldest memory. In other words, forgetting one memory is a way of making room for new learning. We also conducted a simulation study using Forgetron and a modified version of Forgetron under changing environments. The results will assist us in determining the effect of forgetting.

The remainder of this paper is organized as follows. Section 2 reviews the Drosophila mushroom body. Section 3 shows that the $\alpha'3$ behavior can be explained by using a free-energy principle, which does not require the assumption of reward signals, and that its learning rule is represented by Hebbian learning. Section 4 presents the kernelized incremental learning of the MBON. Section 5 presents the computer simulation results. Finally, Sect. 6 discusses the results.

2 Review of Drosophila Mushroom Body

It is well known that the center of the insect brain is referred to as the mushroom body. Mushroom bodies are responsible for memory and learning in insects. Let us now focus on the Drosophila odor perception pathway in the mushroom body.

Odor is sensed by olfactory sensory neurons in the antennae, and its information is distributed to KCs by projection neurons (PNs). The connections between PNs and KCs appear to be random and do not exhibit plasticity. There are approximately 2000 KCs in the region. Each KC neuron represents a feature of the sensory inputs. KCs send their axonal fibers to MBONs. Meanwhile,

MBONs are divided into 15 compartments, each with its own set of duties. Plasticity is found in the connections between KCs and MBONs. KCs and MBONs are adjusted according to the signals from DANs. The main role of DANs is to represent the predicted reward signals. Additionally, DANs also represent the novelty of the current stimulus.

Hattori et al. [7] investigated the MBONs in the $\alpha'3$ compartment, which is activated by novel odors in the $\alpha'3$ compartment. If the $\text{MBON}\alpha'3$ s are activated, the Drosophila elicits an alerting behavior. Novelty detection of MBON activity was quickly suppressed by repeated stimulation by the same novel odor. This suggests that Drosophila learns to recognize a novel odor quickly by repeated stimulation, which renders the odor progressively familiar. Moreover, this suggests that Drosophila has the capacity to engage in incremental learning of novel odors. Odor perception seems to be an important issue in the Drosophila's life.

3 Free-Energy Principle Can Explain $\text{MBON}\alpha'3$ Behaviors

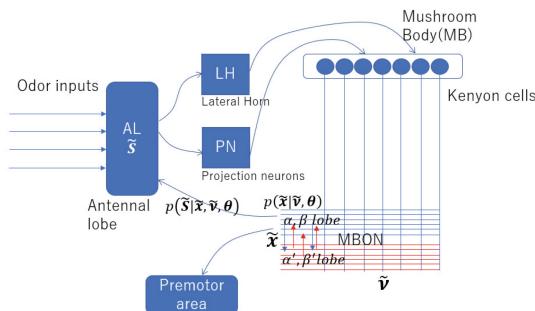


Fig. 1. Connectivity of mushroom body output neurons and the free-energy principle. (Extended figure in [2]).

proposed by Friston et al. (e.g. [6, 13]). Isomura [11] also followed the free-energy principle and provided a statistical interpretation. The free-energy principle is used to model biological behaviors under the biologically inferred assumption that action and learning are performed to reduce "surprises or unexpected situations". Interestingly, free-energy-principle-based computational models do not always assume the existence of reward signals.

In the free-energy method, the biologically inferred learning systems are assumed to maximize

$$p(\tilde{S}, \mathbf{u}, \boldsymbol{\theta}) = p(\tilde{S}|\mathbf{u}, \boldsymbol{\theta})p(\tilde{x}|\tilde{\nu}, \boldsymbol{\theta})p(\boldsymbol{\theta}), \quad (1)$$

where \tilde{S} , $\mathbf{u} = (\tilde{x}, \tilde{\nu})$ and $\boldsymbol{\theta}$ denote a generalized sensory input vector, a generalized hidden state vector, and parameter vector respectively¹.

¹ $\tilde{S} = (s^{'}, s^{''}, \dots)$, $\tilde{x} = (x^{'}, x^{''}, \dots)$ and $\tilde{\nu} = (\nu^{'}, \nu^{''}, \dots)$.

$\text{MBON}\alpha'3$ is activated by a novel odor stimulus. Although existing computational models for MBONs are usually based on the reinforcement of learning, the model was not suitable for $\text{MBON}\alpha'3$. As a novel stimulus cannot be guaranteed to be related to a reward, we decided to construct an MBON model without explicit reward signals.

We assume that

$\text{MBON}\alpha'3$ behaviors are determined by a free-energy principle

and the free-energy principle

is used to model biological behaviors under the biologically inferred assumption

that action and learning are performed to reduce "surprises or unexpected situations".

Interestingly, free-energy-principle-based computational models do not

always assume the existence of reward signals.

Although the system is assumed to minimize surprises, (defined as $S = -\log(\int p(\tilde{S}, \mathbf{u}, \boldsymbol{\theta}) d\mathbf{u} d\boldsymbol{\theta})$), the direct minimization of S is hard to execute. A realistic way to reduce this is to minimize the free-energy function [10]

$$F \equiv E_{q(\mathbf{u}, \boldsymbol{\theta})}[-\log p(\tilde{S}|\mathbf{u}, \boldsymbol{\theta})] + \mathcal{D}_{KL}[q(\mathbf{u}, \boldsymbol{\theta})||p(\mathbf{u}, \boldsymbol{\theta}|\tilde{S})], \quad (2)$$

where $\mathcal{D}_{KL}()$ is the Kullback-Leibler divergence, $q(\mathbf{u}, \boldsymbol{\theta})$ denotes a recognition distribution, and $p(\mathbf{u}, \boldsymbol{\theta}|\tilde{S})$ denotes a true \mathbf{u} distribution.

Under the free-energy principle, we assume the existence of a generative model that represents $p(\tilde{S}|\mathbf{u}, \boldsymbol{\theta})$. MBONs have backward connections to the antennal lobe (AL) [9]. The backward connections might be represented in the generative model $p(\tilde{S}|\mathbf{u}, \boldsymbol{\theta}) = \mathcal{N}(g(\mathbf{u}, \boldsymbol{\theta}), \Sigma_s)$ in the Drosophila brain (see Fig. 1).

MBON $\alpha'3$ is assumed to be a part of the generative model for the hidden state vector $p(\tilde{\mathbf{x}}|\tilde{\boldsymbol{\nu}}, \theta) = \mathcal{N}(f(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\nu}}, \theta), \Sigma_x)$, where the causes $\tilde{\boldsymbol{\nu}}$ is also assumed to be the output vector from KCs.

The minimization of surprise means the learning machine tends to acquire a new unknown instance when encountering the instance. This learning fashion is suitable for the kernelized learning style as described in the next section. Moreover, Friston and Isomura also showed that the Hebbian learning method is applicable to minimize surprise. This finding is also suitable for our hypothesis for the proposed learning model explained in the next sections.

4 Kernelized Incremental Learning on a Budget as a Model of MBON $\alpha'3$

In this study, we built a model of the mushroom body output neurons (MBONs) in the $\alpha'3$ compartment of the Drosophila. Compartment $\alpha'3$ consists of a single MBON layer with related dopamine neurons (PPL1 $\alpha'3$) that control alerting behaviors in response to novel odors. The capacity depends on the number of orthogonal vectors in the KCs (usually 10–20). Hattori et al. [7] found that the learning of a novel odor reduces existing memory in the MBON $\alpha'3$. This suggests that learning about new odors leads to the removal of a portion of the memory. The memory removal facilitates future learning of new odors. To represent this process, the learning model for MBON $\alpha'3$ should include a weight decay property for reducing existing memories. Several hypotheses on how this can be done have been proposed, including weight decay. In this study, we considered three hypotheses for removing a portion of memory in our model as follows:

1. Taking away a portion of the memory to make room for the future learning of new odors.
2. Reducing adverse effects due to the pruning of the oldest memory. (If the oldest memory contributes to proceeding to the tasks, pruning degrades the performance. However, if the weight is small enough to be ignored, the adverse effects are also small.)
3. Adjusting to concept-drifting environments whereby the Drosophila must change its behavior even if the stimuli are the same as in the past. To do so, the Drosophila must forget past experiences.

Similar to other models [1, 15], the proposed MBON model performed Hebbian learning guided by DANs. The main difference is that our model is based on the kernel method, and aims to evaluate incremental learning on a fixed budget. The proposed model does not use reward signals, assuming that learning is performed according to the free-energy principle. Therefore, the model does not require a reinforced learning environment.

4.1 Preliminary: Kernelized Representation of MBONs

Although there is a high possibility that the $\alpha'3$ compartment employs anti-Hebbian learning, its functionality can also be represented by a Hebbian learning model. Therefore, in this study, we applied and described kernelized Hebbian learning, which represents the function for simplicity.

As described in Sect. 2, approximately 2000 KCs represent a random feature vector of sensory inputs that MBONs receive through synaptic connections. An odor activates approximately 5%-10% of the 2000 KCs. These sparse feature vectors are nearly orthogonal to each other. Thus, the sensory inputs are projected onto a higher-dimensional space as an orthogonal transformation, and the MBONs process high-dimensional feature vectors for classification or regression. It is well known that a high-dimensional feature vector $\Phi(\tilde{\mathbf{S}}_1)$ and $\Phi(\tilde{\mathbf{S}}_2)$, whose vector dot product is converted from input vectors $\tilde{\mathbf{S}}_1$ and $\tilde{\mathbf{S}}_2$ by using a specialized nonlinear function $\Phi(\cdot)$, their vector dot product $\Phi(\tilde{\mathbf{S}}_1)^T \Phi(\tilde{\mathbf{S}}_2)$ is represented by a reproduction kernel function $K(\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2)$ (e.g. [17]). One of such kernel functions is a Gaussian kernel function such that

$$K(\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2) \equiv \exp\left(-\gamma \|\tilde{\mathbf{S}}_1 - \tilde{\mathbf{S}}_2\|^2\right), \quad (3)$$

where γ is a coefficient, which depends on the nonlinear function $\Phi(\cdot)$.

The Hebbian and anti-Hebbian learning can be represented as adding or subtracting a target output vector \mathbf{a}_{target} for $\tilde{\mathbf{S}}_{target}$ to or from W_{MBON} . In the kernelized algorithm, this corresponds to adding or pruning the kernel $\mathbf{a}_{target} K(\tilde{\mathbf{S}}_{target}, \cdot)$. We approximated this process using a kernel perceptron with a limited number of kernels bounded by the capacity (budget) B

4.2 Shared Structure of the Three Models: Supervised Kernel Learning Model for the Mushroom-Body $\alpha'3$

In this paper, we propose three MBON $\alpha'3$ learning models. The three differed primarily in their pruning strategies. In this section, the shared structures of the three models are described.

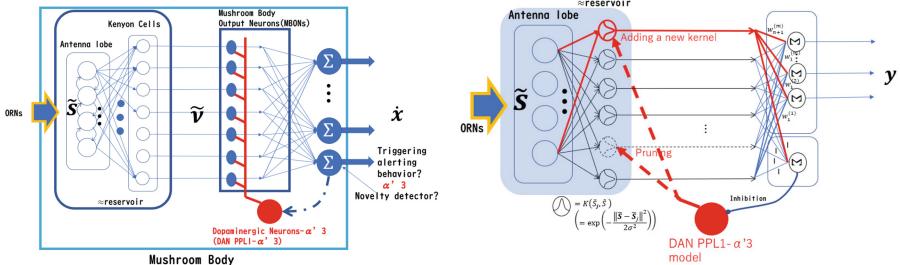


Fig. 2. MBON and its kernelized model. Left: A rough scheme of the mushroom body output neuron $\alpha' 3$. Right: The kernelized MBON model.

We assumed that the model learns $\{\tilde{\mathbf{S}}_p\}_{p=1}^T$ in an online learning manner, where $\tilde{\mathbf{S}}_p \in \mathbf{R}^n$. $\tilde{\mathbf{S}}_p$ represents the generalized current sensory input vector. The MBON predicts the output vector: $\mathbf{y} \in \mathbf{R}^m$. Note that \mathbf{y} should be $\dot{\mathbf{x}}$ under ordinary circumstances. However, in this study, we set \mathbf{y} as a class label to focus on the MBON learning ability within a budget. $\mathbf{y}_t = [y_t^{(1)}, \dots, y_t^{(k)}, \dots, y_t^{(m)}]^T$, where $y_t^{(k)} \in \{-1, 1\}$.

The output of the model at time t and its kernelized representation are

$$f_{mbon}^{(k)}[\tilde{\mathbf{S}}] = \sum_{i \in S_t}^B w_i^{(k)} \sigma_{i,t} K(\tilde{\mathbf{S}}_i, \tilde{\mathbf{S}}), \longleftrightarrow f_{mbon}^{(k)} = \sum_{i \in S_t}^B w_i^{(k)} \sigma_{i,t} K(\tilde{\mathbf{S}}_i, \cdot) \quad (4)$$

where B is the upper bound of the number of hidden units, and S_t is the support set² at time t . $\sigma_{i,t}$ is the ratio of decay (forgetting) and $\sigma_{i,t} \leq 1$. $\sigma_{i,t}$ is reduced each round by $\sigma_{i,t+1} = \phi_t \sigma_{i,t}$, where $\phi_t \leq 1$. Note that an excessively large decay ratio damages f_{mbon} and decreases its accuracy. However, pruning the old kernel without forgetting it (shrinking) also damages f_{mbon} .

The learning samples are presented individually and alternately through repeated inference and learning. In each round, if there is no unit whose output is larger than a threshold, a new unit (kernel) is added : $f_{mbon,t}^{(k)'} \equiv f_{mbon,t}^{(k)} + y_t^{(k)} \sigma_{t,t} K(\tilde{\mathbf{S}}_t, \cdot)$. The weights for each kernel is shrinks after that : $f_{mbon,t}^{(k)''} \equiv \phi_t f_{mbon,t}^{(k)'}$. If the total number of hidden units (kernel functions) exceeds budget B , one of the units is removed. For example, the oldest unit i is removed : $f_{mbon,t+1}^{(k)} = f_{mbon,t}^{(k)''} - w_i^{(k)} \sigma_{i,t} K(\tilde{\mathbf{S}}_i, \cdot)$ (see Algorithm 1).

Note that the oldest unit j has the smallest $\sigma_{j,t}$ because $\sigma_{i,t}$ for $\forall i \in S_t$ are shrunk by $\sigma_{i,t} = \phi_t \sigma_{i,t}$ in each round. So, the hidden unit with the smallest $\sigma_{j,t}$ is removed. Using this strategy, the MBON model maintains the total number of hidden units at B . The performance is evaluated by using the cumulative error $CumulativeErr$. $CumulativeErr$ is increased if $f_{mbon}^{(k)}[\mathbf{x}(t)]$ makes a mistake

² The support set is the set of learned samples stored in the model.

before the learning of current sensory input. If it makes the mistake, $f_{mbon}^{(k)}[\mathbf{x}(t)]$ learns current sensory input.

Algorithm 1. Kernelized MBON learning for model1, model2 and model3

Require: $modelID (\in \{model1, model2, model3\})$, current time t , new input $\tilde{\mathbf{S}}_t$, Support set $S_t (|S_t| \leq B^1)$, activation threshold: θ , budget : B , previous $\mathbf{f}_{mbon,t}$, decay ratio : ϕ_t .

```

1: Calculate  $f_{mbon,t}^{(k)}[\tilde{\mathbf{S}}_t]$  for all  $k$  by (4).
2: receive  $\mathbf{y}_t$  {Get label.}
3: if  $modelID == model3$  then
4:    $winner = \arg \max_j K(\tilde{\mathbf{S}}_j, \tilde{\mathbf{S}}_t)$ 
5:    $\sigma_{winner,t} = 1$ 
6: end if
7: if  $IsNovel(modelID, \tilde{\mathbf{S}}_t)$  then
8:    $f_{mbon,t}^{(k)} \equiv f_{mbon,t}^{(k)} + \mathbf{W}_{t+1}^{(k)} \sigma_{t,t} K(\tilde{\mathbf{S}}_t, \cdot)$ , ( $S_{t+1} = S_t \cup \{t\}$ )
9:    $\sigma_{t,t} = 1$ 
10:   $\mathbf{W}_{t+1}^{(k)} = \mathbf{y}_t^{(k)}$  {initialize  $\mathbf{W}_{t+1}^{(k)}$ }
11:  for all  $i \in S_{t+1}$  do
12:     $\sigma_i = \phi_t \sigma_i$  {Decay  $\sigma_i$ }
13:  end for
14:  if  $|S_{t+1}| \geq B$  then
15:     $p = \arg \min_i \sigma_{i,t}$ ,
16:    if  $PruningCondition(modelID, \sigma_{p,t})$  then
17:       $f_{mbon,t+1}^{(k)} = f_{mbon,t}^{(k)} - w_p^{(k)} \sigma_{p,t} K(\tilde{\mathbf{S}}_i, \cdot)$  ( $S_{t+1} = S_{t+1} \setminus \{p\}$ ) {Pruning the
oldest kernel}
18:    end if
19:  end if
20: end if
21:  $t = t + 1$ 
22: return  $\mathbf{f}_{mbon,t}$ 

```

4.3 Three MBON Models

The pseudo codes for the three models are unified in Algorithm 1.

Model 1. The first model removes the oldest kernel, whose σ_i is less than θ , when $|S_t| \geq B$. This means that if the oldest kernel's σ_i is larger than θ , the MBON model does not learn to recognize the current input. Therefore, if Drosophila frequently encounters a novel sample, it often ignores it. In Algorithm 1 lines 7 and 13 are $IsNovel(model1, \tilde{\mathbf{S}}_t) = \{\forall j K(\tilde{\mathbf{S}}_j, \tilde{\mathbf{S}}_t) < \epsilon\}$ and $PruningCondition(model1, \sigma_p) = \{\sigma_p < \epsilon\}$.

Model 2: Modified Forgetron. We designed a model based on a slightly modified Forgetron [5]. The Forgetron is a kernel perceptron class-learning

machine (e.g., [4, 8, 12, 14, 16]) that introduces a shrinking-weight (forgetting) mechanism. The Forgetron also removes the oldest kernel, whose σ_i is the smallest in S_t . Therefore, $IsNovel(model2, \tilde{S}_t) = \{\prod_{k \in Outputs} y_t^{(k)} f_{mbon,t}^{(k)}[\tilde{S}_t] \leq 0\}$. Although the Forgetron does not ignore any novel samples, the removal of the oldest kernel makes the cumulative error increase. Therefore, $PruningCondition(model2, \sigma_{p,t}) = \{\sigma_{p,t} < \infty\}$. The basic Forgetron in [5] controls a certain ϕ_t to balance them under $B \geq 83$ and single output dimension conditions at every round. In this paper, $\phi_t < 1$ is fixed to a certain value to get minimum mistakes under multiple output dimension condition.

The model is analyzed theoretically in detail (see Sect. 4.4), and the results showed that the shrinking-weight mechanism helps to reduce the mistake-bound of incremental learning on a fixed budget. Therefore, if we can use the Forgetron as a model for MBON learning, we can demonstrate that the forgetting strategy in MBON is essential for learning under limited capacity.

Model 3: Least Recently Used. Both Model1 and 2 work by pruning the oldest kernel when $|S_t| \geq B$. However, the oldest kernel is not always the most useless kernel. The most useless kernel is the least used kernel. If the situation does not change for sometime then, the kernel that has been used the least recently will also be the kernel that will be used least in the near future. Therefore, the least-recently used kernels should be removed. $IsNovel(model3, \tilde{S}_t)$ and $PruningCondition(model3, \sigma_{p,t})$ are the same as those of model2. However, the $\sigma_{i,t}$ is recovered to 1 when the i -th kernel is maximum activated (see Algorithm 1 line 3–6).

4.4 ϕ that Minimizes the Mistake Bound of Model2

Let's consider the shurinkage ratio ϕ that minimizes the mistake bound of the model2. If such ϕ is less than 1, the forgetting might be essential for MBON α '3.

To this end, we simplify our model in the following manner without losing generality.

- The multidimensional output of our model is switched to one-dimensional output.
- The sensory input $\hat{S}_t \in R^n$ is substituted with a simple nortation $\mathbf{x}_t \in R^n$.
- The kernel function used in Eq. 4 can be regarded as a reproducing kernel so that $f_{mbon}[\mathbf{x}]$ can be represented as

$$f_{mbon} \equiv \sum_{j \in I_t} w_j \sigma_{j,t} K(\mathbf{x}_j, \cdot), \quad (5)$$

where $k(\mathbf{x}_i, \cdot)$ is an infinite-dimensional vector converted from \mathbf{x}_i , B denotes the upper bound of the number of kernels. Note that $f_{MBON}(\mathbf{x}) = \langle f_{MBON}, K(\mathbf{x}, \cdot) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes dot product operation. Similar to our proposed model, f_{MBON} learns samples $\chi = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$ online.

In this analysis, f_{MBON} was evaluated using the *hinge-loss* function:

$$l(f; (\mathbf{x}, y)) \equiv \begin{cases} 0 & \text{if } yf(\mathbf{x}) \geq 1 \\ 1 - yf(\mathbf{x}) & \text{otherwise} \end{cases} \quad (6)$$

Dekel et al. [5] showed the mistake bound M in lemma 5.2 in their paper. In their method, the number of kernel (=Budget) is assumed to be larger than 83. In our method, however, we have to assume the budget is around the capacity of MBONs 20. To adjust the original lemma to our setting, the lemma is re-derived.

In the following, we also assume the following:

- Let g be a desired competitor function in the Hilbert space $\mathcal{H}k$ such that $|g| \leq U$, where U is defined as $U \equiv \frac{1}{4} \sqrt{\frac{B+1}{\log(B+1)}}$.
- Let $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_T, \mathbf{y}_T)$ be a sequence of examples such that $K(\mathbf{x}_t, \mathbf{x}_t) \leq 1$ for all t .
- Let J be a set of examples, where f_{MBON} mistakes the predicted label: $J = \{t | y_t f_{MBON}(\mathbf{x}_t) < 0\}$.
- Let $f'_{MBON,t}$ be $f'_{MBON,t} = f_{MBON,t}(\mathbf{x}) + y_t K(\mathbf{x}, \cdot)$
- Let $f''_{MBON,t}$ be $f''_{MBON,t} = \phi_t f'_{MBON,t}$.
- Let Ψ_t be a magnitude of loss due to pruning the old kernel and

$$\Psi_t = \begin{cases} \Psi(\sigma_{r_t, t+1}, y_{rt} f''_{MBON,t}(\mathbf{x}_{rt})) & \text{if } t \in J \wedge |S_t| = B \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where $\Psi(\lambda, \mu) \equiv \lambda^2 + 2\lambda - 2\lambda\mu$ and the suffix r_t denotes the removed kernel at t .

Then, the number of mistakes of f_{MBON} satisfies the following inequality.

$$M \leq \|g\|^2 + 2 \sum_{t=1}^T l_t^* + \underbrace{\left(\|g\|^2 \log \left(\prod_{t \in J} (1/\phi_t) \right) + \sum_{t \in J} \Psi_t \right)}_{D(\phi_t)} \quad (8)$$

where g denotes the desired competitor function, $l_t^* = l(g; (\mathbf{x}_t, y_t))$. The detailed derivation of Eq. (8) is presented in [5]. From this equation, we can observe that M is larger than the mistake bound of the original kernel perceptron, which does not use shrinking or pruning kernels: $M \leq |g|^2 + 2 \sum_{t=1}^T l_t^*$.

In the basic Forgetron, which is described in Fig. 5.1 in [5], ϕ_t is adaptively controlled so that ϕ_t is not a constant value. By adapting ϕ_t , Dekel et al. showed that the number of prediction mistakes of the basic Forgetron is bounded as $M \leq 2|g|^2 + 4 \sum_{t=1}^T l_t^*$. To investigate the real duty of forgetting in the drosophila MBON, we assume that ϕ_t is fixed to $\phi \leq 1$ for all t . Note that the MBON model reduces the mistake bound when $\phi < 1$, and shrinking (forgetting) is essential to maintain the mistake bound low.

By using $\|g\| \leq U$, and $|J| \leq T$, we can derive the following lemma.

Lemma 1 ϕ denotes the shrinking coefficient for the decay ratio of $\sigma_{i,t}$ $\forall i \in S_t$ and $0 \leq \phi \leq 1$. $U = \frac{1}{4} \sqrt{\frac{B+1}{\log(B+1)}}$. $D(\phi)$ is bounded by

$$D(\phi) \leq -U^2 T \log \phi + (T - B) \{2\phi^{2B} + 2\phi^B(1 + U)\}$$

Note that the first term of the right-hand side of $-U^2 T \log \phi$ is reduced to zero when ϕ is increased to 1 but is increased when ϕ is close to zero. On the other hand, the second term of the right-hand side of $(T - B) \{2\phi^{2B} + 2\phi^B(1 + U)\}$ is increased if ϕ is increased to 1 but decreases if ϕ is decreased to zero. So, the sum of these two terms becomes minimum at an intermediate point in $0 < \phi < 1$ (see Fig. 3). If ϕ is set to the minimum point, which is less than 1, the MBON prevents the number of prediction mistakes from being minimum. The optimal ϕ is given as the solution of $\partial[-U^2 T \log \phi + (T - B) \{2\phi^{2B} + 2\phi^B(1 + U)\}] / \partial \phi = 0$. Therefore, if $B \ll T$,

$$\phi_{opt} \simeq \left\{ \frac{1+U}{2} + \sqrt{\frac{(1+U)^2}{4} + U^2} \right\}^{1/B} \leq 1 \quad (9)$$

The findings of this study suggest that MBONs must forget past memories in a certain ratio to reduce the number of prediction mistakes.

5 Computer Simulation

In this section, we described the experiment conducted to verify the performance of the models in changing environments. Through these simulations, we demonstrated that forgetting contributes to reducing cumulative errors.

5.1 Assumed Environments and Their Markov-Chain Model

For this purpose, we use a Markov chain model to generate changes in the environment as follows. Therefore, the 5-dimensional generalized sensory inputs at time t : $\tilde{\mathbf{S}}_t$ is generated from the following multivariate normal distribution, whose covariance matrix is $\Sigma = diag(0.01)$.

$$\tilde{\mathbf{S}}_t \sim \mathcal{N}(\mathbf{m}(e_t), \Sigma), \quad e_t \sim p(e|e_{t-1}) \quad (10)$$

where e_t denotes current state and $e_t \in \{e_1, e_2, \dots, e_{20}\}$. $\mathbf{m}(e_t)$ denotes the mean vector for current state and is a fixed random vector defined for each e_i ($i = 1, 2, \dots, 20$). The state change is simply defined by a staying state probability p as follows:

$$p(e_i|e_j) \equiv \begin{cases} \frac{p}{2} & i = j \\ \frac{1-p}{2} & i \neq j \end{cases} \quad (11)$$

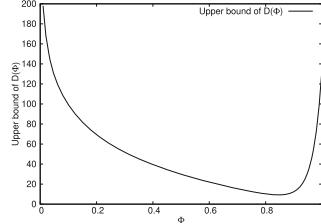


Fig. 3. An example of the upper bound of $D(\phi)$ vs ϕ when $B = 20$ and $T = 100$. Note $\phi = 1$ means no shrinking.

We generated synthetic datasets of nine types with $p = 0.1, 0.2, 0.3, \dots, 0.9$. Each dataset size was 2000. For each \tilde{S}_t , 50 varied datasets were prepared (see Fig. 4). In each dataset, each \tilde{S}_t is stored together with its corresponding state vector $\mathbf{y}_i = \text{OneHot}(e_i)$ as its label. To fit to the kernel perceptron learning, the k -th element of \mathbf{y}_i is $y_i^{(k)} \in \{-1, 1\}$. During the computer simulation, $(\tilde{S}_t, \mathbf{y}_i)$'s were presented to each model.

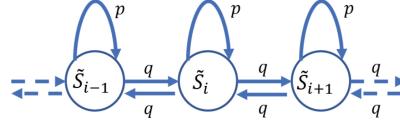


Fig. 4. The distributions used in the experiments. p denotes the probability of steady state. q denotes the state change probability. Note that $p + 2q = 1$.

Evaluation Method. The experiments were repeated at different state-change rates. The cumulative error, which is the value obtained by summing the errors between the values output by the training and the correct labels, was used as an evaluation measure Eq. 12. This experiment evaluates the MBON model described in Sect. 4.

For all models, the single-output model was extended to a multiple-output model to conduct the experiment. The multiple outputs used in this simulation are one-hot representations of the predicted label of the current input vector $\mathbf{x}(t)$. Therefore, the cumulative error is modified according to the following equation for each round:

$$\text{cumerror}+ = \begin{cases} 1 & \text{IsNovel(model2, } \hat{S}_t) == \text{True} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where $\text{IsNovel}(\text{model2}, \hat{S}_t)$ is defined in Sect. 4.3. Note that modification of cumerror should be executed before updating the $\mathbf{f}_{MBON,t}$ parameters in each round.

5.2 Results

The shared network size: input size, kernel size, and output size were set to 5-20-20. The shrinkage ratio was set to $\phi = 0.985^3$

³ From Lemma 1, we should determine ϕ by Eq. 9. However, in this simulation setting, the MBON model has 20 outputs. This does not directly correspond to the lemma. Therefore, we have set the value manually.

An Example of Model Behaviors. Figure 5 shows an example of the cumulative errors of the MBON models 1,2,3 and their competitors (with and without the weight shrinkage) at $p = 0.9$.

In this test, the budget of all models was set to 20, γ was set to 4.0. Moreover, the pruning condition for model1 was set as: $\text{PruningCondition}(\text{model1}, \sigma_p) = \{\sigma_p < 0.1\}$.

In this example, the cumulative error order was: model3 < model2 < model1 without weight shrinkage < model1 < model1 with weight shrinkage. The cumulative error for model1 is larger than model2 and 3. From the next section, the detailed performance of each model is shown.

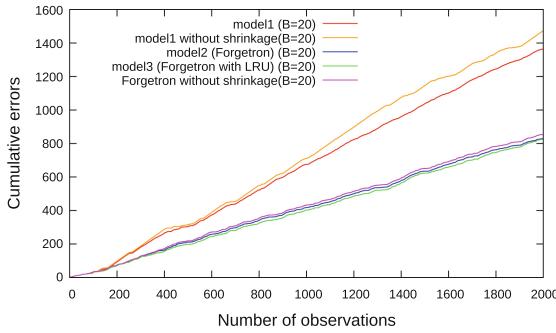


Fig. 5. An example of cumulative error vs number of observations (Budget=20, $p=0.9$) for Model1, Model2 (Forgetron), Model3 (Forgetron with LRU), and their competitors. The competitors are Model1 without weight shrinkage and Model2 without weight shrinkage. (Magnify to see the magnitude relation clearly.)

Model1 vs Model1 Without Weight Shrinkage. To evaluate the effect of weight shrinkage (forgetting), the final cumulative errors after learning 2000 samples were evaluated for model1 and model1 without shrinkage. Figure 6 shows the points, each representing the cumulative error of $(\text{model1}, \text{model1withoutshrinkage})$. Points located above the line $y = x$ indicate that the cumulative error for model1 without shrinkage is larger than that of model1. From the figure, we can see that the distribution of points is dominated above $y = x$, meaning that model1 without weight shrinkage yields larger cumulative errors. The precise ratio of points above $y = x$ was 85.3% in all trials (450 trials). This ratio shows that weight shrinkage (forgetting) has an advantage in reducing cumulative error by making memory space for new learning. However, weight decay (shrinkage) can damage part of the memory, occasionally resulting in larger cumulative errors.

Model2 (Forgetron) vs Model2 Without Weight Shrinkage. Similar to the previous section, the cumulative errors after learning 2000

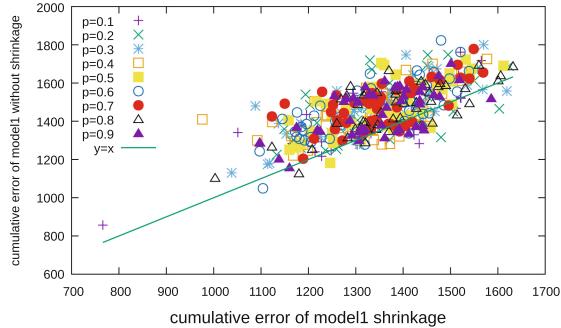


Fig. 6. Cumulative errors of Model 1 vs. those of Model 1 without weight shrinkage. In 450 trials, 85.3% of the cumulative errors of Model 1 were less than those of Model 1 without weight shrinkage.

samples were examined. Figure 7 shows the cumulative error points for (*model2, model2withoutshrinkage*). We can see from the figure that the distribution of points is dominated above the line $y = x$. Precisely, the ratio of points above the line was 97.3% of the 450 trials. This means that weight shrinkage has the advantage of reducing cumulative errors.

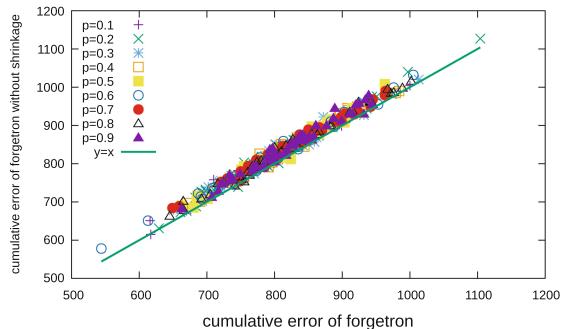


Fig. 7. Forgetron cumulative errors vs Forgetron without weight shrinkage after learning of the 2000 samples. In about 97.3% of the 450 trials, the cumulative errors of Forgetron were less than those of Forgetron without weight shrinkage.

5.3 Model3: LRU vs Forgetron (model2)

Finally, we conducted a comparison between model2 (Forgetron) and model3 (Forgetron with LRU). Similar to the previous two sections, the cumulative errors after learning 2000 samples were compared. Figure 8 shows the cumulative error points for (*model2, model3*). In the figure, points below the line $y = x$ mean that the cumulative errors of model3 are smaller than those of model2. We can see that

the distribution of points is slightly dominated on the lower side. More precisely, the ratio of points below $y = x$ was 59.1% of the 450 trials. This means that the least recently used (LRU) strategy is slightly better than model2 (Forgetron). However, LRU predicts the future usage of each memory based on past memory usage history. The prediction is sometimes incorrect due to variations in inputs. In such cases, the cumulative error of model3 increases. Moreover, the removed kernel is not the oldest kernel, so the damage due to the removal might be larger than that of model 2.

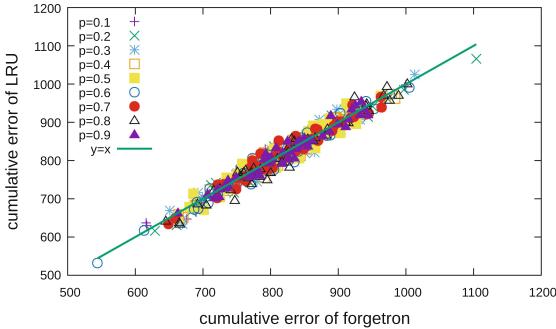


Fig. 8. Comparison of cumulative errors between Forgetron and Forgetron with LRU after learning 2000 samples. The ratio of points below $y = x$ was 59.1% across 450 trials.

6 Discussion and Conclusion

We proposed a kernelized learning model for the Drosophila mushroom body output neurons (MBONA). As MBONA are highly activated by novel stimuli, this could not be modeled as a reinforced learning model. To overcome this difficulty, we attempted to explain the behaviors using the free-energy principle. Under this principle, the learning of biological systems tends to minimize free energy, and the learning rule can be represented by the Hebbian learning rule. Next, we presented a simplified learning model. Consequently, the Hebbian rule was converted into a kernelized learning rule to analyze the incremental learning ability on a budget. We demonstrated three variations of kernelized learning (Model 1, Model 2 Forgetron, and Model 3 LRU). Theoretical analyses and computer simulations were conducted. Results from the theoretical analysis showed that the forgetting property of MBONs contributed to the maintenance of a low cumulative error.

The simulation showed that Model 1 and Model 2 (forgetting model) reduced cumulative errors, with Model 2 outperforming Model 1. Model 3 (LRU) slightly outperformed Model 2 (Forgetron). LRU's mechanism wasn't related to weight decay, but weight decay helped identify kernels to prune. LRUs drawback is

that pruning non-oldest kernels can harm outputs. Extending LRU can improve performance. These models showed unique behaviors at capacity limits. In the Forgetron framework, the oldest kernel was deleted unconditionally, enhancing adaptability and quick response to new data.

On the other hand, future experiments are needed to confirm whether such learning actually occurs in the *Drosophila* brain.

A Derivation of Lemma 1

Proof. From the definition, we obtain

$$D(\phi) = -\|g\|^2 \log \left(\prod_{t \in J} \phi \right) + \sum_{t \in J \wedge |S_t|=B} \Psi(\sigma_{r_t, t+1}, y_{r_t} f_t''(\mathbf{x}_{r_t})).$$

Since $\sigma_{r_t, t+1} = \phi^B$, Ψ can be rewrited as

$$\Psi(\sigma_{r_t, t+1}, y_{r_t} f_t''(\mathbf{x}_{r_t})) = \phi^{2B} + 2\phi^B - 2\phi^B y_{r_t} f_t''(\mathbf{x}_{r_t})$$

From Cauchy-Schwarz inequality, $|f_t''(\mathbf{x}_{r_t})| \leq \|f_t''\| \cdot \|K(\mathbf{x}_{r_t}, \cdot)\| \leq U$. Therefore, we can rewrite the above equation

$$\phi^{2B} + 2\phi^B - 2\phi^B U \leq \Psi(\sigma_{r_t, t+1}, y_{r_t} f_t''(\mathbf{x}_{r_t})) \leq \phi^{2B} + 2\phi^B + 2\phi^B U$$

The maximum number of removing the oldest kernels is $T - B$ times, $\|f_t''\| \leq U$ and $\|g\| \leq U$ so that

$$D(\phi) \leq -U^2 T \log \phi + (T - B) \{ \phi^{2B} + 2\phi^B (1 + U) \}$$

□

References

1. Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., Webb, B.: Using an insect mushroom body circuit to encode route memory in complex natural environments. *PLoS Comput. Biol.* **11**(10), 1–22 (2015). <https://doi.org/10.1371/journal.pcbi.1004683>
2. Arena, P., Patané, L., Strauss, R.: The insect mushroom bodies: a paradigm of neural reuse. In: ECAL 2013: The Twelfth European Conference on Artificial Life, pp. 765–772. MIT Press (2013). <https://doi.org/10.1162/978-0-262-31709-2-ch109>
3. Aso, Y., et al.: The neuronal architecture of the mushroom body provides a logic for associative learning. *elife sciences.org* **3**, e04577 (2014). <https://elifesciences.org/articles/04577>
4. Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast kernel Classifiers with online and active learning. *J. Mach. Learn. Res.* **6**, 1579–1619 (2005)

5. Dekel, O., Shalev-Shwartz, S., Singer, Y.: The forgetron: a kernel-based perceptron on a budget. *SIAM J. Comput. (SICOMP)* **37**(5), 1342–1372 (2008). <https://doi.org/10.1137/060666998>
6. Friston, K., et al.: Active inference and learning. *Neurosci. Biobehav. Rev.* **68**, 862–879 (2016). <https://doi.org/10.1016/j.neubiorev.2016.06.022>
7. Hattori, D., Aso, Y., Swartz, K.J., Rubin, G.M., Abbott, L., Axel, R.: Representations of novelty and familiarity in a mushroom body compartment. *Cell* **169**, 956–969 (2017). <https://doi.org/10.1016/j.cell.2017.04.028>
8. He, W., Wu, S.: A kernel-based perceptron with dynamic memory. *Neural Netw.* **25**, 105–113 (2012). <https://doi.org/10.1016/j.neunet.2011.07.008>
9. Hu, A., Zhang, W., Wang, Z.: Functional feedback from mushroom bodies to antennal lobes in the drosophila olfactory pathway. *Proc. Nat. Acad. Sci. United State Am.* **107**(22), 10262–10267 (2010). <https://doi.org/10.1073/pnas.0914912107>
10. Isomura, T.: Introduction of the free-energy principle perception, action, and inference of another's mind (in Japanese). *Brain Neural Netw.* **25**(3), 71–85 (2018)
11. Isomura, T.: A measure of information available for inference. *Entropy* **20**(7)(512), e20070512 (2018). <https://doi.org/10.3390/e20070512>
12. Kivinen, J., Smola, A.J., Williamson, R.C.: Online learning with kernels. *IEEE Trans. Sig. Process.* **52**(8), 2165–2176 (2004). <https://doi.org/10.1109/TSP.2004.830991>
13. KJ, F., J, D., SJ, K.: Reinforcement learning or active inference? *PLOS Comput. Biol.* **4**(7), e6421 (2009). <https://doi.org/10.1371/journal.pone.0006421>
14. Orabona, F., Keshet, J., Caputo, B.: The Projectron: a bounded kernel-based perceptron. In: ICML2008, pp. 720–727 (2008). <https://doi.org/10.1145/1390156.1390247>
15. Owald, D., et al.: Layered reward signalling through octopamine and dopamine in drosophila. *Nature* **492**, 433–437 (2012). <https://doi.org/10.1038/nature11614>
16. Preda, C.: Regression models for functional data by reproducing kernel Hilbert spaces methods. *J. Stat. Plann. Infer.* **137**, 829–840 (2007)
17. Schölkopf, B.: The kernel trick for distances. In: Leen, T., Dietterich, T., Tresp, V. (eds.) *Advances in Neural Information Processing Systems*. vol. 13. MIT Press (2000). https://proceedings.neurips.cc/paper_files/paper/2000/file/4e87337f366f72daa424dae11df0538c-Paper.pdf
18. Yamauchi, K.: Quick continual kernel learning on bounded memory space based on balancing between adaptation and forgetting. *Evolving Systems* (2022). <https://doi.org/10.1007/s12530-022-09476-8>



Agent Clustering and Information Sharing Underlying MADRQN for Traffic Light Cooperative Control

Haoran Cheng¹, Bo Wang¹, Jie Liu¹, and Tongchun Du^{1,2(✉)}

¹ Anhui Normal University, Wuhu City, Anhui Province, China
tcdu@ahnu.edu.cn

² Yangtze River Delta Information Intelligence Innovation Research Institute,
Anhui, China

Abstract. This paper introduces a Multi-Agent Deep Recurrent Q-Network (MADRQN) designed for real-time traffic light control across multiple intersections. The approach aims to improve joint control efficiency while minimizing communication overhead among agents. By modeling traffic light management as a Markov Decision Process and treating each intersection's controller as an agent, the MADRQN method dynamically clusters agents based on their locations and real-time observations using the Growing Neural Gas algorithm. Within each cluster, information sharing and centralized training are applied to enhance coordination. Simulation results on the Simulation of Urban MObility (SUMO) platform demonstrate that this approach reduces communication load, facilitating more effective and efficient information sharing and training. Consequently, MADRQN achieves shorter average vehicle waiting times compared to leading multi-agent deep reinforcement learning methods, significantly alleviating traffic congestion.

Keywords: Collaborative control of traffic lights · Agent cluster · Deep recurrent Q-network · Growing neural gas · Centralized training and decentralized execution

1 Introduction

Traffic congestion significantly increases energy consumption and carbon emissions. Intelligent control of traffic signals, where phases and durations are dynamically adjusted based on real-time traffic conditions, is crucial for improving traffic flow and reducing vehicle delays.

Current research on Multi-Agent Deep Reinforcement Learning (MADRL) often involves limited agents in simulated games. In Multi-Agent Markov Decision Problems (MAMDPs), ensuring stationarity and full observability during centralized training necessitates extensive information sharing, leading to communication delays and impracticality. Not all agents need to share information, especially those far apart with different observations. Effective information

sharing is essential for enhancing collaboration while minimizing data volume. Additionally, training MADRL algorithms requires numerous environment interactions, demanding high computational resources and lengthy training times.

To address the challenges in multi-intersection traffic signal control modeled as a MAMDP, we propose a new paradigm for cooperative control, tackling the state parameter dimensional explosion caused by increasing intersections. Our contributions include:

- 1. Agent Clustering Algorithm:** Using the growing neural gas (GNG) algorithm [6], we group similar agents at multiple intersections to reduce data requirements and enhance collaboration by identifying agents with comparable positions and observations. The recent works such as Sparse Pinball Twin Bounded Support Vector Clustering [28] and Least Squares Projection Twin Support Vector Clustering (LSPT SVC) [21] further guide our approach in refining agent clustering for improved efficiency and precision.
- 2. Centralized Training Algorithm:** We introduce a centralized training algorithm for the multi-agent deep recurrent Q-network (MDRQN), incorporating global value function decomposition and observation sharing to train traffic signal control models within each cluster.
- 3. Optimal Parameter Transfer:** We present a method for transferring optimal parameters from the best-performing agent to others, accelerating training and facilitating convergence towards optimal parameters for all agents.

All simulations are conducted using the SUMO platform. Comparative experiments demonstrate that our method excels in coordinated control of multi-intersection traffic lights and effectively reduces training time.

2 Related Work

Intelligent traffic signal control methods can be categorized into three main groups:

- 1. Optimization-Based Methods:** This category transforms time-sequential control into optimization problems. Examples include SCATS [15] and SCOOT [10], which rely on complex traffic models and assumptions that may not match actual traffic conditions. These methods often struggle with saturated traffic flows. Techniques such as fuzzy logic [2], evolutionary algorithms [3, 24], linear programming [17], and neural networks [22] offer various advantages but face challenges in adapting to real-time, dynamically changing multi-intersection traffic light control. For instance, fuzzy logic is better suited to single-intersection control and cannot fully capture real-time traffic uncertainties. Genetic algorithms and neural networks require substantial computational resources, and linear programming is limited to linear problems, restricting scalability and optimality.

2. Reinforcement Learning (RL) Approaches: Unlike traditional model-driven methods, RL learns optimal control strategies from data without relying on heuristics or physical models. Prashant et al. [19] introduced an RL-based signal control system using queue length and signal light duration as state variables. Liu et al. [13] proposed using linear functions for clustering and fitting Q-values, but this method only incorporates queue information, neglecting other complexities. Limited state representations in RL can lead to suboptimal control when essential features are omitted [7]. El-Tantawy et al. [4] suggested a multi-agent RL approach, but extensive numerical tables scale exponentially with the number of vehicles. Some studies have applied model-free temporal difference (TD) RL methods, such as Q-learning, to traffic optimization problems. Shao et al. [16] proposed a method using prior vehicle perception to reduce average waiting times. Haddad et al. [8] recommended simplifying state and reward structures at single intersections to facilitate training, but these often lack real-time dynamic traffic conditions, limiting their applicability.

3. Deep Reinforcement Learning (DRL) Methods: DRL combines deep learning's feature extraction capabilities with RL's sequential decision-making strengths, gaining significant traction in traffic control research. Tingga et al. [29] used deep Q-learning to optimize single-intersection control. Kang et al. [12] used vehicle carbon dioxide emissions as a reward, finding LSTM networks within the DQN framework provided superior performance. Tunc et al. [30] proposed combining fuzzy control with deep Q-learning, enhancing stability and robustness. Bálint et al. [1] introduced a novel interpretation of feature-based state representation and reward concepts. These studies demonstrate DRL's potential to significantly improve traffic signal control systems.

While single-agent DRL struggles with multi-intersection traffic due to dimensional and computational issues, multi-agent deep reinforcement learning (MADRL) better adapts to urban road network changes. MADRL offers a suitable solution for distributed traffic signal control, aligning with multi-agent system similarities.

There are two mainstream MADRL methods to enhance collaboration: centralized training with information sharing and independent training based on individual observations. Centralized training requires substantial computation and may not achieve optimal results for each agent, while independent training faces challenges with partial observability and non-stationary environments. A promising direction is hybrid approaches combining centralized training and decentralized execution, involving collaborative training frameworks and methods. Collaborative methods include:

Global Value Function Decomposition: Examples are VDN [27], QMIX [20], and QTRAN [23]. **Centralized Critics:** Examples include MADDPG [14] and COMA [5]. **Information-Sharing-Based Methods:** Examples include CommNet [26], BicNet [18], and ATOC [11]. These methods balance

centralized and decentralized approaches, fostering effective agent collaboration while addressing computational and observational challenges.

Several studies have explored MADRL for traffic light control. Su et al. [25] combined scheduling algorithms with MADRL to develop traffic routes. Yang et al. [32] introduced a causal reasoning-based MADRL algorithm to address the non-stationary nature of multi-agent traffic environments. Yan et al. [31] constructed a graph cooperative network model (GCQN-TSC) integrating self-attention and deep graph Q-learning. Huang et al. [9] proposed RELight, a traffic signal control framework based on stochastic ensemble reinforcement learning. These studies highlight MADRL's potential to improve traffic signal control by leveraging advanced algorithms and techniques to address the complexities of multi-agent environments.

3 Agent Clustering

We assume that agents with similar cognition can effectively collaborate to accomplish tasks. Cognition refers to an agent's understanding of its task, location, environment, and adopted strategy. Although tasks can be based on prior knowledge or learned by agents, this article focuses on cognition related to agents' positions and observations.

The objective of MADRQN training is not considered until it is developed. Therefore, we use mixed features, consisting of position and observation data, as the basis for agent clustering. These mixed features represent an agent's cognition in this context, and the clustering process groups agents with similar cognition to facilitate effective collaboration.

We employ the Growing Neural Gas (GNG) algorithm for agent clustering. GNG is a dynamic, self-organizing algorithm that adapts its network nodes and connections using competitive Hebbian learning (CHL) rules. It does not require specifying the number of categories in advance, continuously updating the network structure to learn the topological distribution of input vectors. Since the transportation network can be represented as a dynamically changing graph structure, agents are re-clustered at several time steps using GNG.

Clustering agents based on their location and traffic flow observations offers several advantages. Firstly, it enables agents within a cluster to access more comprehensive environmental information, mitigating non-convergence issues caused by dynamic environmental changes. This process ensures that agents with similar positions and observations collaborate effectively, leading to more efficient training and decision-making.

Once clustering is completed, information sharing and centralized training are conducted within each cluster. This approach reduces the amount of shared data and enhances collaboration between agents. The agent clustering algorithm is detailed in rows 5 to 14 of Algorithm 1.

4 Centralized Training of MADRQN with Observation Sharing and Optimal Parameters Moving

During the centralized training stage, as depicted in the left part of Fig. 1, at time step t , all agents within the cluster receive the global state s_t and the hidden state h_{t-1} from the previous time step. Using gated recurrent unit (GRU), temporal features of traffic flow are extracted to represent the state-action value $Q^i(\tau^i, a_t^i)$ for each individual agent. Then, each agent adopts an ϵ -greedy strategy to select its action a_t^i . Since all agents within the cluster have similar positions and observations, their contributions are considered equal during the centralized training process.

Multiple clusters can be trained in parallel, thereby accelerating the training process for multiple agents. The global state s_t in Eq. (1) represents a joint observation, which comprises observations from all agents within the same cluster. Similarly, the joint action a_t in Eq. (2) represents the actions of all agents in the cluster. By utilizing observations from other agents, effective information sharing is achieved, addressing issues of partial observable and non-stationary conditions prevalent in multi-intersection traffic signal control.

Furthermore, the significant reduction in the number of internal clusters compared to the total number of initial agents allows the decentralized execution stage to adopt shared observations without concerns about amounts of data. This streamlined communication process ensures efficient coordination among agents during the execution phase, further enhancing the collaborative performance of the overall traffic signal control system.

The framework of the MADRQN algorithm is depicted in Fig. 1. The various components and processes are detailed as follows:

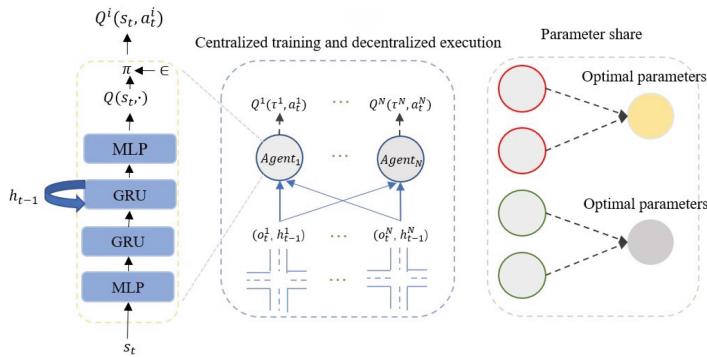


Fig. 1. MADRQN Algorithm framework

Prediction Process and Q-Function: The left portion illustrates the Q-function prediction process for a single agent, utilizing the global state s_t and the previous hidden state h_{t-1} .

Centralized Training and Decentralized Execution: The middle section outlines the central framework for training the agents in a cluster using a centralized approach while enabling decentralized execution.

Parameter Optimization within Clusters: The right part visualizes how parameters evolve toward the optimal values within each cluster.

State, Action and Reward Representation: The state s_t at time t is defined as the collection of observations from all agents within the cluster as

$$s_t = \{o_t^1, \dots, o_t^N\} \quad (1)$$

The action a_t at time t is represented as

$$a_t = \{a_t^1, \dots, a_t^N\} \quad (2)$$

The reward r_t is set to the difference between the average waiting time of all vehicles at the intersection at time t and the previous time as shown in Eq. (3).

$$r_t = \{r_t^1, \dots, r_t^N\} \quad (3)$$

Agent Loss Function: For the i th agent, the loss function $L(\theta^i)$ is formulated as the mean squared error between the estimated Q-values and target Q-values as shown in Eq. (4).

$$L(\theta^i) = \frac{1}{M} \sum_{j=1}^M (y_j - Q^j(s_j, a_j))^2 \quad (4)$$

where M is the batch size, and the training samples consist of tuples $\{s_t, a_t, s_{t+1}, r_t\}$.

Target Q-Value Estimation: As shown in Eq. (5), the target Q-value y_j is calculated based on the immediate reward r_j and the maximum Q-value of the target network Q_j^- for the next state-action pair.

$$y_j = r_j + \gamma \cdot \max_{a'} Q_j^-(s', a') \quad (5)$$

Centralized Training: During centralized training, the TD error is back propagated to update the parameters of the critic network. The gradient of the loss with respect to parameters ($\nabla_{\theta^i} L(\theta^i)$) is calculated as shown in Eq. (6), and the parameters are updated using a learning rate (α) as shown in Eq. (7).

$$\nabla_{\theta^i} L(\theta^i) = \frac{1}{M} \sum_{j=1}^M \left[-2(y_j - Q^j(s_j, a_j)) \cdot \frac{\partial Q^j(s_j, a_j)}{\partial \theta^i} \right] \quad (6)$$

$$\theta^i \leftarrow \theta^i - \alpha \cdot \nabla_{\theta^i} L(\theta^i) \quad (7)$$

Decentralized Execution: During decentralized execution, each agent employs its learned strategy π^{θ^i} to generate actions based on observations from all agents and the previous hidden state as shown in Eq. (8).

$$a_t^i = \pi^{\theta^i}(o_t^1, \dots, o_t^N, h_{t-1}^1, \dots, h_{t-1}^i) = \arg \max_{a_t^i \in A} Q_t^i(s_t, a_t^i) \quad (8)$$

Parameter Optimization via Particle Swarm Optimization (PSO): Inspired by the particle swarm optimization (PSO) algorithm, a method is proposed to accelerate training by transferring parameters among agents. Agents are viewed as particles exploring the parameter space. At the end of each episode, agent parameters move towards global and individual optima as

$$v_{\theta^i} = v_{\theta^i} + c_1 \cdot (pbest_{\theta^i} - \theta^i) + c_2 \cdot (gbest_{\theta} - \theta^i) \quad (9)$$

$$\theta^i \leftarrow \theta^i + v_{\theta^i} \quad (10)$$

Global and Individual Optimal Parameters: The global optimal parameters ($gbest_{\theta}$) are determined by Eq. (11), and parameters are updated accordingly.

$$gbest_{\theta} = \arg \max_{\theta^i} Q^i \quad (11)$$

This framework represents the MADRQN algorithm. The pseudocode is provided in Algorithm 1, where parallelism can significantly speed up the training process and enhance efficiency.

5 Experimental Environment and Parameter Design

As shown Fig. 2, the road network in Yijiang District, Wuhu City, has been imported into the SUMO simulation environment and it consists of several two-lane roads with 15 intersections, each represented and numbered as points.

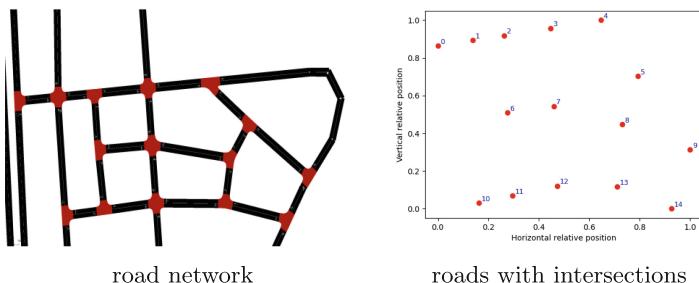


Fig. 2. Road network topology and crossing number

To model traffic signal control as a Markov Decision Process (MDP), we consider the six tuples: $\langle S, A, R, S', P, \gamma \rangle$. Here, S represents the agent's observation of traffic conditions at the intersection. The observation includes information about the vehicles in the four lanes of the intersection. Each lane is 400 m long, and each vehicle is 5 meters in size. As shown in Fig. 3, The observation vector indicates the presence, denoted as 1, or absence, denoted as 0, of vehicles at specific positions within the lanes. These vectors from the four lanes are combined to form the final observation vector.

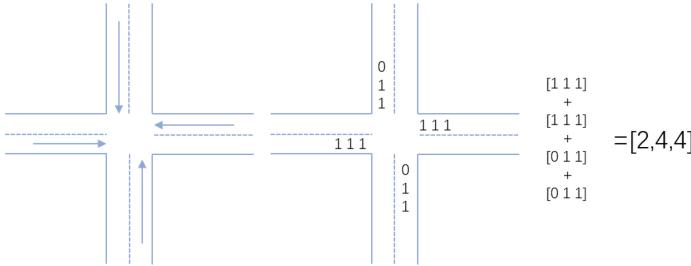


Fig. 3. State definition of crossing

The action space, denoted as A , includes four actions: north-south straight ahead, north-south left turn, east-west straight ahead, and east-west left turn. For standard intersections, these actions correspond to green lights for straight and left turns on both north-south and east-west roads, with all other directions showing red lights. At T junctions, there are two actions: one road has green lights for both straight and left turns, while another road has a left turn green light, and the remaining routes display red lights. In all configurations, the right-most lane is designated for right turns. The duration for each action, or green light phase, is set to 10 s, with a 4-second yellow light period between phases. The reward function R is defined as the difference in the average waiting time of all vehicles at the intersection between two consecutive time steps. Since the goal is to minimize waiting time, the reward is a negative number. Additional performance metrics include the average queue length of congested vehicles and the cumulative waiting time, which help evaluate the algorithm's effectiveness.

The state transition probability, denoted as P , is determined by the environment and is thus unknown. The discounted factor γ for immediate rewards is set to 0.9. By modeling traffic signal control as an MDP problem and considering the aforementioned six components $\langle S, A, R, S', P, \gamma \rangle$, the MADRQN algorithm can effectively learn to control traffic signals at multiple intersections, aiming to reduce waiting times and alleviate traffic congestion.

In the experiment, the traffic flow across the entire road network was set to 1,000 vehicles per second, randomly distributed among all intersections. The aim of this experiment is to demonstrate the effectiveness of our proposed method. To this end, the traffic flow at each intersection at specific time intervals was captured and recorded in Table 1. This data provides a detailed snapshot of the traffic conditions at each intersection during the experiment.

Algorithm 1. Cluster-based MADRQN with observation sharing and optimal parameter moving

Initialization: MDRQN network parameters θ^i , target network θ^{-i} , buffer D , episodes $N_e = 100$, steps $T = 300$, agents $N = 15$, agent positions p^i , clusters $N_c = N$, agents per cluster $N_A = N$, cluster interval $C = 1000$, cluster iterations $K = 2500$, node movement coefficients $\varepsilon_b = 0.1$, $\varepsilon_n = 0.1$, max edge age $AM = 30$, error decay $\beta = 0.9$, parameter shift $c_1 = 0.9$, $c_2 = 0.3$, discount $\gamma = 0.9$, learning rate $\alpha = 5e - 4$

- 1: **for** $ep = 1$ to N_e **do**
- 2: At $t = 0$, get observation o_0^i and position p^i for all agents $i = 1, \dots, N$
- 3: Initialize rewards $r_0^i = 0$, actions $a_0^i = 0$, greedy coefficient $\epsilon = (1 - \frac{ep}{M})^2$
- 4: **while** $t < T$ **do**
- 5: **if** $t \% C = 0$ **then**
- 6: Initialize GNG with vectors v_a, v_b from $\{(p^i, o_t^i)\}$
- 7: **for** $k = 0$ to K **do**
- 8: Choose vector x from $\{(p^i, o_t^i)\}$, find closest nodes s_1, s_2
- 9: Move s_1 and neighbors $s_{\mathcal{N}_{s_1}}$ towards x
- 10: $s_1 \leftarrow s_1 + \varepsilon_b \|x - v_{s_1}\|_2^2$
- 11: $s_{\mathcal{N}_{s_1}} \leftarrow s_{\mathcal{N}_{s_1}} + \varepsilon_n \|x - v_{\mathcal{N}_{s_1}}\|_2^2$
- 12: Update edges and ages, remove old edges and isolated nodes
- 13: Insert new node r between nodes q and p with largest errors
- 14: Update errors for nodes p, q , and new node r
- 15: **end for**
- 16: **end if**
- 17: **for** cluster $i_c = 1$ to N_c **do**
- 18: **for** agent $i_a = 1$ to N_A **do**
- 19: Receive observation $o_t^{i_a}$, choose action $a_t^{i_a}$, execute, get reward $r_t^{i_a}$, next observation $o_{t+1}^{i_a}$
- 20: **end for**
- 21: Store $(o_t^1, \dots, o_t^{N_A}, a_t^1, \dots, a_t^{N_A}, r_t^1, \dots, r_t^{N_A}, o_{t+1}^1, \dots, o_{t+1}^{N_A})$ in D
- 22: **end for**
- 23: **end while**
- 24: **for** cluster $i_c = 1$ to N_c **do**
- 25: **for** agent $i_a = 1$ to N_A **do**
- 26: **for** batch in D **do**
- 27: Share observations $s_t = \{o_t^1, \dots, o_t^{N_A}\}$, $s_{t+1} = \{o_{t+1}^1, \dots, o_{t+1}^{N_A}\}$
- 28: Calculate loss and update parameters
- 29: **end for**
- 30: **end for**
- 31: Move parameters towards optimal according to Eq. (9) to (11)
- 32: **end for**
- 33: Clear buffer D
- 34: **end for**

Table 1. Traffic flow setup at all crossings

Intersection Number	0	1	2	3	4	5	6	7	8	9	19	11	12	13	14
Traffic flow at intersections (vehicles/second)	57	29	25	27	34	96	48	79	61	72	21	92	42	62	22

6 Experimental Results and Analysis

In this section, we conduct experiments and analyze the results of various methods, including agent clustering, centralized training based on observation sharing, and parameter transfer, as detailed in Algorithm 1.

(1) Clustering agents based on location and observation

Figure 4 illustrates the agent clustering process using the GNG algorithm. The first figure depicts the GNG network topology formed during clustering, The blue dots indicate each intersection and the red dots indicate the dynamically generated network topology by GNG, while the third figure shows the final clustering results and the relationships among agents. In the second figure, clustering results are shown for a traffic flow of 1 vehicle per second at the entry lanes of intersections 5, 6, and 7. The agents are clustered into three groups: blue dots $\{0, 1, 2, 3, 4\}$, red dots $\{5, 6, 7, 8\}$, and green dots $\{9, 10, 11, 12, 13, 14\}$. The fourth figure presents clustering results for a traffic flow of 2 vehicles per second at the entry lanes of intersections 0 and 14, with agents clustered into three groups: green dots $\{0, 1, 2, 3, 4, 5, 6, 7\}$, red dots $\{10, 12, 13\}$, and blue dots $\{8, 9, 11, 14\}$.

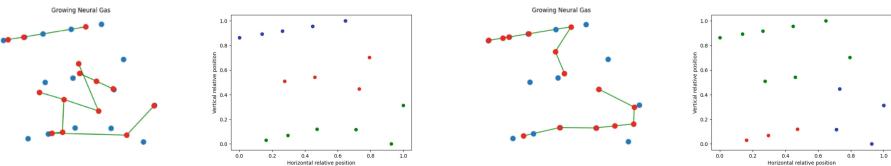


Fig. 4. Median network topology and results of agent cluster

Figure 5 displays the clustering results based on parameters set in Tables 1 and 2. The left part of the figure shows the GNG network topology during the intermediate clustering process, while the right part presents the final clustering results. The agents are divided into four groups: red dots $\{0, 1, 2, 3, 4\}$, green dots $\{6, 7, 10, 12, 13\}$, blue dots $\{5, 8\}$, and deep blue dots $\{9, 11, 14\}$.

The clustering results are primarily influenced by the location of the intersections and traffic flow. Even with similar traffic flows, agents may not be clustered together if they are geographically distant. As traffic conditions constantly change, the clustering results may also vary over time. Figure 5 shows the clustering outcomes from the first round of training using Algorithm 1.

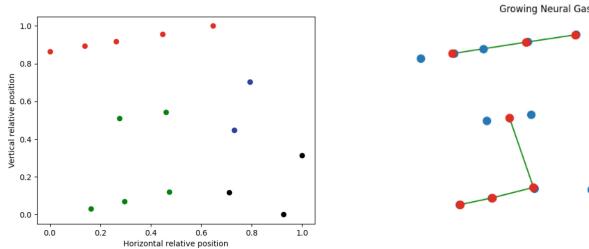
GNG can adaptively generate optimal clusters based on the number and location of intersections. These clusters enable more efficient and effective traffic signal control strategies within the MADRQN algorithm.

(2) The Overall Control Effect and Analysis of MADRQN Algorithm

Figure 6 compares the proposed MADRQN algorithm, which utilizes information and parameter sharing, with other MADRL algorithms such as QMIX,

Table 2. Experiment and algorithm parameter setup

Intersection Number	Value (range)
Parameter	400 m into each intersection lane
Number of intelligent agents N	15
Vehicle length	5 m
Minimum vehicle gap	1~2 m
Maximum speed	40~50km/h
Vehicle acceleration	0.5~1 m/s ²
Vehicle deceleration	-3.5 ~ -5 m/s ²
Training episode M	100
Each episode max step T	5400
Learn rate α	5e-4
Discount factor γ	0.9
Action selection greedy parameter ϵ	Initially decreasing from 1 to 0.001
Replay buffer size D	5000

**Fig. 5.** Agent cluster results with parameters of Table 1 and Table 2

VDN, and DRQN. The comparison is based on the average reward for each round of traffic signal control across 15 intersections.

In Fig. 6, the horizontal axis represents the number of training rounds (totaling 100), and the vertical axis represents the average reward per round. The average reward is calculated by summing the negative rewards for all steps in a round, resulting in a cumulative negative reward. Each algorithm is executed three times, with each run consisting of 100 rounds and each round comprising 300 steps. The cumulative negative rewards from the three runs are averaged to obtain the average reward. This multi-run approach allows for observing the stability and performance of the algorithms over multiple iterations.

To analyze algorithm stability, the standard deviation of the average rewards is calculated. The upper and lower colored bands around the curve in Fig. 6 represent this standard deviation, indicating the extent of reward deviation from the average value across the three runs. The average reward calculation is computed over three runs.

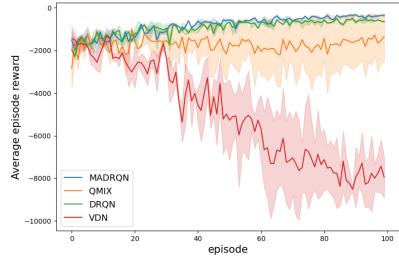


Fig. 6. Average episode cumulative negative reward comparison of algorithms.

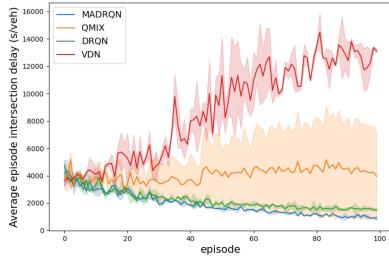


Fig. 7. Average episode cumulative delay comparison of algorithms.

By comparing the average rewards and their standard deviations across different algorithms and training rounds, the paper evaluates the performance and stability of the MADRQN algorithm against other MADRL algorithms. The results in Fig. 6 demonstrate the effectiveness and robustness of the proposed MADRQN algorithm for traffic signal control at multiple intersections.

Figure 7 compares the average cumulative delay per round for different algorithms. The horizontal axis represents the number of training rounds, totaling 100, while the vertical axis represents the average cumulative delay per round. The average cumulative delay is calculated by summing the average delays of all steps in a single round of traffic signal control to obtain the cumulative delay. Each algorithm is run three times, with each run comprising 100 rounds, and each round having $T_i = 300$ steps. The cumulative delay is then averaged over the three runs. From Fig. 7, the following observations can be made:

- **MADRQN vs. VDN:** MADRQN slightly outperforms VDN, benefiting from its Recurrent Neural Network (RNN) architecture that captures temporal features for more accurate Q-value predictions.
- **MADRQN vs. DRQN:** MADRQN significantly reduces average cumulative delay compared to DRQN. This is due to MADRQN’s agent clustering and observation sharing, which improves handling of partially observable and non-stationary environments through enhanced agent collaboration.
- **MADRQN vs. QMIX:** MADRQN also surpasses QMIX, which relies on parameterized hyper networks and global states for value function decomposition. MADRQN’s use of agent clustering and observation sharing results in a more uniform environment and better information sharing, improving overall traffic control performance.

Overall, Fig. 7 demonstrates that MADRQN, with its advanced agent clustering and observation sharing features, effectively reduces average cumulative delay and outperforms other MADRL algorithms, showcasing its potential for improving multi-intersection traffic signal control.

(3) The Influence and Analysis of Agent Clustering on Control Effectiveness

The paper introduces the clustering of agents for two primary purposes:

- **Similar Observations and Positions:** Clustering groups agents with comparable observations and positions, enhancing cooperation within clusters. This shared understanding of the environment facilitates more effective traffic signal control.
- **Reduced Data Transmission:** Clustering decreases data transmission needs by consolidating communication within clusters rather than among all agents. This reduction in data exchange is critical for decentralized execution, reducing communication delays.

To verify this hypothesis, we compared the MADRQN algorithm with and without agent clustering using the Growing Neural Gas (GNG) algorithm. Figures 8 and 9 demonstrate that MADRQN with clustering (blue curve) significantly outperforms the non-clustered version (orange curve) in both cumulative negative reward and average delay per round, confirming the effectiveness of clustering.

Figure 10 reveals that clustering also shortens training time, as the smaller data sets within each cluster allow for more efficient parallel processing.

In summary, agent clustering enhances cooperation, minimizes data transmission, and boosts the efficiency and effectiveness of the MADRQN algorithm for multi-agent traffic signal control.

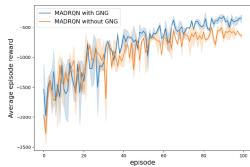


Fig. 8. Average episode cumulative negative reward comparison of MADRQN with/without cluster agent

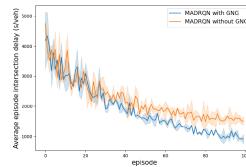


Fig. 9. Average episode cumulative delay comparison of MADRQN with/without cluster agent

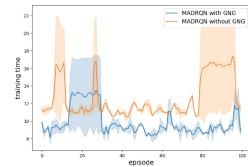


Fig. 10. Average episode training time comparison of MADRQN with/without cluster agent

Figures 11 and 12 further explore the impact of parameter sharing. Figure 11 shows that parameter sharing reduces training time per round, speeding up convergence. However, Fig. 12 indicates a trade-off, where optimal parameter sharing can lead to reduced cumulative rewards. This suggests that while parameter sharing accelerates training, it may result in locally optimal solutions. Future work should investigate alternative parameter sharing strategies to balance convergence speed and overall performance.

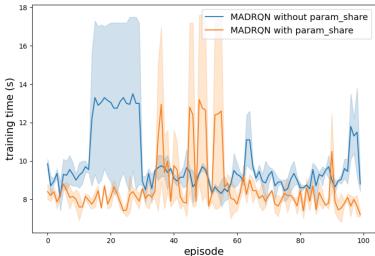


Fig. 11. Average episode training time comparison of MADRQN with/without optimal parameter sharing

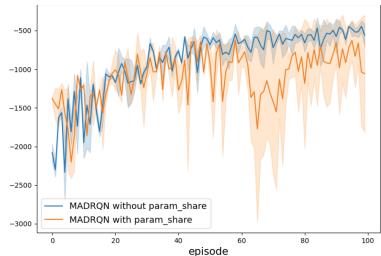


Fig. 12. Average episode cumulative reward comparison of MADRQN with/without optimal parameter sharing

7 Conclusion

This research underscores the importance of clustering agents based on location and observation in MADRL algorithms for traffic signal control. The proposed method of information sharing and centralized training outperforms traditional multi-agent reinforcement learning algorithms like QMIX and VDN by enhancing cooperation and addressing issues in observable and non-static environments.

Our approach enables effective cooperation among multiple agents, making it scalable for managing large, complex traffic networks, and highly applicable to smart city initiatives. Additionally, insights gained from clustering agents based on multiple factors can enrich the theoretical framework of multi-agent systems and be applied to other domains. Despite these advancements, there are areas for improvement including agent clustering features, algorithm structure, and information selection and representation.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China (No. 62306010) and Wuhu Municipal Science and Technology Bureau Project (Grant No. 2023jc17).

Disclosure of Interests. All authors declare that they have no competing financial interests or personal relationships that could influence the work reported in this paper.

References

1. Bálint, K., Tamás, T., Tamás, B.: Deep reinforcement learning based approach for traffic signal control. *Transp. Res. Procedia* **62**, 278–285 (2022)
2. Chiu, S.: Adaptive traffic signal control using fuzzy logic. In: Proceedings of the Intelligent Vehicles92 Symposium, pp. 98–107. IEEE (1992)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)

4. El-Tantawy, S., Abdulhai, B., Abdelgawad, H.: Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1140–1150 (2013)
5. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
6. Fritzke, B.: A growing neural gas network learns topologies. *Adv. Neural Inf. Process. Syst.* **7** (1994)
7. Genders, W., Razavi, S.: Using a deep reinforcement learning agent for traffic signal control. arXiv preprint [arXiv:1611.01142](https://arxiv.org/abs/1611.01142) (2016)
8. Haddad, T.A., Hedjazi, D., Aouag, S.: A new deep reinforcement learning-based adaptive traffic light control approach for isolated intersection. In: 2022 5th International Symposium on Informatics and its Applications (ISIA), pp. 1–6. IEEE (2022)
9. Huang, J., Tan, Q., Qi, R., Li, H.: RELight: a random ensemble reinforcement learning based method for traffic light control. *Appl. Intell.* **54**(1), 95–112 (2024)
10. Hunt, P., Robertson, D., Bretherton, R., Royle, M.C.: The scoot on-line traffic signal optimisation technique. *Traffic Eng. Control* **23**(4) (1982)
11. Jiang, J., Lu, Z.: Learning attentional communication for multi-agent cooperation. *Adv. Neural Inf. Process. Syst.* **31** (2018)
12. Kang, L., Huang, H., Lu, W., Liu, L.: A dueling deep q-network method for low-carbon traffic signal control. *Appl. Soft Comput.* **141**, 110304 (2023)
13. Liu, W., Qin, G., He, Y., Jiang, F.: Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering. *IEEE Trans. Veh. Technol.* **66**(10), 8667–8681 (2017)
14. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **30** (2017)
15. Luk, J.: Two traffic-responsive area traffic control methods: scat and scoot. *Traffic Eng. Control* **25**(1) (1984)
16. Mingli, S., and HU Ming, C.E., Yue, Z., Wenjie, C., Mingsong, C., et al.: Traffic light optimization control method for priority vehicle awareness. *J. Softw.* **32**(8), 2425–2438 (2021)
17. Pandit, K., Ghosal, D., Zhang, H.M., Chuah, C.N.: Adaptive traffic signal control with vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **62**(4), 1459–1471 (2013)
18. Peng, P., et al.: Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. arXiv preprint [arXiv:1703.10069](https://arxiv.org/abs/1703.10069) (2017)
19. Prashanth, L., Bhatnagar, S.: Threshold tuning using stochastic optimization for graded signal control. *IEEE Trans. Veh. Technol.* **61**(9), 3865–3880 (2012)
20. Rashid, T., Samvelyan, M., De Witt, C.S., Farquhar, G., Foerster, J., Whiteson, S.: Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **21**(178), 1–51 (2020)
21. Richhariya, B., Tanveer, M., Initiative, A., et al.: Least squares projection twin support vector clustering (LSPT SVC). *Inf. Sci.* **533**, 1–23 (2020)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)

23. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 5887–5896. PMLR (2019)
24. Storn, R., Price, K.V., Lampinen, J.: Differential evolution-a practical approach to global optimization (2005)
25. Su, H., Zhong, Y.D., Dey, B., Chakraborty, A.: A decentralized reinforcement learning framework for efficient passage of emergency vehicles. arXiv preprint [arXiv:2111.00278](https://arxiv.org/abs/2111.00278) (2021)
26. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with back-propagation. *Adv. Neural Inf. Process. Syst.* **29** (2016)
27. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint [arXiv:1706.05296](https://arxiv.org/abs/1706.05296) (2017)
28. Tanveer, M., Tabish, M., Jangir, J.: Sparse pinball twin bounded support vector clustering. *IEEE Trans. Comput. Soc. Syst.* **9**(6), 1820–1829 (2021)
29. Tigga, A., Hota, L., Patel, S., Kumar, A.: A deep q-learning-based adaptive traffic light control system for urban safety. In: 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 2430–2435. IEEE (2022)
30. Tunc, I., Soylemez, M.T.: Fuzzy logic and deep q learning based control for traffic lights. *Alex. Eng. J.* **67**, 343–359 (2023)
31. Yan, L., et al.: Graph cooperation deep reinforcement learning for ecological urban traffic signal control. *Appl. Intell.* **53**(6), 6248–6265 (2023)
32. Yang, S., Yang, B., Zeng, Z., Kang, Z.: Causal inference multi-agent reinforcement learning for traffic signal control. *Inf. Fusion* **94**, 243–256 (2023)



A Hybrid Contextual Deep Learning Model to Predict Renewable Energy Generation

Deepak Kanneganti¹ Sajib Mistry¹ Sumedha Rajakaruna¹ Aneesh Krishna¹ and Amin Beheshti²

¹ School of Electrical Engineering, Computing and Mathematical Sciences,
Curtin University, Perth, Australia

s.kanneganti,sajib.mistry,s.rajakaruna,a.krishna}@curtin.edu.au

² School of Computing, Macquarie University, Sydney, Australia
amin.beheshti@mq.edu.au

Abstract. Renewable energy sources are widely adopted as they are safer for generating energy with less atmospheric harm. Solar power prediction depends on sunlight to forecast solar power effectively, and one of the significant challenges is weather dependency. Weather conditions, especially during overcast and rainy days, significantly impact power generation and can lead to infrastructure challenges and disruptions in the energy supply. We explore state-of-the-art deep learning algorithms to analyze contextual time series data, crucial for understanding the impact of weather uncertainty on power generation. To address this, we propose a novel deep learning-based hybrid contextual model built using Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Gated Recurrent Unit (GRU) networks. This model leverages big data collected over ten years (from 2013 to 2023) to account for the variability and uncertainty inherent in weather conditions. Experimental results demonstrate that the proposed hybrid contextual model exhibits a 4.23% performance improvement over the individual deep learning models in predicting power under weather uncertainty conditions.

Keywords: Renewable energy prediction · Long Short-Term Memory (LSTM) · Bidirectional Long Short-Term Memory (Bi-LSTM) · Gated Recurrent Unit (GRU) networks · Time series generator

1 Introduction

Climatic changes have become a global issue, primarily driven by the emission of greenhouse gases from fossil fuel energy generation. An increasing number of people are transitioning to renewable energy sources such as solar, wind, hydro, geothermal, and biomass. The International Energy Agency's 2019 report shows that solar energy comprises 60% of the energy market, and by 2025, renewable

energy is expected to account for 33% of global electricity [1]. The growing reliance on solar power has created an essential requirement for accurate solar forecasting.

Solar forecasting involves predicting the amount of energy the solar panels generate over time. Unlike other energy sources, such as fossil fuels and nuclear power plants, solar energy production is intermittent and relies solely on weather conditions. Studies show that solar power management operators, such as Transmission System Operators (TSOs) and Distribution System Operators (DSOs), require photovoltaic (PV) power forecasting to effectively manage grid operations and avoid service disruptions [2]. However, DSOs and TSOs require *accurate* solar forecasting to meet demand and prevent service disruptions due to the dependency on weather conditions such as cloud cover and rain [3].

Existing studies offer a variety of solutions for power forecasting models, primarily utilizing artificial intelligence (AI) based designs. These models include backpropagation (BP) neural network prediction models [4], Artificial Neural Networks (ANN) [5], Regularized-ELM (R-ELM) [6], and advanced deep neural network models such as Long Short Term Memory (LSTM) [7], Bidirectional Long Short Term Memory (Bi-LSTM) [8], and Gated Recurrent Units (GRU) [9]. However, these models require extensive training data, i.e., big data and computational resources, to address weather uncertainties, and they often need help to handle and process large volumes of data efficiently. This leads to limitations in scalability and performance when dealing with big datasets. Additionally, the complexity and high dimensionality of big data has the ability to overwhelm these models, resulting in *suboptimal predictions* and slower processing times.

We propose a *hybrid context model* for effective power predictions, trained using a real-world big dataset collected over ten years (from 2013 to 2023) at the Green Electric Energy Park in Curtin University. The improved model effectively analyses contextual changes present in weather patterns and provides the following improvements over existing models:

1. **Scalability and Efficiency:** The hybrid context model is designed to efficiently process large datasets, overcoming the limitations of traditional AI models that struggle with high-dimensional data. This makes it capable of handling the extensive bid data collected over ten years.
2. **Contextual Awareness:** The model incorporates mechanisms to understand and adapt to contextual changes in the data over time. This is crucial for accurately forecasting power in the face of evolving weather patterns and other dynamic factors that affect solar power generation.
3. **Improved Accuracy:** By leveraging historical and contextual data, the hybrid context model can provide more accurate power predictions. It considers the temporal dependencies and variations that occur over a long period, leading to more reliable forecasts.

Figure 1 illustrates the proposed solar forecasting and power management framework. Weather and solar data are collected and processed by the weather forecasting model to predict future conditions. The power forecasting model then

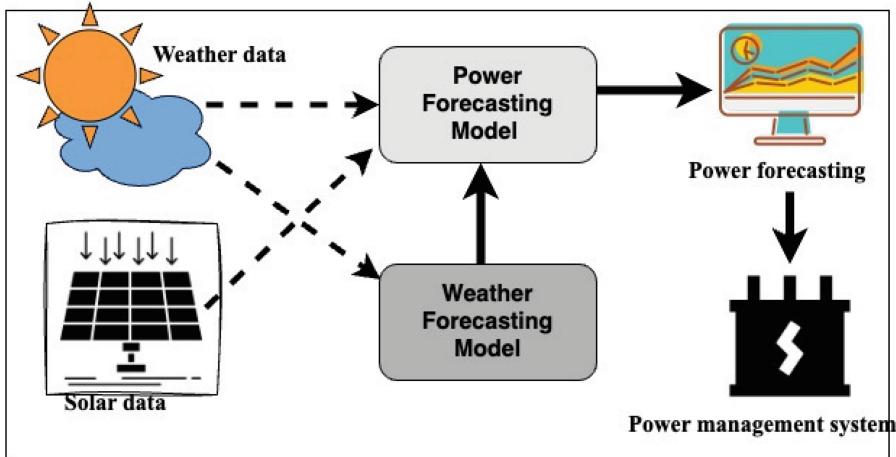


Fig. 1. Solar forecasting and power management system

uses these predictions and historical solar data to forecast solar power generation. This forecast is used by the power management system to optimize and align power generation with demand. The proposed hybrid contextual model integrates Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (Bi-LSTM), and Gated Recurrent Units (GRU) for improved power forecasting. LSTM is used for its ability to capture long-term dependencies in sequential data, making it suitable for handling time-series data with temporal dependencies. Bi-LSTM enhances this capability by processing data in both forward and backward directions, providing a more comprehensive understanding of the sequence. GRU, a variant of LSTM, offers a simplified architecture that reduces computational complexity while maintaining performance, making it efficient for large datasets. Combining these deep learning models, our hybrid approach leverages their strengths to handle contextual changes over a decade of data, resulting in highly accurate solar power predictions.

2 Related Work

This section presents an overview of related work on power forecasting models, highlighting their significance and interconnections. Power forecasting models are broadly categorized into three classes: Physical models, Statistical models, and AI-based approaches. Physical models, such as NAM and WRF, are primarily used for local area PV power prediction, while models like GFS are applied for wide-area solar forecasting [10]. Although these models perform accurately under various weather conditions, they often struggle with precise output characterization and predictions. Statistical models leverage large datasets to identify relationships between power output and various features. This category includes linear, multiple, and nonlinear regression models [11] and SVM [12]. Time series

models like ARIMA [13] and SARIMA are adept at handling complex patterns, with SARIMAX excelling in capturing seasonal and cyclic behaviours. Additionally, multivariate models such as VAR [14] and VECM and DFA demonstrate effectiveness in capturing intricate dynamics of solar power prediction [15].

Recent studies have increasingly focused on AI models to enhance the accuracy of solar power prediction. Artificial Neural Networks (ANN) employing solar radiation data, along with algorithms like Back Propagation and Radial Basis Function Network, have shown promising results [5]. Further studies introduced advanced models, such as Extreme Learning Machine (ELM) and Wavelet Neural Networks (WNN), for similar applications. The approaches to solar power forecasting are further categorized based on their forecasting horizon: short-term (less than three months) [16], mid-term (less than a year), and long-term (beyond two years) [17]. Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) [7], Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Units (GRU) [9], have demonstrated significant potential in solar power prediction. These RNN models, including LSTM and GRU, consistently outperform traditional models like ARIMA in forecasting accuracy due to their superior sequence learning and long-term dependency capabilities [18]. However, LSTM models are known for their complexity and high computational cost, which GRU models address by offering faster computation times, albeit with slightly reduced accuracy. The introduction of bidirectional LSTM and GRU models has further improved performance, especially in environments with high data variability, such as the Arctic.

Furthermore, hybrid models that combine LSTM with other techniques have been explored to enhance prediction accuracy. For example, Ju and Thu developed a hybrid LSTM-AR model, which showed promising results on real-time data, though its effectiveness varies across different time series datasets [19]. Similarly, Lim's study integrated Convolutional Neural Networks (CNN) with LSTM, achieving promising outcomes for solar power prediction despite requiring more training and computational resources than standard RNN models [20]. Additionally, a hybrid model incorporating a CNN with Bi-LSTM has shown superior prediction capabilities. However, it faces challenges such as increased complexity, computation time, and identifying optimal CNN parameters. We propose a novel hybrid model combining the LSTM, GRU, and Bi-LSTM models. Our model, trained on a *big dataset of 5,356,800 records* from March 2013 to March 2023, captures intricate patterns and dependencies in real-world solar power data. We conducted a comparative analysis of LSTM, Bi-LSTM, and GRU models and explored hybrid configurations. This approach highlights the potential of hybrid models in handling big data and optimizing solar power prediction.

3 A Hybrid Contextual Model for Solar Power Prediction

This section introduces our proposed hybrid contextual model for solar power prediction. Fig. 2 illustrates the overview of the proposed framework. First, we present the architecture of the proposed hybrid contextual model, employing

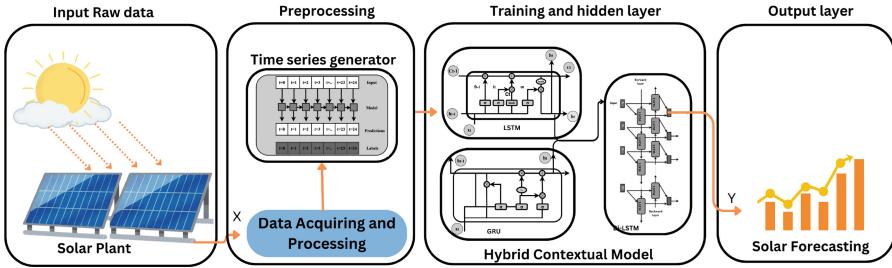


Fig. 2. Hybrid Contextual Model for Solar Power Prediction

cutting-edge deep learning techniques such as LSTM, Bi-LSTM, and GRU. The second stage focuses on data acquisition and preprocessing. The data preprocessing pipeline involves several stages such as data cleaning, prepossessing, feature scaling, and feature engineering. A timing generator is created to generate different time stamps to generate training instances for the proposed model. These stages work together to transform the data and prepare it for use by the model.

3.1 Architecture of the Hybrid Contextual Model

The hybrid contextual model for solar power prediction integrates multiple deep learning architectures, including LSTM, GRU, and Bi-LSTM networks, to enhance prediction accuracy by leveraging various contextual features. Initially, a time series generator (T) is initialized to create and manage time sequences for the input data (Algorithm 1 Line 3). The model generates a series, selecting an appropriate sequence length (L) and batch size (B). Exploratory Data Analysis (EDA) techniques are applied to transform the input features (N) into a suitable format (Algorithm 1 Line 5), and these transformed features are then flattened to prepare them for input into the LSTM and GRU networks (Algorithm 1 Line 6). The LSTM model takes the input vector $(h(t - 1), x(t))$ at time step t , and the output number will be 0 or 1 at each cell state (C_{t-1}). The Sigma layer is the decision-making layer, called a forget gate layer (f_t) (Algorithm 1 Line 7). The input gate ($i(t)$) layer decides to update the information. Moreover, the tanh layer creates a new vector for the new cell state C' . An updated state is produced using (f_t) . The update of the cell state involves a series of steps where the previous cell state $C(t - 1)$ is modified to produce the new cell state C_t . Initially, the previous state f_t is multiplied by $C(t - 1)$ and added to the new candidate value. The new candidate value is computed by combining the results of f_t and C_t , as shown in $f_t * C(t - 1)$. Finally, the output depends on the filtered cell state. In the initial stages, the sigmoid layer determines the contribution to the cell state, which then passes through the tanh layer, yielding an output that ranges from -1 to 1. The sigmoid gate ultimately generates the necessary output for the required part. In the LSTM network, the forget gate (f_t) is computed using the input (x_t) and the previous hidden state (h_{t-1}) (Algorithm 1 Line number 7). The hidden state (h_t) is updated accordingly (Algorithm 1 Line number 8), and

this hidden state is then passed through a fully connected layer to generate an intermediate output (s_t) (Algorithm 1 Line number 9) (Fig. 3).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), C_t = (f_t * C_{t-1} - 1) + (i_t * \hat{C}_t) \quad (2)$$

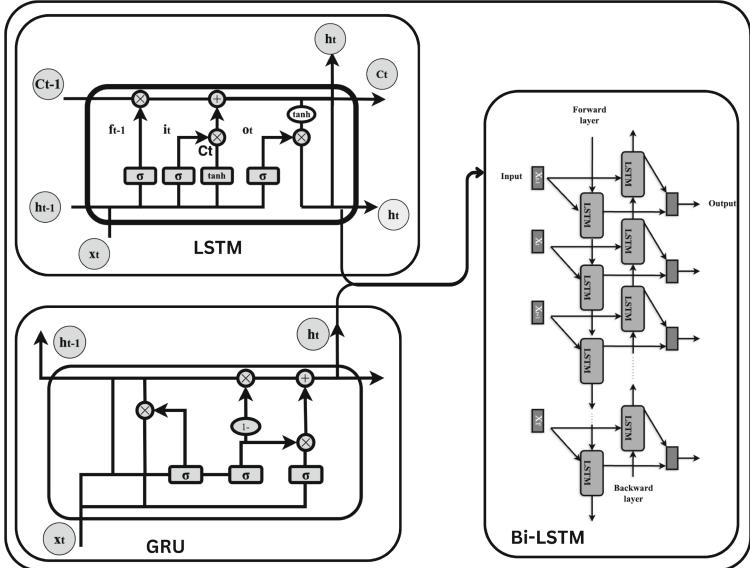


Fig. 3. Architecture of hybrid Contextual model

The Gated Recurrent Unit (GRU) mitigates the vanishing gradient problem found in standard RNNs by employing update and reset gates. The update gate, Z_t , combines the input x_t and the previous hidden state h_{t-1} , weighted by W_t and U_t , respectively, and processed through a sigmoid function. This gate determines the contribution of the previous state to the current state(Algorithm 1 Line 11–13).

$$Z_t = \sigma(W_t * x_t + U_t * h_{t-1}), r_t = \sigma(W_r * x_t + U_r * h_{t-1}) \quad (3)$$

$$\hat{h} = \tanh(Wx_t + r_t * U_{ht-1}), h_t = (Z_t * h_{t-1} + (1 - Z_t)) * \hat{h}_t \quad (4)$$

The reset gate, r_t , modulates the influence of past information similarly, using weights W_r and U_r with a sigmoid activation. The candidate activation, \hat{h}_t , is computed by scaling the previous state with the reset gate's output, combining it with the current input, and applying a hyperbolic tangent activation. The final hidden state, h_t , results from an element-wise multiplication of the update gate's

output with the previous state and the candidate activation scaled by $1 - Z_t$. This ensures a smooth transition between past and present information, effectively managing long-term dependencies and addressing the vanishing gradient issue in sequential data processing (Algorithm 1 Line 11–13). This mechanism enables the GRU to manage long-term dependencies effectively. The outputs from the LSTM and GRU networks are combined to produce the final power prediction (P). The model also incorporates a Bi-LSTM network. The Bi-LSTM is initialized, and the input features are flattened (Algorithm1 Lines 14–16).

Algorithm 1. Contextual model based solar power prediction

Input: $\mathbf{X}(\mathbf{C}, \mathbf{H}, \mathbf{R}, \mathbf{B}, \mathbf{S}, \mathbf{D}, \mathbf{T})$

Output: Predicted Power

```

1: procedure CONTEXTUAL-MODEL( $x, C, d, M$ , Contextual model)
2:   Output: Solar Power  $P$ 
3:   Initialize: Time Series Generator ( $T$ )
4:   Generate a gait series and select the length (L), Batch size (B)
5:   Apply EDA technique to transform features (N)
6:   Flatten the features for the LSTM, GRU network
7:    $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$                                 ▷ LSTM Input
8:    $h_t = o_t \cdot \tanh(C_t)$                                          ▷ LSTM Output
9:    $s_t = h_t(y_t)$                                               ▷ Fully Connected Layer
10:   $\bar{y}_{t+1} + b_z$                                               ▷ LSTM Input
11:   $Z_t = \sigma(W_t \cdot x_t + U_t \cdot h_{t-1})$                          ▷ GRU Update Gate
12:   $h_t = (Z_t \cdot h_{t-1} + (1 - Z_t)) \cdot \hat{h}_t$                    ▷ GRU Hidden State
13:   $s_t = h_t(y_t)$                                               ▷ Fully Connected Layer
14:   $P = \{P_L, P_G\}$                                          ▷ Power Output
15:  Initialize: Time Series Generator ( $T$ )
16:  Flatten the Power features of the LSTM, GRU network
17:   $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$                                 ▷ LSTM Input
18:   $h_t = o_t \cdot \tanh(C_t)$                                          ▷ LSTM Output
19:   $s_t = h_t(y_t)$                                               ▷ Fully Connected Layer
20:   $\bar{y}_{t+1} + b_z$                                               ▷ Bi-LSTM Input
21:   $\overset{\rightarrow}{h}_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_t)$                       ▷ F-Bi-LSTM
22:   $\overset{\leftarrow}{h}_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_t)$                       ▷ B-Bi-LSTM
23:   $H_t = \sigma(W_f + W_b) + b_y$                                          ▷ Bi-LSTM Hidden Layer
24:   $h_t = o_t \odot \tanh(c_t)$                                          ▷ Bi-LSTM Output
25:   $s_t = h_t(y_t)$                                               ▷ Fully Connected Layer
26: end procedure

```

The forget gate (f_t) in the Bi-LSTM is computed, followed by the calculation of the hidden state (h_t). This hidden state is passed through a fully connected layer to generate an intermediate output (s_t) (Algorithm1 Line 17–19). The forward and backward passes of the Bi-LSTM network are performed (Algorithm 1 Lines 21–22), and the Bi-LSTM hidden layer (H_t) is computed. The final Bi-LSTM output (h_t) is then obtained (Algorithm 1 Line 24) and passed through

another fully connected layer to produce the final output (s_t) (Algorithm 1 Line 22–25). By processing the sequence of sensor values and time through these networks, the hybrid contextual model effectively learns to predict solar power output based on both current and historical information, capturing temporal dependencies and trends in the data. This approach leverages the strengths of each network type to improve the overall prediction accuracy.

3.2 Data Acquisition and Pre-processing

The hybrid model is trained on seven years of consecutive data and evaluated at various intervals. In the data pre-processing stage, a large volume of historical data is extracted from the power station in text file format. These data files underwent two stages of data cleaning, which are discussed further in this analysis. Initially, we identified the percentage of missing values, outliers, and duplicates. The data contains offline periods and substantial anomalies with the intermittent solar power storage in inverters. We develop functions to detect and handle these contextual outliers. The weather data may remain the same sometimes, resulting in duplicate values being placed and removed from the data. Duplicate weather data entries are also identified and removed to maintain data quality. Sensor data points are normalized using the min-max technique to standardize the ranges of different features. The min-max normalization is applied where x is the original value, x_{\min} and x_{\max} are the minimum and maximum values of the feature, respectively, and x' is the normalized value. Feature engineering techniques are applied to derive new insights from the raw data. For instance, wind speed and direction are used to create wind vector components x and y . Additionally, historical weather data has numerous attributes. We employ feature selection techniques to pinpoint the elements that most influence power prediction, thereby improving the model's accuracy and efficiency.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (5)$$

3.3 Training the Hybrid Contextual Model

Time Series Generator. A time series generator is essential to maintain the sequential relationship between observations in our study. Recurrent neural networks (RNNs), employed here, require a mechanism to prepare time series data that organizes sequences, handles variable-length sequences, and preserves the necessary temporal relationships. Each timestamp t includes both input and target values. For example, time steps t_0 to t_7 are used as inputs, while t_8 serves as the target. The tables on the left side of the image illustrate the progression of input and target values at each time step. In this study, each time step encompasses 12 h of data as input to predict the subsequent hour. Parameters for each time step include a batch size of 32, a sample rate of 1, and 12 features. This data structuring enables the model to learn weather pattern changes effectively, leading to improved predictions. The model's performance is evaluated

by comparing the predicted output values at each time step t_0 with the actual labels. This approach ensures the model accurately captures temporal dynamics, enhancing its predictive capability. Tuning parameters through trial and error is time-consuming and error-prone. To improve this process, random and grid search methods are employed. Table 1 illustrates the parameters used to design the models, including layer structure, learning rate, dropout rate, alpha, loss function, and epochs. TensorFlow is used to build deep neural networks for solar prediction. The Recurrent Neural Networks (RNNs) consist of three RNN layers and three fully connected layers, with dropout layers to prevent overfitting.

Table 1. Hybrid Contextual Model parameters

Hyper-parameters	LSTM	Bi-LSTM	LSTM-GRU-BiLSTM
Layer	512-256-128-64	512-256-128-64	(64-32-32)
Activation function	Leaky ReLU	Leaky ReLU	Leaky ReLU
Alpha	0.5	0.5	0.5
Learning rate	0.001	0.001	0.001
Dropout	0.3	0.3	0.3
Loss function	MSE	MSE	MSE
Training epochs	30	30	20
L2 Regularization	1e-15-1e-2	1e-15-1e-2	1e-15-1e-2

Hyper-Parameter Tuning A dropout rate of 0.3 is used for medium-sized models. The RNN architecture has hidden layers with 512, 256, and 128 neurons. Mean Squared Error (MSE) is the loss function and the Adam optimizer with a learning rate of 0.001 updates the weights. The Leaky ReLU activation function, with an alpha value of 0.5, prevents the dying ReLU problem. The models, LSTM and GRU, are trained for 30 epochs. The hybrid model uses outputs from the best two models for final power predictions. LSTM and GRU models excel in predicting higher and lower power with longer datasets. The third column in the tables details the parameters for the Bi-LSTM model. The Bi-LSTM model combines outputs from the top two models. The Bi-LSTM model has hidden layers with neurons sized 64-32-32 and uses a Leaky ReLU activation function with an alpha of 0.5 to avoid the dying rule problem.

4 Experiments and Results

Experiments are conducted to assess the performance of the proposed hybrid contextual model with traditional deep learning models. First, we compare the proposed model predictions with the traditional recurrent neural networks like LSTM, BiLSTM and GRU, considering the evaluation metrics like R2 and

MAPE score. Second, we evaluated the efficiency of the hybrid contextual model efficacy in predicting the time power at different time intervals. All the experiments are conducted on Pawsey supercomputer with Intel Core i7 CPU (3.3 GHz and 16 GB RAM). The corresponding code and data can be accessed in our repository¹.

4.1 Dataset

Ten years of the solar grid and weather data are extracted from the *Green Electric Energy Park in Bentley, Western Australia* (GEEP)². Table 2 provides a detailed description of the features and their respective units used in the model development. The independent features in this analysis include temperature ($^{\circ}\text{C}$), humidity (g/m^3), solar radiation (W/m^2), barometric pressure (Hg), wind speed (m/s), and wind direction (degree). Additionally, the dataset includes solar grid temperature as a key feature. Weather and solar power measurements are recorded every minute on a monthly basis. The model training utilized a substantial dataset comprising 5,356,800 records, spanning from March 2013 to March 2023. This extensive dataset captures intricate patterns and dependencies inherent in real-world solar power data, making it a valuable resource for accurate solar power prediction.

Table 2. Statistic of the Dataset

Features	Units	Range	Features	Units	Range
Power	Watt/m W	15–1300	Wind Speed	m/s	0–74
Module Temperature	C/m	1–6	Wind Direction	degrees	0–255
Barometer data	Hg	1000–1033	Date	dd/mm/yyyy	01-01-2013– 31-12-2022
Humidity	g.m ³	15–98	Time	hh:mm	06:00–18:35
Temperature	C/m	7–34	Solar Radiation	W/m ²	0–1388

4.2 Evaluation Metrics

To assess the efficiency of the proposed Hybrid Contextual Model, we perform a quantitative comparison analysis with state-of-the-art deep learning models such as LSTM, Bi-LSTM, and GRU. We employ evaluation metrics, including the R^2 Score and the Mean Absolute Percentage Error (MAPE) score, as defined by Eqs. 6 and 7, respectively. The Mean Absolute Percentage Error (MAPE) measures the accuracy of a model by calculating the average absolute percentage

¹ <https://github.com/deepakkanneganti9/A-Hybrid-Contextual-Deep>.

² <https://research.curtin.edu.au/research-areas/energy-transition/geep-green-electric-energy-park/>.

error between the predicted values (\hat{y}_i) and the actual values (y_i). This metric provides an intuitive understanding of the prediction error as a percentage. The R² Score, also known as the coefficient of determination, is a statistical measure that explains the proportion of variance in the dependent variable that is predictable from the independent variables. It indicates how well the predicted values (\hat{y}_i) approximate the actual values (y_i).

$$MAPE = \frac{100}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \quad (6)$$

$$R2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (7)$$

4.3 Performance Comparison

The experiments primarily compare the proposed multivariate deep learning-based hybrid model with individual models such as LSTM, Bi-LSTM, and GRU. The performance metrics, including Mean Absolute Percentage Error (MAPE) and R-squared (R2), evaluated at different time intervals, demonstrate that the proposed hybrid model significantly outperforms the individual models. As illustrated in Table 3, the hybrid model achieves a MAPE score of 4.04 and an R2 value of 0.96, indicating superior accuracy and fit compared to the LSTM model's MAPE of 16.0 and R2 of 0.92, and the GRU model's MAPE of 20.0 and R2 of 0.84. These results highlight the hybrid model's enhanced capability in providing precise predictions and maintaining consistency over time, showcasing its robustness and reliability in solar power forecasting.

Table 3. Comparison of predictions effects of LSTM, GRU, and hybrid model at different time intervals

Time Step	Model	R2 score	MAPE	MAE
1 min	LSTM [21]	0.921	16.12	56.3
1 min	Bi-LSTM [22]	0.9031	13.2187	52.1
1 min	GRU [23]	0.8426	20.6219	100.6701
1 min	Hybrid Contextual model	0.9623	4.0325	24.1777

Figure 4 visually represents the solar forecasting results of all models against the actual values over six days with a time step of five minutes. It is evident that the hybrid model's predictions (purple line) closely follow the actual values (green line), especially during peaks and troughs, demonstrating its robustness in capturing power fluctuations. In contrast, the LSTM (blue line) and GRU (red line) models, while showing relatively good performance, exhibit noticeable deviations from the actual values, particularly during periods of rapid change.

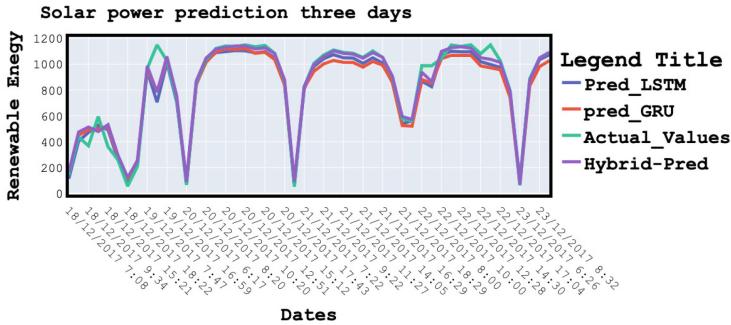


Fig. 4. Three-Day Solar Power Forecasting Comparison.

Although effective in predicting low power levels, the GRU model shows lower overall accuracy than the LSTM and Bi-LSTM models. The LSTM model, despite its general accuracy, struggles during fluctuating periods. These observations indicate that the hybrid model effectively combines the strengths of both LSTM and GRU, resulting in a more reliable and precise forecasting tool. Thus, the hybrid model is the most effective in predicting solar power with high fidelity to actual values, underscoring its practical applicability in real-world scenarios. In-depth analysis at various intervals is performed to further investigate the hybrid model's performance. We gained valuable insights into the model's predictive capabilities by partitioning the dataset into different time intervals, such as 1 min, 5 min, 10 min, 15 min, 30 min, and 1 h. We assessed its proficiency in accurately forecasting future values within each interval through rigorous testing. This approach allowed us to comprehensively evaluate the hybrid model's performance across various time scales, enabling us to derive meaningful conclusions about its predictive accuracy. Table 4 displays the evaluation metrics values of the hybrid contextual model at different time steps. The hybrid model maintained the highest value of the R2 score, which determines that the model is not facing an over or under-fitting problem. The hybrid model has the highest R2 score, ranging from 0.94–0.96. Out of all the time intervals models, the results of

Table 4. Comparison of predictions effects of the hybrid model at different time intervals

Time Step	Model	R2 score	MAPE
1 min	Hybrid Contextual model	0.9538	5.6231
5 min	Hybrid Contextual model	0.9563	5.4395
10 min	Hybrid Contextual model	0.9578	5.4903
15 min	Hybrid Contextual model	0.9578	5.6531
30 min	Hybrid Contextual model	0.9475	6.8361
1 h	Hybrid Contextual model	0.9648	6.1693

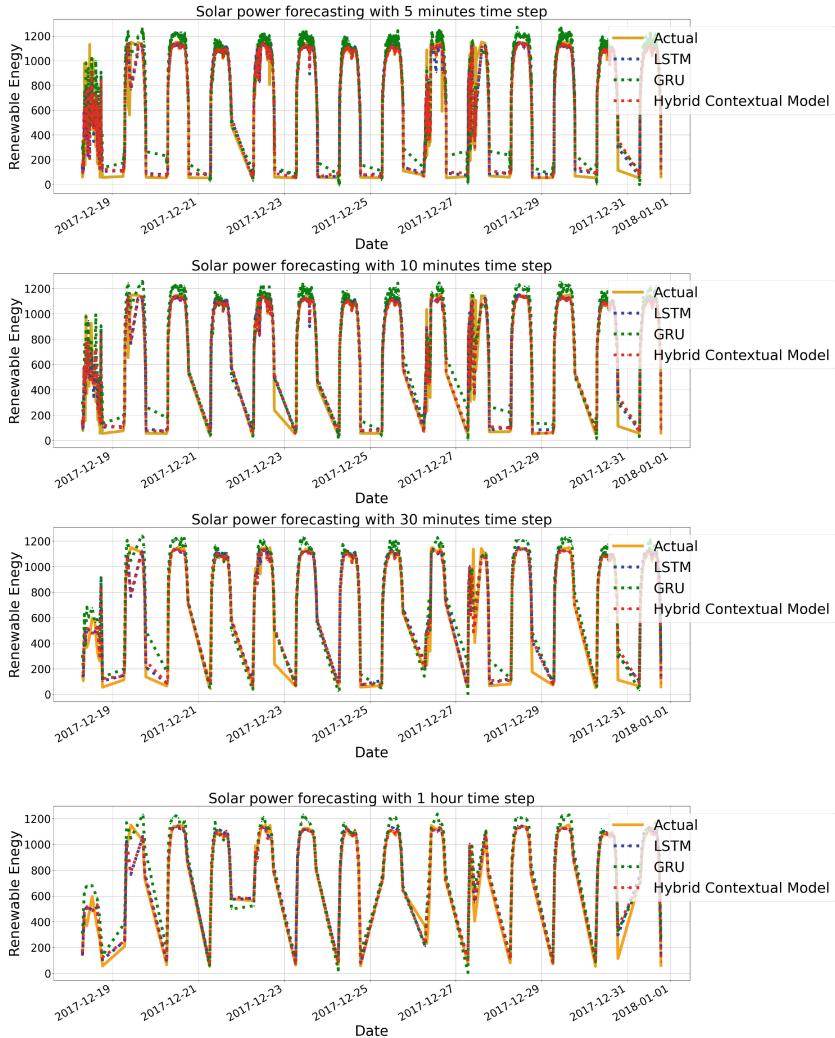


Fig. 5. Hybrid Contextual Model Predictions at Different Time Intervals. (a) 5 min, (b) 10 min, (c) 30 min, (d) 1 h.

the model at 30 min times step has the best values for all the evaluation metrics parameters.

Figure 5 illustrates the model's performance across different periods. Specifically, we compare the model's performance using data with varying time steps, including 1, 10, 15, 30-min, and 1-h intervals. As the figure shows, the R2 score value significantly increases as the time step increases. Remarkably, the model achieves highly accurate solar predictions at a one-hour time step. At this interval, the hybrid model achieves an impressive R2 score of 0.9648 and a MAPE

(Mean Absolute Percentage Error) score of 6.16. These results emphasize the model's exceptional performance in highly accurate predictions at a time resolution of 1 h.

5 Conclusions

We propose a deep learning-based hybrid contextual model for power forecasting by integrating Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Gated Recurrent Unit (GRU) architectures to address the forecasting challenges of solar power due to its stochastic nature. Using a decade of big data (2013–2023) solar datasets, the model combines the strengths of LSTM, Bi-LSTM, and GRU to capture long-term dependencies and manage large datasets effectively. Extensive experiments show that the hybrid model significantly outperforms traditional deep learning models, particularly during weather changes and high-power fluctuations, with superior R^2 and MAPE scores. Specifically, our model achieves an R^2 of 0.96 and MAPE of 4.03 after hyperparameter tuning, highlighting its robustness and reliability. The hybrid model addresses the limitations of existing AI-based solutions, facilitating reliable and consistent power forecasts, optimizing grid operations, and aligning with demand. However, there is increased complexity as the hybrid model incurs higher computational costs compared to single models due to the integration of multiple architectures. In future work, we will consider additional environmental factors, apply transfer learning for different locations, and develop more efficient algorithms to reduce computational requirements without sacrificing accuracy.

Acknowledgement. This research was made possible by the data provided from the Green Electric Energy Park of Curtin University. The statements made herein are solely the responsibility of the authors.

References

1. IEAWorld Energy Outlook: International energy agency (2018). India Energy Outlook, vol. 24 (2019)
2. Agrell, P.J., Bogetoft, P.: International benchmarking of electricity transmission system operators. In: 11th International Conference on the European Energy Market (EEM14), pp. 1–5. IEEE (2014)
3. Panteli, M., Mancarella, P.: Influence of extreme weather and climate change on the resilience of power systems: impacts and possible mitigation strategies. *Electric Power Syst. Res.* **127**, 259–270 (2015)
4. Kaushika, N.D., Tomar, R.K., Kaushik, S.C.: Artificial neural network model based on interrelationship of direct, diffuse and global solar radiations. *Solar Energy* **103**, 327–342 (2014)
5. Praynlin, E., Jenson, J.I.: Solar radiation forecasting using artificial neural network. In: 2017 Innovations in Power and Advanced Computing Technologies (i-PACT). IEEE (2017)

6. Cao, J., Lin, X.: Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks. *Energy Convers. Manage.* **49**, 1396–1406 (2008)
7. Gensler, A., Henze, J., Sick, B., Raabe, N.: Deep learning for solar power forecasting—an approach using autoencoder and LSTM neural networks. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE (2016)
8. Peng, T., Zhang, C., Zhou, J., Nazir, M.S.: An integrated framework of bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting. *Energy* **221**, 119887 (2021)
9. Wojtkiewicz, J., Hosseini, M., Gottumukkala, R., Chambers, T.L.: Hour-ahead solar irradiance forecasting using multivariate gated recurrent units. *Energies* **12**, 4055 (2019)
10. Mathiesen, P., Kleissl, J.: Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states. *Solar Energy* **85**, 967–977 (2011)
11. Li, G., Liao, H., Li, J.: Discussion on the method of grid-connected PV power system generation forecasting. *Univ. J. Yunnan Norm* (2011)
12. Li, R., Li, G.M.: Photovoltaic power generation output forecasting based on support vector machine regression technique. *Electric Power* **41**, 74–78 (2008)
13. Alsharif, M.H., Younes, M.K., Kim, J.: Time series ARIMA model for prediction of daily and monthly average global solar radiation: the case study of Seoul, South Korea. *Symmetry* **11**, 240 (2019)
14. Yang, D., Dong, Z., Reindl, T., Jirutitijaroen, P., Walsh, W.M.: Solar irradiance forecasting using spatio-temporal empirical kriging and vector autoregressive models with parameter shrinkage. *Solar Energy* **103**, 550–562 (2014)
15. Alonso, A.M., García-Martos, C., Rodríguez, J., Jesús Sánchez, M.: Seasonal dynamic factor analysis and bootstrap inference: application to electricity market forecasting. *Technometrics* **53**, 137–151 (2011)
16. Giebel, G., Brownsword, R., Kariniotakis, G., Denhard, M., Draxl, C.: The state-of-the-art in short-term prediction of wind power: a literature overview. DTU (2011)
17. Hong, T., Wilson, J., Xie, J.: Long term probabilistic load forecasting and normalization with hourly information. *IEEE Trans. Smart Grid* **5**, 456–462 (2013)
18. Qing, X., Niu, Y.: Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **148**, 461–468 (2018)
19. Ju, J., Liu, K.N., Liu, F.A.: Prediction of SO₂ concentration based on AR-LSTM neural network. *Neural Process. Lett.* **55**, 5923–5941 (2022). <https://doi.org/10.1007/s11063-022-11119-7>
20. Lim, S.C., Huh, J.H., Hong, S.H., Park, C.Y., Kim, J.C.: Solar power forecasting using CNN-LSTM hybrid model. *Energies* **15**, 8233 (2022)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
22. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Sig. Process.* **45**, 2673–2681 (1997)
23. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)



Knowledge Tracing Method Based on Enhanced Global and Local Knowledge State Representation

Jiagui Xiong, Hua Chen^(✉), Jiayu Hu, Xinyu Zhou, Wenlong Ni, and Hongwei Li

Jiangxi Normal University, Nanchang 330022, Jiangxi, China

{jiaguixiong, hua.chen, jyhu, xyzhou, wni, lihongwei}@jxnu.edu.cn

Abstract. In the realm of Intelligent Tutoring Systems (ITS), knowledge tracing plays a vital role in capturing students' evolving knowledge states to predict their future performance. Although significant strides have been made in deep learning-based knowledge tracing research, current methods often fall short of adequately considering students' global knowledge state representation and modeling local knowledge state representation across different time spans. To address this issue, we introduce a method, Knowledge Tracing Method Based on Enhanced Global and Local Knowledge State Representation (EGLKT), a model that thoroughly incorporates global knowledge state representation and introduces a dynamic multi-step radiation-based local feature extraction method to model local knowledge state representation comprehensively. Our experiments on four public datasets show that EGLKT surpasses comparative models in terms of AUC, affirming its efficacy and potential.

Keywords: Knowledge tracing · Intelligent tutoring systems · Cognitive modeling · Deep learning

1 Introduction

With the rapid advancement of artificial intelligence (AI) technology, ITS [1] has garnered widespread attention in the field of education. Compared to traditional in-person education, a significant advantage of ITS is its ability to fully retain learners' interaction records, allowing for the assessment of the effectiveness of learning behaviors and knowledge mastery. This enables it to dynamically adjust teaching content and difficulty based on evaluations of students, thereby supporting personalized learning more effectively [2]. In recent years, breakthroughs in deep learning research have further enhanced the potential and application prospects of ITS in automated teaching and personalized education [3, 4].

One critical task of ITS is to predict students' future performance, and knowledge tracing is a primary method for achieving this. Knowledge tracing analyzes students' learning behaviors and response history to dynamically assess their mastery of specific skills and make predictions about their future learning performance. With the advancements in deep learning technologies, numerous deep learning-based knowledge tracing

models have been developed to enhance prediction accuracy. For instance, DKT [5] utilizes Recurrent Neural Networks (RNNs) to capture students' learning dynamics, significantly enhancing prediction performance. DKT was the first model to apply deep learning to knowledge tracing. Subsequently, knowledge tracing models based on attention mechanisms, such as SAKT [6] and AKT [7] have further enhanced prediction accuracy.

However, previous methods have only considered students' local knowledge state for predicting future performance. Recent research has attempted to improve these methods by incorporating global knowledge state representation of student, such as the Global and Local Neural Cognitive (GLNC) [8] model, which integrates cognitive diagnosis [9] with knowledge tracing to enhance the accuracy of predicting students' future performance. Cognitive diagnosis is an important research area in personalized education, originating from psychology and based on the Item Response Theory (IRT) proposed by Danish statistician Georg Rasch and American psychometricians [10]. Its main goal is to evaluate students' mastery of skills by analyzing their historical response data, thereby reflecting their global knowledge state.

Global knowledge state plays a significant role in predicting student performance by providing a comprehensive view of students' cognitive levels across all interactions, aiding in the assessment of their learning abilities, and identifying long-term trends in their learning progress. Additionally, the global knowledge state reduces sensitivity to local dynamic changes, enhancing the overall reliability of the model. By integrating this global knowledge state, the predictive information becomes more comprehensive, thus improving the accuracy of predictions regarding students' future performance.

However, the GLNC model has certain limitations in encoding students' knowledge states. Specifically, the model employs random initialization to encode the global knowledge state, which may oversimplify the actual knowledge levels of students. Furthermore, in terms of local knowledge state representation, the GLNC model uses a fixed-step-length historical interaction sequence to encode students' local knowledge state, limiting the model's ability to capture knowledge state representations across different time spans adequately.

To overcome these limitations, we propose the EGLKT model. In our approach, we use a Multilayer Perceptron (MLP) to capture the global knowledge state representation of students, allowing the model to learn more accurately global knowledge state based on students' historical interaction sequences. For local knowledge state representation, we introduce a dynamic multi-step radiation-based local feature extraction method to more comprehensively model students' local knowledge states across multiple time spans. By fusing global and local knowledge states, we obtain a comprehensive knowledge state representation, thereby improving the accuracy of predicting students' future performance. These improvements aim to enhance the accuracy of student performance prediction and provide higher-quality support for ITS. In summary, the main contributions of this paper are as follows:

- We propose the EGLKT model, which considers both global and local knowledge state representations to more comprehensively model students' knowledge state representation, thereby improving the accuracy of student performance prediction.

- We introduce a global knowledge state representation method based on MLP and a dynamic multi-step radiation-based local feature extraction method, which can more finely capture knowledge state representations over multiple time spans.
- We conducted experimental evaluations on four public datasets, and the results show that the EGLKT model achieves the best performance in comparison with multiple baseline models.

2 Related Work

Knowledge tracing is a critical research area in educational technology. It evaluates students' knowledge state and predicts their future performance by analyzing their learning behaviors and historical response sequences. Through knowledge tracing, educators can gain deep insights into students' mastery levels and learning needs, enabling personalized learning support and guidance. This allows educators to adjust teaching strategies, provide targeted materials, feedback, and supplementary resources based on students' actual learning conditions, thereby maximizing learning outcomes and growth. Knowledge tracing plays a vital role in personalized education, adaptive learning, and intelligent educational technologies.

Early research on knowledge tracing was conducted by Corbett and Anderson [11] in 1994, utilizing Hidden Markov Models and Bayesian networks to track students' learning processes, known as Bayesian knowledge tracing (BKT). Over time, the BKT model has undergone several improvements. For instance, Baker et al. [12] in 2008 introduced regression steps and feature generation, while Pavlik et al. [13] proposed Performance Factor Analysis, which uses logistic functions to estimate the probability of learners' knowledge mastery.

With the advent of deep learning, researchers have increasingly begun to apply deep learning models to knowledge tracing, resulting in a significant improvement in the accuracy of modeling knowledge states. In 2015, Piech et al. [5] introduced the DKT, the first to apply deep learning to knowledge tracing. Subsequently, several RNN and Long Short-Term Memory (LSTM) based models, such as DKVMN [14] and SKVMN [15], have been developed, further enhancing accuracy in prediction.

The introduction of attention mechanisms has led to the emergence of many related models in knowledge tracing. In 2019, Pandey and Karypis [6] proposed the SAKT model, which uses attention mechanisms to capture long-term dependencies in student learning. In 2020, Ghosh et al. [7] proposed the AKT, enhancing the model's interpretability and accuracy. Subsequently, Choi et al. [16] introduced the SAINT model, which incorporates an encoder-decoder architecture to finely model student interaction sequences. Pandey and Srivastava [17] proposed the RKT, specifically addressing problem relationships and forgetting behaviors, which led to further improved performance.

The advancement of Graph Neural Networks (GNN) has demonstrated strong capabilities in capturing and predicting knowledge state representation. For example, Nakagawa et al. [18] proposed the GKT, which uses GNN to model the graph structure among skills. Subsequent models, such as GIKT [19] and SKT [20], further enhance interpretability and prediction performance by modeling the various structural relationships that exist among knowledge concepts.

However, previous knowledge tracing methods primarily rely on students' local knowledge state, with less emphasis on their global knowledge state, which limits the model's effective assessment of students' overall knowledge mastery from a global perspective. Recently, Su et al. [8] introduced the GLNC model, which combines global and local knowledge state representations, significantly improving prediction accuracy.

Although the GLNC model has taken student's global knowledge states into account, there is still room for improvement in the encoding and integration of students' knowledge states. In contrast to previous models, our proposed EGLKT model not only fully considers students' global knowledge state but also encodes and integrates students' local knowledge states across multiple time spans to optimize the representation of students' knowledge state. As a result, EGLKT captures the dynamic changes in local knowledge states over different time spans and assesses students' cognitive states from a global perspective, providing a comprehensive understanding of their knowledge mastery. This addresses certain limitations of previous methods and improves the accuracy of predictions regarding students' future performance.

3 Problem Definition

The task of knowledge tracing involves predicting a student's chance of answering the next question correctly, based on their historical interaction. Specifically, a student's interaction sequences are represented $X_t = \{(q_1, S_1, a_1), (q_2, S_2, a_2), \dots, (q_t, S_t, a_t)\}$. Here, q denotes the question, S represents the set of skills associated with the question, and the binary variable a indicates the student's response (1 for correct, 0 for incorrect). The objective is to predict the probability that the student will answer the question q_{t+1} correctly, denoted as $p_{t+1} = P(a_{t+1} = 1|X_t, q_{t+1})$. This task typically employs binary cross-entropy as the loss function.

4 The EGLKT Model

In this section, we provide an overview of the EGLKT model's architecture, as depicted in Fig. 1. First, we employ a Transformer Layer to capture interactions between questions and skills, allowing for a nuanced understanding of their relationships. Following this, the MLP models the student's historical interaction sequences, effectively capturing the global knowledge state representation. Subsequently, a dynamic multi-step radiation-based local feature extraction method is used to derive local knowledge state representations across different time spans. These local knowledge state representations are then fused using an attention mechanism, which allows the model to prioritize relevant information. Finally, we show how the global and local knowledge state representations are adaptively combined to comprehensively represent the student's knowledge state.

4.1 Information Embedding and Fusion

In student-item interactions, four factors are involved: student s , question q , the set of skills S_q associated with the q , and the student's answer a . To integrate these factors, we propose a method.

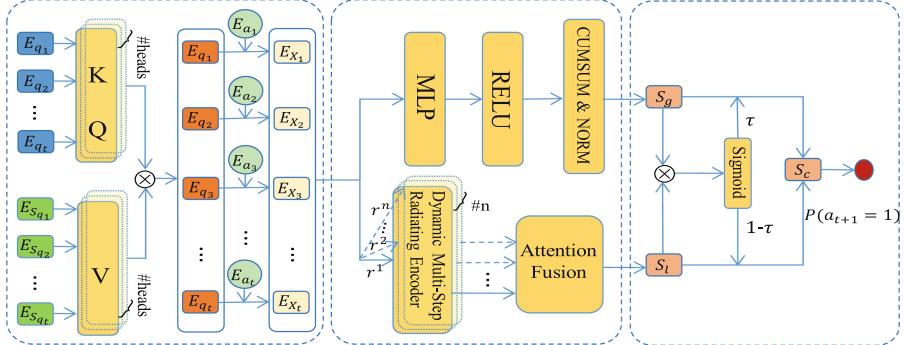


Fig. 1. illustrates the structure of our EGLKT model designed for knowledge tracing tasks. The model consists of three main parts: (1) **Information fusion**: This part integrates embeddings of questions (E_q), skills (E_{S_q}), and answers (E_a) to obtain a unified representation vector $\{E_{X_1}, E_{X_2}, \dots, E_{X_t}\}$. (2) **Knowledge state representation**: This part takes $\{E_{X_1}, E_{X_2}, \dots, E_{X_t}\}$ as input to model the global knowledge states (S_g) and local knowledge states (S_l). For local knowledge state representation, multiple historical interaction sequences with different time spans are selected based on parameter r to capture recent learning dynamics. This produces multiple local knowledge state representations across different time spans, which are fused using an attention mechanism. (3) **State representations fusion and prediction**: This part combines the S_g and S_l to obtain the final knowledge state representation (S_c). Using S_c , the model predicts the probability of the next answer being correct.

Initially, We use an embedding layer to map the question q and the student's answer a to their respective embeddings E_q and E_a . The skill set S_q is mapped to skill embeddings $E_{S_q} = [e_{s_1}, e_{s_2}, \dots, e_{s_n}]$, which then undergo processing via a MLP to capture their high-dimensional features, as indicated by the formula:

$$E_{S_q} = W_1 E_{S_q} + b_1 \quad (1)$$

where W_1 is a learnable weight matrix and b_1 is a bias term. Subsequently, the question embedding E_q and the skills embedding E_{S_q} are input into a Transformer Layer for processing. The multi-head attention mechanism within the Transformer Layer effectively captures significant interactions between questions and skills, thereby enhancing information compared to traditional concatenation methods. The attention mechanism operates as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where $K = Q = E_q$, $V = E_{S_q}$. The attention mechanism's output is processed through multi-head attention:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_n)W_h \quad (3)$$

Where W_h is a learnable weight matrix and $\text{head}_i = \text{Attention}(Q, K, V)$. Subsequently, the output of the multi-head attention is processed through feed-forward neural network layers and layer normalization:

$$\text{FFN}(x) = \text{Relu}(xW_2 + b_2)W_3 + b_3 \quad (4)$$

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \quad (5)$$

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \odot \gamma + \beta \quad (6)$$

where W_2, W_3 are learnable weight matrices, b_2, b_3 are bias terms, μ is the mean of input vector x , σ^2 is the variance of input vector x , ϵ is a constant to avoid division by zero, γ is a learnable scaling parameter, and β is a learnable shift parameter. Through these computations, we obtain the fused interaction pair representation E_f . Finally, E_f is concatenated with the embedding representation E_a of the answer to form the final input representation:

$$E_{X_t} = \text{Concat}(E_f, E_a) \quad (7)$$

By integrating information from question, skills, and answer, we construct a comprehensive representation vector for subsequent model predictions.

4.2 Knowledge State Representation

Global Knowledge State Representation. Understanding students' global knowledge state is crucial for predicting their future performance, as the global knowledge state represents the overall knowledge level of a student at a specific point in time. To this end, we propose a method to encode students' global knowledge state.

Firstly, the input data E_{X_t} is processed using a MLP to obtain an initial representation of the global knowledge state:

$$h_t = \sigma(W_4 E_{X_t} + b_4) \quad (8)$$

where W_4 is the learnable weight matrix, b_4 is the bias term, and σ is the *sigmoid* activation function. Subsequently, perform cumulative sum processing on h_t and gradually accumulate the information in the sequence:

$$c_t = \sum_{i=1}^t h_{t,i} \quad (9)$$

where c_t represents the cumulative sum at time step t . This operation ensures that the state at each time step incorporates information from the current step as well as from all previous steps, thereby capturing information from the entire history sequence. To prevent c_t from becoming too large, we normalize it to obtain the final global knowledge state representation.

$$S_g = \frac{c_t}{t} \quad (10)$$

This step ensures numerical stability, smoothing the variation of knowledge state representation, reducing numerical discrepancies, and enhancing the training stability of the model.

Local Knowledge State Representation. The prediction of student's future performance is significantly influenced by recent interaction sequences, necessitating thorough consideration of the student's local knowledge state. To address this, we propose a dynamic multi-step radiation-based local feature extraction method for modeling this state. Specifically, we introduce a scaling parameter r and select n subsequences with different time spans from the student's entire interaction sequence based on r . Each subsequence is processed through a Transformer Layer to obtain local feature representation with varying time spans. The detailed steps are as follows:

Firstly, define a subsequence X_t^r from the student's interaction sequence X_t as:

$$X_t^r = \{(q_{\lfloor r \cdot T \rfloor}, S_{\lfloor r \cdot T \rfloor}, a_{\lfloor r \cdot T \rfloor}), (q_{\lfloor r \cdot T \rfloor + 1}, S_{\lfloor r \cdot T \rfloor + 1}, a_{\lfloor r \cdot T \rfloor + 1}), \dots, (q_t, S_t, a_t)\} \quad (11)$$

Subsequently, apply a Transformer Layer to encode X_t^r and obtain its local knowledge state representation H_t^r :

$$H_t^r = \text{Transformer}(X_t^r) \quad (12)$$

For instance, when the scaling parameter r is set at 0.6, the subsequence selected from the student's entire interaction sequence is:

$$X_t^{0.6} = \{(q_{\lfloor 0.6 \cdot t \rfloor}, S_{\lfloor 0.6 \cdot t \rfloor}, a_{\lfloor 0.6 \cdot t \rfloor}), (q_{\lfloor 0.6 \cdot t \rfloor + 1}, S_{\lfloor 0.6 \cdot t \rfloor + 1}, a_{\lfloor 0.6 \cdot t \rfloor + 1}), \dots, (q_t, S_t, a_t)\}$$

Proceed sequentially with $r = 0.7, 0.8, 0.9$ to obtain multiple local knowledge state representations spanning different time spans:

$$H_t^{0.7} = \text{Transformer}(X_t^{0.7})$$

$$H_t^{0.8} = \text{Transformer}(X_t^{0.8})$$

$$H_t^{0.9} = \text{Transformer}(X_t^{0.9})$$

Finally, stack these knowledge state representations of different time spans together to form a set of local features capturing information across multiple time spans: $\{H_t^{0.6}, H_t^{0.7}, H_t^{0.8}, H_t^{0.9}\}$. This approach effectively captures changes in the student's local knowledge state representation.

To integrate these local knowledge state representations, an attention mechanism is utilized to fuse them. By adaptively assigning weights to each local feature, this approach effectively captures changes in the student's local knowledge state representation highlighting representations from important time spans while suppressing those from less relevant time spans. The specific implementation is as follows:

$$S_l = \sum_{i=1}^n \alpha_i H_t^i \quad (13)$$

where attention weights α_i are computed using an attention mechanism:

$$\alpha_i = \frac{\exp(score(H_t^i))}{\sum_{j=1}^n \exp(score(H_t^j))} \quad (14)$$

In summary, through dynamic multi-step radiation-based feature extraction, this method comprehensively models the student's knowledge state representations across different historical interaction time spans.

Integration of Global and Local Knowledge State Representation. To integrate S_g and S_l for predicting student performance on future tasks, we propose a method that calculates the similarity between global and local knowledge state representations and fuses them based on assigned weights to obtain a comprehensive knowledge state representation. Firstly, we compute the similarity between the S_g and S_l as:

$$sim = \sigma(S_g \odot S_l) \quad (15)$$

Where, \odot denotes element-wise multiplication. This step aims to capture the relationship between S_g and S_l , generating a weight coefficient sim . A higher value of sim in a specific feature dimension indicates a stronger preference towards the S_g , while a lower value indicates a stronger preference towards the S_l .

Subsequently, we further process sim through a fusion network to produce a final weight for weighted fusion:

$$\tau = \sigma(W_5 sim + b_5) \quad (16)$$

where W_5 is the learnable weight matrix, b_5 is the bias term. Finally, we fuse the S_g and S_l using the computed weight τ to obtain the ultimate representation of student knowledge state:

$$S_c = \tau \cdot S_g + (1 - \tau) \cdot S_l \quad (17)$$

This approach effectively integrates the S_g and S_l , resulting in a more comprehensive representation of the state of student knowledge for predicting student future performance.

4.3 Model Learning

In the final part of this study, our objective is to predict the probability p_{t+1} of a student correctly answering the next question q_{t+1} . The prediction model takes as input the student's current knowledge state S_c and the embedding vector $E_{q_{t+1}}$ of the q_{t+1} . This input is processed through a Fully Connected Layer, followed by a sigmoid function to produce the probability p_{t+1} , where $p_{t+1} \in [0, 1]$:

$$p_{t+1} = \sigma(MLP(concat(S_c, E_{q_{t+1}}))) \quad (18)$$

During the entire process, model parameters are learned by minimizing the standard cross-entropy loss between the predicted probability p_{t+1} and the actual labels:

$$L = -\sum_{t+1}(r_{t+1} \log p_{t+1} + (1 - r_{t+1}) \log(1 - p_{t+1})) \quad (19)$$

5 Experiments

5.1 Datasets

We evaluated our method and all comparison methods on four public real-world datasets, and the detailed statistical results are presented in Table 1.

Table 1. Dataset statistics

Dataset	#of learners	#of questions	#of skills	#of interaction records
ASSISTments17	1709	3126	102	942816
EdNet	5000	12372	188	347864
JunYi	247606	722	41	259825992
STATICS	335	1224	85	361092

- ASSISTments17: This dataset consists of student response records from elementary mathematics courses in the ASSISTments ITS. It was released in 2017 and includes practice data from the academic years 2004–2005 and 2005–2006.
- EdNet [21]: EdNet is a large-scale hierarchical dataset, and we randomly selected a subset for experimentation.
- JunYi [22]: This dataset was collected from a uniform education platform in Taiwan, spanning from November 2010 to March 2015.
- STATICS [23]: The STATICS dataset was collected from an engineering statics course at Carnegie Mellon University in the fall of 2011.

These datasets cover different subjects and learning environments, providing a rich experimental basis for evaluating our method.

5.2 Baseline Methods

To evaluate the effectiveness of our model, we compared it against several representative knowledge tracing methods. All methods were implemented using open-source code and tested on a Linux system equipped with an NVIDIA 3080 GPU.

- DKT [5]: This was the first model to apply deep learning to knowledge tracing, using RNNs to model learners' knowledge states.
- DKVMN [14]: Inspired by memory-augmented neural networks, this model uses a dynamic memory matrix to predict students' knowledge states.
- SKVMN [15]: The model combines skill awareness with the Variational Memory Network to predict students' knowledge states.
- SAKT [6]: The first model to introduce the attention mechanism, capturing the internal correlation of learning sequence data through self-attention.
- AKT [7]: This model integrates the self-attention mechanism with a psychometric model, computing attention weights using the monotonic attention mechanism and the assumption of time-seriality.
- GIKT [19]: The model uses GNN to represent the complex relationships between questions and knowledge concepts and predicts learners' responses by aggregating embedding.
- GLNC [8]: The model combines cognitive diagnosis and knowledge tracing to improve the accuracy of student performance prediction.

5.3 Evaluation Metric

To evaluate the performance of our model and the baseline models, we used the classification metric AUC (Area Under the Curve). AUC values range from 0 to 1, where 0.5 indicates random prediction and higher values indicate stronger predictive ability.

5.4 Comparative Experiments

Table 2. Comparisons of the AUC results of different models on the four datasets

Methods	Usage of Questions	Usage of Skills	AUC			
			ASSISTments17	EdNet	Jun Yi	STATICS
DKT	✗	✓	0.7300	0.7007	0.7406	0.8423
DKVMN	✗	✓	0.7631	0.7478	0.7862	0.8172
SKVMN	✗	✓	0.7529	0.7091	0.7618	0.8112
SAKT	✗	✓	0.7670	0.7598	0.7752	0.8379
AKT	✓	✓	0.7737	0.7621	0.7833	0.8424
GIKT	✓	✓	0.7751	0.7632	0.7825	0.8386
GLNC	✓	✓	0.7766	0.7643	0.7862	0.8424
EGLKT	✓	✓	0.7818	0.7655	0.7901	0.8503

Table 2 presents the performance of various knowledge tracing methods across four datasets. The experimental results illustrate that the EGLKT model consistently outperforms other comparative models across all datasets. For example, on the ASSISTment17 dataset, EGLKT achieves an AUC of 0.7818, which is notably higher than DKT's 0.7300 and GLNC's 0.7766. On the EdNet dataset, EGLKT achieves an AUC of 0.7655, showing an improvement of 6.48% over DKT's 0.7007 and a slight increase of 0.12% over GLNC's 0.7643. Similarly, on the Jun Yi dataset, EGLKT achieves an AUC of 0.7901, surpassing DKT's 0.7406 and GLNC's 0.7862. On the STATICS dataset, EGLKT achieves an AUC of 0.8503, which exceeds SKVMN's 0.8112 and GLNC's 0.8424. In summary, the EGLKT model demonstrates significantly superior performance across all datasets compared to other comparative models, thus validating its effectiveness and potential in the field of knowledge tracing. The superiority and contributions of our method can be explained as follows:

- **Integration of global and local knowledge states:** The EGLKT model integrates global and local knowledge states to overcome the limitations of relying solely on local knowledge state representation. The global knowledge state provides a comprehensive view of the student's overall cognition, while the local knowledge state offers detailed insights into performance at specific time points. This integration improves prediction accuracy by capturing knowledge states across different time spans more comprehensively.

- **Optimization through MLP:** EGLKT utilizes a MLP for modeling the global knowledge state, which enhances the precision and stability of the global knowledge state representation compared to traditional methods that use random initialization.
- **Introduction of dynamic multi-step radiation-based local feature extraction method:** EGLKT models the local knowledge state by using a dynamic multi-step radiation local feature extraction method. This method allows for flexible handling of learning performance over different time spans by extracting dynamic features. It overcomes the limitations of fixed-step sequences and enhances the capability to model local knowledge states.

These improvements can enhance the performance of the model in terms of knowledge tracing, so as to better meet the needs of personalized learning.

5.5 Ablation Experiments

In this section, we aim to demonstrate the effectiveness of each component in our model through experiments conducted on four datasets. We compare the performance of several variants of EGLKT:

- EGLKT w/o g, which excludes the measurement of students' global knowledge state.
- EGLKT w/o l, which excludes the measurement of students' local knowledge state.
- EGLKT w/o s, which does not consider skills related to the question.

Table 3 shows that the AUC values for all variants are significantly lower than those of the full model, highlighting the crucial role of integrating global knowledge state, local knowledge state, and skill information in enhancing prediction accuracy. Specifically, the performance drop in EGLKT w/o g indicates the importance of the global knowledge state in capturing students' overall cognitive state and enhancing the stability of the model. The reduced performance of EGLKT w/o l demonstrates that the local knowledge state is essential for capturing the dynamic learning changes of students, and its absence hampers the model's ability to reflect temporal knowledge variations. The results for EGLKT w/o s reveal that omitting skill information diminishes the model's ability to accurately represent question features, adversely affecting overall prediction performance. In summary, the ablation experiments confirm that the fusion of global knowledge state, local knowledge state, and skill information significantly enhances the prediction accuracy of the EGLKT model, validating its rationality and effectiveness.

Table 3. The AUC results of different variants in our model

Methods	AUC			
	ASSISTments17	EdNet	Jun Yi	STATICS
EGLKT w/o g	0.7754	0.7646	0.7881	0.8426
EGLKT w/o l	0.7740	0.7629	0.7750	0.8210
EGLKT w/o s	0.7786	0.7650	0.7890	0.8488
EGLKT	0.7818	0.7655	0.7901	0.8503

6 Conclusion

This paper proposes a model called EGLKT, which comprehensively considers students' global knowledge state and introduces a dynamic multi-step radiation-based local feature extraction method to better represent students' knowledge state. Results from experiments on four public datasets consistently show that EGLKT performs better than other models in terms of the AUC metric, confirming its effectiveness and potential. Our approach significantly enhances the accuracy of students' future performance, highlighting its broad applicability in the field of knowledge tracing.

References

- Chrysafiadi, K., Virvou, M.: Student modeling approaches: a literature review for the last decade. *Expert Syst. Appl.* **40**(11), 4715–4729 (2013)
- VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)
- Lyu, H., Sha, N., Qin, S., Yan, M., Xie, Y., Wang, R.: Advances in neural information processing systems. *Adv. Neural Inform. Process. Syst.* **32** (2019)
- Chen, X., Xie, H., Hwang, G.J.: A multi-perspective study on artificial intelligence in education: grants, conferences, journals, software tools, institutions, and researchers. *Comput. Educ. Artific. Intell.* **1**, 100005 (2020)
- Piech, C., et al.: Deep knowledge tracing. *Advances in neural information processing systems*. Association for Computing Machinery, pp. 201–204 (2015)
- Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint [arXiv: 1907.06837](https://arxiv.org/abs/1907.06837) (2019)
- Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2330–2339 (2020)
- Su, Y., et al.: Global and local neural cognitive modeling for student performance prediction. *Expert Syst. Appl.* **237**, 121637 (2024)
- Wang, F., et al.: Neural cognitive diagnosis for intelligent education systems. In: Proceedings of the AAAI Conference on artificial Intelligence, vol. 34, no. 04, pp. 6153–6161 (2020)
- Lord, F.M.: Applications of item response theory to practical testing problems. Routledge (2012)
- Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**, 253–278 (1994)

12. Baker, R.S.D., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) Intelligent Tutoring Systems. ITS 2008. LNCS, vol. 5091. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_4
13. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis—a new alternative to knowledge tracing. In: Artificial Intelligence in Education, pp. 531–538. Ios Press (2009)
14. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)
15. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 175–184 (2019)
16. Choi, Y., et al.: Towards an appropriate query, key, and value computation for knowledge tracing. In: Proceedings of the Seventh ACM Conference on learning@ Scale, pp. 341–344 (2020)
17. Pandey, S., Srivastava, J.: RKT: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 1205–1214 (2020)
18. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp. 156–163 (2019)
19. Yang, Y., et al.: GIKT: a graph-based interaction model for knowledge tracing. In: Hutter, F., Kersting, K., Lijffijt, J., Valera, I. (eds.) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2020. LMCs, vol. 12457. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67658-2_18
20. Tong, S., et al.: Structure-based knowledge tracing: An influence propagation view. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 541–550. IEEE (2020)
21. Choi, Y., et al.: Ednet: a large-scale hierarchical dataset in education. In: Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) Artificial Intelligence in Education. AIED 2020. LNCS, vol 12164. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_13
22. Chang, H.S., Hsu, H.J., Chen, K.T.: Modeling exercise relationships in e-learning: a unified approach. In: EDM, pp. 532–535 (2015)
23. Koedinger, K.R., Baker, R.S., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the EDM community: the PSLC DataShop. Handbook Educ. Data Min. **43**, 43–56 (2010)



Tensor Mutual Information for Similarity Measurement of High-Dimensional Data: An Image Classification Perspective

Joarder Kamruzzaman^{1,2} , Shaoning Pang^{1,2}(✉) , Liangfu Lu^{1,2} , and Jianwei Liu³

¹ The Internet Commerce Security Lab (ICSL), Federation University, Melbourne 3353, Australia

² Centre for Smart Analytics, Federation University, Melbourne 3353, Australia
p.pang@federation.edu.au

³ School of Mathematics, Tianjin University, Tianjin 300072, China

Abstract. Machine learning is predominantly employed to classify or cluster image and video data, leveraging the similarities among their inherent features. Amidst the profound surge in the scale of image and video data, identifying appropriate similarity metrics has become paramount for both machine learning and deep learning tasks. Mutual information, rooted in Shannon entropy, stands as a prevalent similarity metric across disciplines like statistics and deep learning. Notably, extracting mutual information between tensors while preserving the integrity of tensor space information is of significant value. In the paper, we extend the concept of mutual information between random variables to higher-order tensors, formulating definitions for tensor mutual information and entropy. Furthermore, we delve into the properties of tensor mutual information, exploring its concavity and convexity within the context of average tensor mutual information. In the realm of evolutionary algorithms, tensor mutual information introduces innovative elements: an internal radius parameter r and an embedded function $f(x)$, distinguishing it from conventional mutual information. To ensure the meaningfulness of tensor mutual information, it suffices for $f(x)$ to be a continuous function. Notably, when $r = 0$ and $f(x)$ represents the identity mapping, tensor mutual information reduces to the conventional form defined by Shannon entropy. Experimental simulations underscore the remarkable performance of this metric, both in supervised and unsupervised learning contexts, showcasing its effectiveness and potential applications.

Keywords: Image analysis · Signal processing · Mutual information · Unsupervised learning

1 Introduction

In recent years, the explosive growth of images and videos has presented increasingly daunting challenges for researchers in the fields of computer vision and

machine learning [1]. Whether it is supervised learning or unsupervised learning, similarity metric is an important component of these algorithms [2]. Therefore, it is crucial to find appropriate similarity metrics for high-dimensional tensor data [3]. Traditional similarity metrics such as Euclidean distance, Manhattan distance, mutual information, and cosine similarity are highly effective in handling low-dimensional data. However, for current high-dimensional tensor data, they are often insufficient due to the inability to extract internal spatial information. Attempts have been made to leverage Gaussian kernel functions to assess sample similarity [4], yet for high-order tensor data, this necessitates vectorization or unfolding procedures, ultimately impeding the preservation of valuable tensor spatial information. Other research endeavors propose extracting high-order similarity via sophisticated constructs such as manifolds [5] or hypergraphs [6, 7]. However, these approaches often involve intricate computational processes, rendering similarity metric a cumbersome task.

A ubiquitous similarity metric is mutual information, rooted in Shannon entropy [8], which finds applications across diverse disciplines including physics, statistics, machine learning, and deep learning [9]. It holds a pivotal position in areas like image clustering [10], image segmentation [11], video segmentation [12], and image registration [13], among many others [14–17]. Mutual information meticulously quantifies the statistical interdependence between random variables [18], enabling the measurement of nonlinear relationships that would otherwise remain undetected by conventional linear metrics. Distinct from methods that utilize rank correlation coefficients for monotonic dependencies or Pearson correlation coefficients for linear dependencies, mutual information provides a comprehensive assessment of all forms of dependencies [19]. Incorporating mutual information into clustering algorithms can minimize the statistical correlation between categories [20]. This is something traditional k -means and fuzzy C-means algorithms cannot achieve, as they minimize the total variance between different categories. Furthermore, the integration of mutual information in deep learning and machine learning domains has garnered substantial attention. Guo et al. [21] harnessed mutual information maximization to enable online continual learning, while Do et al. leveraged deep learning frameworks to enhance clustering performance by maximizing mutual information between different views [22]. Liu et al. achieved remarkable results in video-based human pose estimation by maximizing mutual information [23]. Kachouie et al. studied the realm of weighted mutual information kernel clustering algorithms [24], and Aghagolzadeh et al. introduced a hierarchical clustering methodology rooted in mutual information maximization. Their approach innovatively utilized information potential computed pairwise between data points to estimate mutual information, doing so without any preconceived assumptions regarding the data's density function [25]. This not only yielded promising results but also boasted low computational overhead. Collectively, these studies underscore the pivotal role that mutual information-driven deep learning research now plays across diverse practical applications.

Currently, despite advancements where select studies have embarked on exploring the statistical dependence among volumetric pixels and devised spatial mutual information metrics, these endeavors primarily confine their calculations to the realm of mutual information between random variables. Notably, these approaches are devoid of rigorous mathematical justifications and neglect to formally define and derive mutual information within the context of tensors. Consequently, the paper introduces the novel concept of mutual information for tensors, extending the conventional framework of mutual information between random variables to encompass higher-order tensors. Tensor mutual information framework not only encompasses the conventional understanding but also offers profound insights into the remarkable success of mutual information in prevalent deep learning models. Given that neural networks possess the remarkable capability to approximate any continuous function on a compact domain, and interestingly, the intrinsic function of tensor mutual information, denoted as $f(x)$, inherently necessitates only continuity. In the subsequent sections, we outline the notation and preliminaries in Sect. 2, delve into the concept of tensor mutual information in Sect. 3, showcase its application in unsupervised clustering in Sect. 4, and conclude our findings in Sect. 5.

2 Notation and Preliminaries

To guarantee that tensor mutual information retains the favorable attributes akin to conventional mutual information, we introduce two pivotal definitions and theorems: the concept of tensor independence and the theorem of independence preservation under continuous transformations. These foundational constructs are indispensable in substantiating the properties of tensor mutual information, as elaborated in Sect. 3.

Definition 1. Independence of tensors

Given any two tensors, \mathcal{X} and \mathcal{Y} , let us assume they comprise multiple random variables, whereby each tensor can be conceptualized as a distribution spanning a multi-dimensional space of points. If the multi-dimensional random variables constituting tensor \mathcal{X} and the multi-dimensional random variables constituting tensor \mathcal{Y} are independent, then these two tensors are considered independent.

Theorem 1. Theorem of independence preservation for continuous functions [26]

Assuming that m -dimensional random variable $X = (X_1, X_2, \dots, X_m)$ and n -dimensional random variable $Y = (Y_1, Y_2, \dots, Y_n)$ are independent of each other, then X_i ($i = 1, 2, \dots, m$) and Y_j ($j = 1, 2, \dots, n$) are also independent of each other. If the functions g and h are continuous functions, then $g(X_1, X_2, \dots, X_m)$ and $h(Y_1, Y_2, \dots, Y_n)$ are also independent of each other.

3 Tensor Mutual Information

Russakoff et al. introduced the concept of regional mutual information [27]. Specifically, they treated each image as a distribution of multi-dimensional points, where each point represented a pixel and its neighborhood. They expanded the 3×3 neighborhoods in the image sequentially into vectors, and assumed that the vector followed a multivariate Gaussian distribution. Next, the entropy of the d -dimensional Gaussian distribution can be obtained through the covariance matrix Σ_d , i.e.,

$$H(\Sigma_d) = \log((2\pi e)^{\frac{d}{2}} \det(\Sigma_d)^{\frac{1}{2}}). \quad (1)$$

Regional mutual information has achieved significant success, but there are still many issues. First, the assumption of a Gaussian distribution does not align with the actual situation. Second, the computation of entropy in the paper is based on the determinant of the covariance matrix under the Gaussian assumption. According to the definition of the probability density function and covariance matrix of the multi-dimensional Gaussian distribution, if the positions of any random variables are exchanged, the probability density function and the determinant of the covariance matrix remain unchanged. This leads to the fact that in practical applications, when the 3×3 pixels are expanded in any form, the final mutual information result remains the same, which does not conform to the one-to-one correspondence between the expansion method and the mutual information result. Thus, from this perspective, spatial information is not utilized. Then, for tensor data, how to expand it is obviously a complex problem. Tensor data contains a significant amount of spatial information, and expanding it into a vector form will disrupt this important information.

In the realm of existing research, a dichotomy emerges regarding the handling of data: some scholars presume a Gaussian distribution underpinning the data, whereas others overlook the inherent tensorial structure, directly computing mutual information among random variables. While recent endeavors have ventured into exploring the spatial organization of data and inter-pixel relationships, a pivotal aspect of mutual information, its positive semi-definiteness, remains unexplored when spatial information is factored in. Consequently, the notion of tensor mutual information necessitates fulfillment of the following criteria:

- (1) Independence from Gaussian assumption: tensor mutual information should operate without reliance on the assumption of a Gaussian distribution, broadening its applicability to diverse datasets.
- (2) Preservation of tensor integrity: it should refrain from vectorizing or flattening tensors, thereby preserving the rich, multi-dimensional structure essential for accurate information extraction.
- (3) Extension of conventional mutual information: tensor mutual information must encompass and extend the conventional concept, maintaining its desirable properties while accommodating higher dimensions.

(4) Utilization of spatial information: core to its formulation, tensor mutual information must harness the spatial cues embedded within tensors, enhancing the precision and relevance of information quantification.

(5) Data-driven distribution modeling: given the intricate structure of tensors, it is postulated that their distribution follows that of multi-dimensional random variables, with the precise distribution determined empirically from the data itself, fostering a more adaptive and accurate approach.

3.1 Tensor Mutual Information and Tensor Entropy

By extending conventional mutual information, the specific procedure for calculating tensor mutual information is provided. Fundamentally, it is effective and applicable to tensors of any order. For ease of description, the following explanation will use a third-order tensor.

(1) For two three-order tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, take each pixel of the tensor as the center and r as the radius to intercept small tensor blocks. Therefore, the corresponding dimension of small tensor block is $d = (2r+1)^3$ and the number of samples collected from each image is $N = (I_1 - 2r) \times (I_2 - 2r) \times (I_3 - 2r)$. Since we have assumed that the tensor data follows the distribution of multi-dimensional random variables, the radius r uniquely associated with the distribution of the tensor, i.e., the distribution of tensor data is d -dimensional. At this time, the distribution of small tensor blocks is also d -dimensional.

(2) Each small tensor block collected in step (1) is refined, i.e., the information contained in the small tensor block is merged. In the process of merging, the spatial information of small tensor block needs to be considered, so the details are as follows:

For d -dimensional small tensor block samples, we treat small tensor block as a distribution of d -dimensional points. We consider the pixel at the center of the small tensor block as the core. Therefore, we calculate the Pearson correlation coefficient between each pixel and the central pixel of the small tensor block, and consider these coefficients as the similarity of the corresponding pixels. Subsequently, these coefficients can be utilized as weights in linear combination processing, such as a d -dimensional random variable is transformed into a 1-dimensional random variable. This 1-dimensional random variable represents the original d -dimensional small tensor block, and this random variable contains the local spatial information of the small tensor block. Specifically formulated as:

$$f(x_1, x_2, \dots, x_{(2r+1)^3}) = \frac{1}{(2r+1)^3} (k_1 x_1 + k_2 x_2 + \dots + k_{(2r+1)^3} x_{(2r+1)^3}), \quad (2)$$

where x_i is the random variable represented by each pixel of the small tensor block, and k_i is the Pearson correlation coefficient between each pixel and the central pixel of the small tensor block. The function $f(x)$ here is not unique, just ensure that $f(x)$ is continuous.

(3) For two three-order tensors \mathcal{X}, \mathcal{Y} , it can be seen from step (2) that two random variables can be found to represent the two tensors, and then the mutual information of these two random variables can be calculated to obtain the final tensor mutual information.

Formulating the above process:

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{\substack{\mathcal{M} \in \mathcal{X} \\ f(\mathcal{M}) \in X}} \sum_{\substack{\mathcal{N} \in \mathcal{Y} \\ f(\mathcal{N}) \in Y}} p(f(\mathcal{M}), f(\mathcal{N})) \cdot \log \frac{p(f(\mathcal{M}), f(\mathcal{N}))}{p(f(\mathcal{M}))p(f(\mathcal{N}))}, \quad (3)$$

where \mathcal{X}, \mathcal{Y} are two original three-order tensors, \mathcal{M}, \mathcal{N} are small tensor blocks with dimensions $d = (2r+1) \times (2r+1) \times (2r+1)$ taken from tensor \mathcal{X}, \mathcal{Y} , r is the radius of the small tensor block and generally taken as 1. The function $f(x)$ maps the d -dimensional random variable represented by the d -dimensional small tensor block to the 1-dimensional space, this mapping includes the spatial information of the tensor. X, Y are the 1-dimensional random variables corresponding to the different 1-dimensional spaces, and these spaces are obtained by mapping small tensor blocks \mathcal{M} and \mathcal{N} through $f(x)$, respectively. In essence, this mapping can also be mapped to 2-dimensional space or multi-dimensional space, so the meaning of this function is to reduce the dimensionality and extract spatial information, i.e., the original high-dimensional probability distribution is reduced to a low-dimensional probability distribution accompanied by spatial information extraction.

When $r = 0$ and $f(x)$ is an identity map, the above tensor mutual information becomes conventional mutual information defined by Shannon entropy. That is to say, in the method of paper [28], image tensor is regarded as a random variable, and the mutual information between the two images is obtained by calculating the mutual information of random variables representing the two images. Because neural networks can approximate any continuous function on a compact set, the neural network + mutual information paradigm is essentially the tensor mutual information proposed in this paper.

Similar to the entropy corresponding to conventional mutual information, the formulas of tensor entropy corresponding to tensor mutual information are as follows.

$$H(\mathcal{X}) = - \sum_{\substack{\mathcal{M} \in \mathcal{X} \\ f(\mathcal{M}) \in X}} p_X(f(\mathcal{M})) \log p_X(f(\mathcal{M})), \quad (4)$$

$$H(\mathcal{Y}) = - \sum_{\substack{\mathcal{N} \in \mathcal{Y} \\ f(\mathcal{N}) \in Y}} p_Y(f(\mathcal{N})) \log p_Y(f(\mathcal{N})), \quad (5)$$

$$H(\mathcal{X}, \mathcal{Y}) = - \sum_{\substack{\mathcal{M} \in \mathcal{X} \\ f(\mathcal{M}) \in X}} \sum_{\substack{\mathcal{N} \in \mathcal{Y} \\ f(\mathcal{N}) \in Y}} p_{X,Y}(f(\mathcal{M}), f(\mathcal{N})) \cdot \log p_{X,Y}(f(\mathcal{M}), f(\mathcal{N})), \quad (6)$$

3.2 Properties of Tensor Mutual Information

For any two tensors \mathcal{X}, \mathcal{Y} , their mutual information $I(\mathcal{X}; \mathcal{Y})$ satisfies:

- (1) Symmetry: $I(\mathcal{X}; \mathcal{Y}) = I(\mathcal{Y}; \mathcal{X})$,
- (2) Positive semi-definiteness: $I(\mathcal{X}; \mathcal{Y}) \geq 0$, if and only if \mathcal{X}, \mathcal{Y} are independent, $I(\mathcal{X}; \mathcal{Y}) = 0$,
- (3) Extreme value: $I(\mathcal{X}; \mathcal{Y}) \leq H(\mathcal{X})$, $I(\mathcal{X}; \mathcal{Y}) \leq H(\mathcal{Y})$,
- (4) $I(\mathcal{X}; \mathcal{X}) = H(\mathcal{X})$.

Proof. From the definition of tensor mutual information and tensor entropy, it can be seen that (1) (3) (4) and $I(\mathcal{X}; \mathcal{Y}) > 0$ are obviously established. For the independence of tensors \mathcal{X}, \mathcal{Y} , we can know from Definition 1 that the multi-dimensional random variables constituting tensor \mathcal{X} and those constituting tensor \mathcal{Y} are independent. From Theorem 1, it can be seen that the 1-dimensional random variable X and Y are also independent after mapping the tensors \mathcal{X} and \mathcal{Y} through a continuous function $f(x)$. Therefore, at this time, the mutual information $I(\mathcal{X}; \mathcal{Y}) = 0$.

3.3 Concavity and Convexity of Average Tensor Mutual Information

According to the definition of tensor mutual information, the concept of average tensor mutual information is as follows:

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{\substack{\mathcal{M} \in \mathcal{X} \\ f(\mathcal{M}) \in X}} \sum_{\substack{\mathcal{N} \in \mathcal{Y} \\ f(\mathcal{N}) \in Y}} q(f(\mathcal{M})) \cdot p(f(\mathcal{N}) | f(\mathcal{M})) \log \frac{p(f(\mathcal{N}) | f(\mathcal{M}))}{\omega(f(\mathcal{N}))}. \quad (7)$$

where $\omega(y)$ is the edge density function of (X, Y) with respect to Y . From (21), we can see that $I(\mathcal{X}; \mathcal{Y})$ is a function of prior probability distribution $q(x)$ and transition probability distribution $p(y|x)$, which can be recorded as $I(\mathcal{X}; \mathcal{Y}) = h(q(x), p(y|x))$, where $x \in X, y \in Y$.

When conditional probability distribution $p(y|x)$ is given, average tensor mutual information $I(\mathcal{X}; \mathcal{Y})$ is an upward convex function of prior probability distribution $q(x)$. When prior probability distribution $q(x)$ is given, average tensor mutual information $I(\mathcal{X}; \mathcal{Y})$ is a downward convex function of conditional probability distribution $p(y|x)$.

4 Unsupervised Classification Applications

Currently, the intersection of neural networks and mutual information has garnered a plethora of research outcomes, as evidenced by the extensive literature spanning from [21]- [25]. This fertile ground offers ample opportunities for the application of our proposed tensor mutual information. Notably, the present

work eschews the necessity for supervised learning experiments to underscore the metric's superiority. Instead, we embark on an unsupervised image classification endeavor, aiming to illuminate the robustness and efficacy of our tensor mutual information measure in a more challenging, label-agnostic setting.

The classification method adopts the conventional k -means classification, using Euclidean distance (ED), Manhattan distance (MD) and conventional mutual information (MI) as similarity metrics for comparison with improved tensor mutual information based on the Pearson correlation coefficient (P-TMI). The unsupervised experiments are conducted on the MNIST and CIFAR-10 datasets. For MNIST dataset, it consists of 2D black and white images with dimensions of 28×28 . Difficult-to-distinguish digit images 3 and digit images 5 are selected for the binary classification experiments. Since the training samples for digit images 3 and 5 in the MNIST database are only 6131 and 5421 respectively, we randomly select 500 digit images of 3 and 500 digit images of 5 as one set of experimental samples. This process is repeated 10 times to determine 10 sets of experimental samples. When selecting the ten sets of experimental samples, efforts are made to avoid duplication. The above experimental methods are designed to avoid the occurrence of randomness. All experiments are repeated 1000 times and the maximum classification accuracy attained by each method is recorded.

The tensor mutual information distance between tensors \mathcal{X} and \mathcal{Y} is:

$$\begin{aligned} d_{P-TMI}(\mathcal{X}; \mathcal{Y}) &= 1 - NMI(\mathcal{X}; \mathcal{Y}) \\ &= 1 - 2 \frac{I(\mathcal{X}; \mathcal{Y})}{H(\mathcal{X}) + H(\mathcal{Y})}. \end{aligned} \quad (8)$$

Similarly, the conventional mutual information distance between tensors \mathcal{X} and \mathcal{Y} is:

$$\begin{aligned} d_{MI}(\mathcal{X}; \mathcal{Y}) &= 1 - NMI(X; Y) \\ &= 1 - 2 \frac{I(X; Y)}{H(X) + H(Y)}. \end{aligned} \quad (9)$$

where X and Y are 1-dimensional random variables corresponding to tensors \mathcal{X} and \mathcal{Y} , which are obtained by ignoring the spatial structure of the tensors \mathcal{X} and \mathcal{Y} . This can be understood as treating the tensor directly as a 1-dimensional random variable, and directly unfolding it into vectors to obtain X and Y .

The Euclidean distance between tensors \mathcal{X} and \mathcal{Y} is:

$$d_E(\mathcal{X}; \mathcal{Y}) = \left(\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} (\mathcal{X}_{ijk} - \mathcal{Y}_{ijk})^2 \right)^{\frac{1}{2}}. \quad (10)$$

The Manhattan distance between tensors \mathcal{X} and \mathcal{Y} is:

$$d_M(\mathcal{X}; \mathcal{Y}) = \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} |\mathcal{X}_{ijk} - \mathcal{Y}_{ijk}|. \quad (11)$$

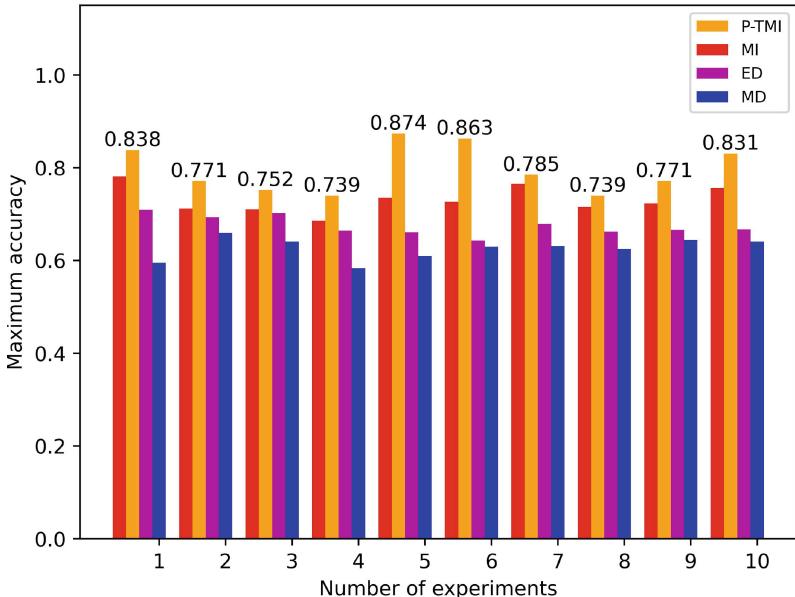


Fig. 1. Binary classification experiment results on digital image 3 and 5 (MNIST).

The visualization results in Fig. 1 illustrate the efficiency of iterations under different similarity metrics. P-TMI significantly outperforms the other three similarity metrics. Compared to ED, MD, and MI, P-TMI shows maximum improvements in maximum accuracy by 22%, 26.4%, and 13.9% respectively. The above results indicate that for images with clear and distinct features, P-TMI outperforms ED and MD. The reason is that P-TMI compares similarity from the perspective of image distribution, while ED and MD compare similarity based on pixel values, which leads to lower accuracy. Compared to MI, the high accuracy achieved by P-TMI also indicates that P-TMI can effectively utilize the spatial information of tensors.

In addition, binary classification experiments are conducted in the CIFAR-10 database, with trucks and birds selected as the image categories. The selection of these two categories is random. The situation is the same as faced by the MNIST dataset, in order to avoid the limitations of the dataset, we randomly divide all the data of truck images and bird images into ten sets for separate experiments and ultimately conduct the final experiment on the entire dataset. Since there are only 5000 training samples for each category in the CIFAR-10 database, we divide the mixed images of 10,000 trucks and birds into 10 sets of samples for conducting experiments. Each set of samples contains 500 truck images and 500 bird images. Since the k -means method uses random initialization, all experiments are repeated 1000 times to obtain the maximum classification accuracy. The classification accuracy results are shown in Table 1.

Table 1. Classification accuracy for binary classification experiments of truck and bird in the CIFAR 10 database.

Experiment #	ED	MD	MI	P-TMI
1	0.647	0.601	0.548	0.698
2	0.626	0.598	0.529	0.719
3	0.618	0.585	0.531	0.721
4	0.605	0.572	0.552	0.736
5	0.599	0.565	0.519	0.715
6	0.592	0.568	0.675	0.721
7	0.614	0.581	0.544	0.729
8	0.627	0.599	0.562	0.718
9	0.637	0.602	0.569	0.745
10	0.604	0.593	0.579	0.729

Table 1 shows that TMI is a similarity metric can significantly improve classification accuracy. Compared to ED, MD, and MI, P-TMI shows a maximum improvement of 13.1%, 16.4% and 19.6% in terms of maximum accuracy. In terms of overall average improvement in maximum accuracy, P-TMI outperforms ED, MD, and MI by 10.62%, 13.67% and 16.23% respectively.

5 Conclusions

This paper introduces a groundbreaking tensor-centric similarity metric: tensor mutual information. This novel approach not only encapsulates the essence of conventional mutual information but also offers profound insights into its exceptional performance within deep learning frameworks. By extending the dimensionality of conventional mutual information, tensor mutual information emerges as a natural evolution, retaining its predecessor's advantageous characteristics while introducing novel capabilities. Central to its design, tensor mutual information seamlessly integrates the spatial organization of tensors into the realm of mutual information, thereby eliminating the need for tensor vectorization. This integration not only preserves the rich structural information inherent in tensors but also enhances the metric's versatility and applicability. Furthermore, the continuity of the intrinsic function $f(x)$ within tensor mutual information renders it particularly suitable for unsupervised learning scenarios. This feature underscores its potential to unlock new avenues of research and applications where label information is scarce or unavailable, broadening the horizons of mutual information-based analysis in data-driven domains. The tensor mutual information in this paper essentially treats tensors as multi-dimensional distributions and maps these distributions into one-dimensional distributions through a continuous function $f(x)$. The final mutual information is then calculated using the obtained one-dimensional distribution. Compared to conventional mutual infor-

mation, the computational complexity of tensor mutual information is lower because the data dimensionality is significantly reduced.

There are still many areas worth exploring, as the following aspects:

(1) Spatial information is extracted from small tensor blocks through $f(x)$ in the form of multivariate linear combination, with the corresponding coefficient being the Pearson correlation coefficient. However, there are still many ways to extract spatial information from small tensor blocks, as long as $f(x)$ is a continuous function. For example, principal component analysis or gradient information can also be used.

(2) The d -dimensional small tensor block is reduced to a 1-dimensional random variable using $f(x)$, and then the mutual information is calculated. However, it is possible to extract spatial information from the d -dimensional small tensor blocks of each tensor using different $f(x)$. By applying different $f(x)$ to different tensors and then calculating the mutual information of the two sets of tensors, one can obtain the mutual information of the tensors \mathcal{X} and \mathcal{Y} .

(3) Applying the relevant knowledge of Tucker decomposition, one can obtain the low-rank approximation tensor from the original tensor. Then, calculate the tensor mutual information of the nuclear tensor (or low-rank approximation tensor) to obtain the mutual information of the original tensor.

(4) Tensor mutual information can be applied to practical application scenarios such as anomaly detection, image registration, epileptic seizure detection, and smart city applications (e.g., [29]) and can be integrated closely with neural networks and other machine learning algorithms (e.g., [30]).

References

1. Ohri, K., Kumar, M.: Review on self-supervised image recognition using deep neural networks. *Knowl. Based Syst.* **224**, 107090 (2021)
2. Yang, S., et al.: Asteria-pro: enhancing deep-learning based binary code similarity detection by incorporating domain knowledge. *ACM Trans. Softw. Eng. Methodol.* **33**, 1–40 (2023)
3. Lei, J., Rinaldo, A.: Consistency of spectral clustering in stochastic block models. *Ann. Stat.* **43**, 215–237 (2013). <https://api.semanticscholar.org/CorpusID:88519551>
4. Bai, X., Yan, C., Yang, H., Bai, L., Zhou, J., Hancock, E.: Adaptive hash retrieval with kernel based similarity. *Pattern Recognit.* **75**, 136–148 (2017)
5. Jain, S., Govindu, V.: Efficient higher-order clustering on the Grassmann manifold. In: IEEE International Conference on Computer Vision (ICCV), pp. 3511–3518 (2013)
6. Zhang, L., Guo, J., Wang, J., Wang, J., Li, S., Zhang, C.: Hypergraph and uncertain hypergraph representation learning theory and methods. *Mathematics* **10**, 1921 (2022)
7. Li, M., Zhang, Y., Li, X., Zhang, Y., Yin, B.: Hypergraph transformer neural networks. *ACM Trans. Knowl. Discov. Data* **17**, 1–22 (2022)
8. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 623–656 (1948). <https://api.semanticscholar.org/CorpusID:55379485>
9. Cover, T.M.: Elements of Information Theory (1999)

10. Li, T., Cai, Y., Zhang, Y., Cai, Z., Liu, X.: Deep mutual information subspace clustering network for hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–1 (2022)
11. Mirsadeghi, E., Royat, A., Rezatofighi, H.: Unsupervised image segmentation by mutual information maximization and adversarial regularization (2021)
12. Keuper, M., Brox, T.: Point-wise mutual information-based video segmentation with high temporal consistency. In: Hua, G., Jégou, H. (eds.) *ECCV 2016*. LNCS, vol. 9915, pp. 789–803. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_65
13. Sengupta, D., Gupta, P., Biswas, A.: A survey on mutual information based medical image registration algorithms. *Neurocomputing* **486**, 174–188 (2021)
14. Hassan, K., Islam, M., Nguyen, T., Molla, M.K.: Epileptic seizure detection in EEG using mutual information-based best individual feature selection. *Expert Syst. Appl.* **193**, 116414 (2022)
15. Souza, F., Premeida, C., Araújo, R.: High-order conditional mutual information maximization for dealing with high-order dependencies in feature selection (2022)
16. Conviv, I., Huggins, W., Liao, H., Whaley, K.: Mutual information scaling for tensor network machine learning. *Mach. Learn. Sci. Technol.* **3**, 015017 (2022)
17. Lei, X., Xia, Y., Wang, A., Jian, X., Zhong, H., Sun, L.: Mutual information based anomaly detection of monitoring data with attention mechanism and residual learning. *Mech. Syst. Signal Process.* **182**, 109607 (2023)
18. Dionisio, A., Menezes, R., Mendes, D.: Mutual information: a measure of dependency for nonlinear time series. *Physica A Stat. Mech Appl.* **344**, 326–329 (2004)
19. Sulistiani, H., Muludi, K., Syarif, A.: Implementation of dynamic mutual information and support vector machine for customer loyalty classification. *J. Phys: Conf. Ser.* **1338**, 012050 (2019)
20. Zhou, X., Wang, X., Dougherty, E., Russ, D., Suh, E.: Gene clustering based on clusterwide mutual information. *J. Comput. Biol. J. Comput. Mol. Cell Biol.* **11**, 147–61 (2004)
21. Guo, Y., Liu, B., Zhao, D.: Online continual learning through mutual information maximization. In: International Conference on Machine Learning (2022). <https://api.semanticscholar.org/CorpusID:250340833>
22. Do, K., Tran, T., Venkatesh, S.: Clustering by maximizing mutual information across views (2021)
23. Liu, Z., et al.: Temporal feature alignment and mutual information maximization for video-based human pose estimation (2022)
24. Kachouie, N., Shutaywi, M.: Weighted mutual information for aggregated kernel clustering. *Entropy* **22**, 351 (2020)
25. Aghagolzadeh, M., Soltanian-Zadeh, H., Araabi, B., Aghagolzadeh, A.: A hierarchical clustering based on mutual information maximization. In: IEEE International Conference on Image Processing (ICIP), pp. 277–280 (2007)
26. Goodman, N.R.: Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction). *Ann. Math. Stat.* **34**, 152–177 (1963)
27. Russakoff, D.B., Tomasi, C., Rohlfing, T., Maurer, C.R.: Image similarity using mutual information of regions. In: Pajdla, T., Matas, J. (eds.) *ECCV 2004*. LNCS, vol. 3023, pp. 596–607. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24672-5_47
28. Sun, L., Zhou, Y.: A key frame extraction method based on mutual information and image entropy. In: IEEE International Conference on Multimedia Technology (ICMT) (2011)

29. Sarker, S., Rakib, M.A., Islam, S., Shafin, S.S.: An IoT-based smart grid technology: bidirectional power flow, smart energy metering and home automation. In: 2021 International Conference on Maintenance and Intelligent Asset Management (ICMIAM), pp. 1–6, Australia. IEEE (2021)
30. Haque, M.S.M., Latif, G., Hasan, M.R., Arifuzzaman, M., Shafin, S.S., Rahman, Q.A.: Scalable parallel SVM on cloud clusters for large datasets classification. In: 2nd Smart Cities Symposium (SCS 2019), pp. 1–6. Bahrain. IET (2019)



Enforcing Specific Behaviours via Constrained DRL and Scenario-Based Programming

Davide Corsi¹ , Raz Yerushalmi^{2,3} , Guy Amir³ , Alessandro Farinelli⁴ , David Harel³ , and Guy Katz²

¹ University of California, Irvine, USA
`dcorsi@uci.edu`

² The Hebrew University of Jerusalem, Jerusalem, Israel
`raz.yerushalmi@weizmann.ac.il`, `guykatz@cs.huji.ac.il`

³ The Weizmann Institute of Science, Rehovot, Israel
`guyam@cs.huji.ac.il`, `david.harel@weizmann.ac.il`

⁴ University of Verona, Verona, Italy
`alessandro.farinelli@univr.it`

Abstract. Deep reinforcement learning (DRL) has achieved groundbreaking results in robotics, cyber-physical systems, healthcare, and many other real-world applications in recent years. However, despite their success, the inherent opacity and unpredictability of DRL controllers limit their widespread adoption in many safety-critical scenarios. In such contexts, it is crucial to consider additional safety and behavioral requirements pertaining to the deployed agents in addition to their performance. In this paper, we propose using Scenario-Based Programming (SBP) to define a cost signal that can be optimized together with the standard reward function to enforce additional behaviors in the final agents. To this end, we rely on the constrained DRL framework, particularly on a modified version of Lagrangian-PPO, which we call λ -PPO, designed especially for the multi-step and temporal nature of the SBP requirements. This approach allows us to easily design and enforce the agent’s adherence to these requirements during training without compromising its freedom to explore the state space and converge to an optimal

The work was partially funded by an NSFC-ISF grant to Harel, issued jointly by the National Natural Science Foundation of China (NSFC) and the Israel Science Foundation (ISF grant 3698/21), and in part by the Minerva grant 714132. Additional support was provided by a research grant to Harel from Louis J. Lavigne and Nancy Rothman, the Carter Chapman Shreve Family Foundation, Dr. and Mrs. Donald Rivin, and the Estate of Smigel Trust. The work of Yerushalmi, Amir and Katz was partially funded by the European Union (ERC, VeriDeL, 101112713). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The work of Amir was further supported by a scholarship from the Clore Israel Foundation.

D. Corsi and R. Yerushalmi—Contributed equally.

policy, enabling the use of a simple reward function. We have validated our method extensively by experimenting with real robotic platforms in a mapless navigation task, demonstrating the method’s success. We use SBP to define different types of requirements, including a more predictable behavior, safety properties, and the injection of prior knowledge to drive training.

1 Introduction

In recent years, *deep neural networks* (DNNs) have achieved state-of-the-art results in a large variety of tasks, including image recognition [11], game playing [37], protein folding [25], and more. In particular, *deep reinforcement learning* (DRL) [50] has emerged as a popular paradigm for training DNNs that perform complex tasks through continuous interaction with their environment. Indeed, DRL models have proven remarkably useful in robotic control tasks, such as navigation [29] and manipulation [8, 40], where they often outperform classical algorithms [61]. The success of DRL-based systems has naturally led to their integration as control policies in safety-critical tasks, such as autonomous driving [46], surgical assistance [42], flight control [28], and more. Consequently, the learning community has been seeking to create DRL-based controllers that demonstrate high *performance* and high *reliability*; i.e., they can perform their primary tasks while adhering to some prescribed safety properties and additional behavioral requirements.

An emerging family of approaches for achieving this goal, known as *Constrained DRL* (CDRL) [44], attempts to simultaneously optimize two functions: the *reward*, which encodes the main objective of the task, and the *cost*, which represents the constraints. Several approaches attempt to tackle this problem with different strategies, ranging from convex optimization [1] to carefully designed penalty functions [31, 58] and model selection [35]. In this paper, we focus on the Lagrangian dual relaxation of the constrained optimization problem, following a strategy similar to that used for PID-Lagrangian-PPO (LAG-PPO) [49] and RCPO [52]. However, despite their success in many applications, even state-of-the-art methods generally suffer from significant setbacks: (i) it is unclear how to translate such constraints into an effective signal for the training algorithm; (ii) there is no uniform human-readable way of defining the required constraints; and (iii) there is no clear understanding of the meaning of “cost” and how to balance cost and performance.

In this paper, we present a novel methodology for addressing these challenges by enabling domain experts to use formal language to define constraints relating to the agent’s behavior. Our proposed method pushes the optimization process towards an optimal solution for its primary task while also adhering to the different constraints. To achieve this goal, we extend and integrate two approaches: The λ -*PPO* algorithm that we describe in this paper for the actual training, and the *Scenario-Based Programming* (SBP) [10, 20] framework for encoding user-defined constraints. Scenario-based programming is a software engineering

paradigm that allows engineers to create a complex system that is aligned with how humans perceive that system. A scenario-based program is comprised of scenarios, each of which describes a single desirable (or undesirable) behavior of the system at hand; these scenarios are eventually combined to run simultaneously and produce cohesive system behavior.

In this work, we demonstrate how such scenarios can directly incorporate subject-matter-expert (SME) knowledge into the training process, thus forcing the resulting agent's behavior to abide by various safety, efficiency, and predictability requirements. Our methodology, however, raises many additional research questions, that we discuss in Sect. 4, such as the meaning of the cost function in this context and how to deal with the temporal nature of the SBP requirements.

In order to demonstrate the effectiveness of our approach on a real-world task, we apply it to train a policy for performing robotic *mapless navigation* [51, 57] on the popular Robotis Turtlebot3 platform. Although common DRL-training techniques were shown to give rise to high-performance policies for this task [33], these policies are often unsafe, inefficient, or unpredictable, thus dramatically limiting their potential deployment in real-world systems [34, 35]. In contrast, and as demonstrated in Sect. 5, our approach can generate trustworthy policies that are both safe and performant.

2 Preliminaries

Deep reinforcement learning (DRL) [30] is a popular paradigm for training deep neural networks [13]. In DRL, an agent learns to solve tasks by interacting with its environment through a trial-and-error process. The agent is driven only by a high-level objective, represented by a reward signal. Typically, a DRL problem is modeled as a Markov Decision Process (MDP), described by a tuple $\langle S, A, T, r \rangle$, where S is the state space, A is the actions space, $T : S \times A \rightarrow S$ is the transition function that encodes the probability $T(s_{t+1}|s_t, a_t)$ of transitioning from the state s_t to the next state s_{t+1} (given an action $a_t \in A$ at timestep t), and $r : S \times A \rightarrow \mathbb{R}$ is the reward signal. The objective of a standard DRL algorithm is to maximize the *expected reward* by finding a *policy*, denoted as π_θ , that maps an observed environment state s to action a .

In safety-critical tasks, the concept of optimality often goes beyond the maximization of a reward and can involve safety constraints that the agent should adhere to. A *constrained Markov decision process* (CMDP) is an alternative framework for sequential decision-making that includes these additional requirements. CMDP extends the standard MDP with an additional signal: the *cost function*, defined as $C : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, whose expected values must remain below a given threshold $d \in \mathbb{R}$. CMDP can support multiple cost functions (and their thresholds), denoted by $\{C_k\}$ and $\{d_k\}$, respectively. The set of *valid* policies for a CMDP is defined as:

$$\Pi_C := \{\pi_\theta \in \Pi : \forall k, J_{C_k}(\pi_\theta) \leq d_k\} \quad (2.1)$$

where $J_{C_k}(\pi_\theta)$ is the expected sum of the k^{th} cost function over the trajectory and d_k is the corresponding threshold. Intuitively, the objective is to find a policy function that respects the constraints (i.e., is *valid*) and which also maximizes the expected reward (i.e., is *optimal*).

A formal approach to encode constraints in a classical optimization problem is using *Lagrange multipliers* and relaxing the constrained problem into the corresponding dual unconstrained version [1, 31]. The optimization problem can then be encoded as follows:

$$J(\theta) = \min_{\pi_\theta} \max_{\lambda \geq 0} \mathcal{L}(\pi_\theta, \lambda) = \min_{\pi_\theta} \max_{\lambda \geq 0} J_R(\pi_\theta) - \sum_K \lambda_k (J_{C_k}(\pi_\theta) - d_k) \quad (2.2)$$

Equation 2.2 encodes both the reward signal and the constraint monitors for the optimizer.

2.1 Scenario-Based Programming

Scenario-based programming (SBP) [10, 18] is a paradigm designed to facilitate the development of reactive systems, by allowing engineers to program a system in a way that is close to how humans perceive it. In SBP, a system is composed of *scenarios*, each describing a single aspect of system behavior, either desired or undesired; and these scenarios are then executed in unison as a cohesive system. Execution of a scenario-based (SB) program is formalized as a discrete sequence of events. At each time step, all the scenarios are synchronized to determine the next event to be triggered. Each scenario declares events that it *requests* and events that it *blocks*, corresponding to desirable and undesirable (forbidden) behaviors from its perspective; and also events that it passively *waits-for*. After making these declarations, the scenarios are temporarily suspended, and an *event-selection mechanism* triggers a single event that was requested by at least one scenario and blocked by none. Scenarios that request or wait for the triggered event wake up, perform local actions, and then synchronize again; and the process is repeated ad infinitum. The resulting execution thus complies with the requirements and constraints of each of the individual scenarios [18, 20]. For a formal definition of SBP, we refer to the work of Harel et al. [20]. Finally, in Appendix A, we show a concrete example of a Scenario Based Program for the control of the temperature and water level in a water tank, inspired by the work of Harel et al. [19, 22].

3 Case Study: Mapless Navigation

As a running example, we explain and demonstrate our proposed technique on the *mapless navigation* problem, in which a robot is required to reach a given target efficiently while avoiding collision with obstacles. Unlike in classical planning, the robot can rely only on local observations—e.g., from lidar sensors or cameras, and can not access a map of the surrounding environment. Thus, a successful agent needs to be able to adjust its strategy dynamically as it progresses

toward its target. Mapless navigation has been studied extensively and is considered difficult to solve. Specifically, the local and partially observable nature of the problem renders learning a successful policy extremely challenging and hard to solve using classical algorithms [41]. Prior work has shown DRL to be among the most successful approaches for tackling this task, often outperforming hand-crafted algorithms [33]. As a platform for our study, we rely on the *Robotis Turtlebot 3* platform (Turtlebot, for short; see Fig. 1), which is widely used in the community [5, 38].

3.1 The Primary Reward Function

The Turtlebot is capable of horizontal navigation and is equipped with lidar sensors to detect nearby obstacles. In order to train DRL policies for controlling the Turtlebot, we built a simulator based on the *Unity3D* engine [24], which is compatible with the *Robotic Operating System* (ROS) [43] and streamlines deploying the trained agent on the actual platform (*sim-to-real* [60]). We designed a hybrid reward function, which includes a discrete component for the terminal states (“collision” or “target reached”) and a continuous component for the non-terminal states. Formally:

$$R_t = \begin{cases} \pm 1 & \text{terminal states} \\ (dist_{t-1} - dist_t) \cdot \eta - \beta & \text{otherwise} \end{cases} \quad (3.1)$$

where $dist_k$ is the distance from the target at time k ; η is a normalization factor; and β is a penalty, intended to encourage the robot to reach the target quickly (in our experiments, we empirically set $\eta = 3$ and $\beta = 0.001$). Additionally, in terminal states, we increase the reward by 1 if the target is reached, or decrease it by 1 in case of collision. The DNN topology we used is an architecture that was shown to be successful in similar settings [4, 33]: (i) an input layer of nine neurons, including seven for the lidar scans and two for the polar coordinates of the target; (ii) two fully connected hidden layers of 32 neurons each; and (iii) an output layer of three neurons for the discrete actions (i.e., move FORWARD, turn LEFT, and turn RIGHT).



Fig. 1. The Robotis Turtlebot3 platform.

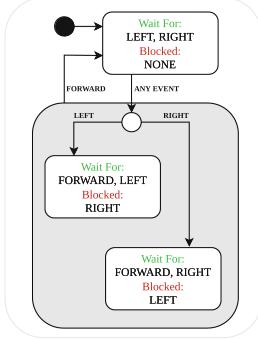


Fig. 2. A visualization of the first scenario: *avoid back-and-forth rotation*

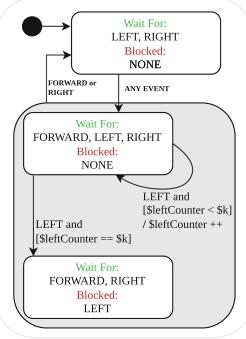


Fig. 3. A visualization related to the second scenario: *avoid left turns larger than 180°*

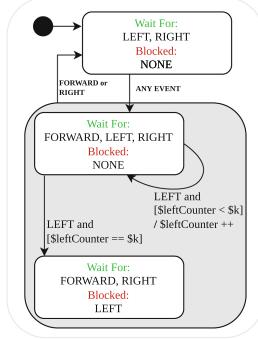


Fig. 4. A visualization of the third scenario: *avoid turning when clear*

Using these reward function and network topology, we were able to train agents that achieved high performance—i.e., obtained a success rate of approximately 95%, where “success” means that the robot reached its target without colliding with walls or obstacles. Analyzing the trained agents further, we observed that even DRL agents that achieved a high success rate might still demonstrate undesirable behavior in different scenarios. One such behavior is a sequence of back-and-forth turns, which causes the robot to waste time and energy. Another undesirable behavior is when the agent makes a lengthy sequence of right turns instead of a much shorter sequence of left turns (or vice versa), again wasting time and energy. A third undesirable behavior that we observed is that the agent might decide not to move forward toward a target that is directly ahead, even when the path is clear. Our goal was thus to use our approach and demonstrate how we can remove these undesirable behaviors without altering the defined reward function (Eq. 3.1).

3.2 A Rule-Based Approach

We propose integrating a scenario-based program into the DRL training process to suppress undesired behaviors without changing the primary reward function (and potentially compromising the learning process). More concretely, we propose to create specific scenarios to rule out each of the undesirable behaviors mentioned above. This is achieved by creating a mapping between each possible action $a_t \in [\text{Move FORWARD}, \text{Turn LEFT}, \text{Turn RIGHT}]$ of the DRL agent and a dedicated event $e_{a_t} \in [\text{SBP_MoveForward}, \text{SBP_TurnLeft}, \text{SBP_TurnRight}]$ within the scenario-based program. These events allow the various scenarios to keep track of and react to the agent’s actions. Similarly to the work of Yerushalmi et al. [56], we refer to these e_{a_t} events as *external events*, indicating that they

can only be triggered when requested from outside the SB program proper. By convention, we assume that after each triggering of a single external event, the scenario-based program executes a sequence of internal events (*super-step*) until it returns to a steady state and then waits for another external event.

Considering our running example (i.e., the Turtlebot mapless navigation case study) again, we create scenarios for discouraging the three undesirable behaviors we had previously observed. Scenario *avoid back-and-forth rotation* (Fig. 2) seeks to prevent in-place, back-and-forth turns by the robot. Scenario *avoid turns larger than 180°* (Fig. 3) seeks to prevent left turns in angles that are greater than 180° (the right-turn case is symmetrical). A forward slash indicates an action that is performed when a transition is taken; square brackets denote guard conditions, and \$k and \$leftCounter are variables. Each turn rotates the robot by 30°, so we set $k = 7$. Scenario *avoid turning when clear* (Fig. 4) seeks to force the agent to move toward the target when it is ahead and there is a clear path to it. This is performed by blocking any turn actions when this situation occurs. Triggered events carry data, which can be referenced by guard conditions.

4 λ -PPO: From Scenarios to Constrained DRL

In order to integrate the requirements expressed via SBP in a CDRL training process, we define the cost function to correspond to violations of scenario constraints. Intuitively, whenever the agent selects an action that is mapped to a *blocked* SBP event (during training), we should increase the *cost*. However, obtaining a differentiable function for the training process, which is also trajectory-dependent, is not straightforward. To this end, we propose the following binary (indicator) function:

$$c_k(s_t, a, s_{t+1}) = I(\text{the tuple } \langle s_t, a, s_{t+1} \rangle \text{ is blocked by the } k^{\text{th}} \text{ rule}) \quad (4.1)$$

Intuitively, summing the values of c_k over a training episode yields the number of violations to the k^{th} scenario rule during a single trajectory; this value, if normalized over the number of steps, can be seen as the probability of having a violation during an episode. Crucially, the threshold can be interpreted as a bound for the probability of violating a requirement.

Interpretation of the Cost Function. Before introducing the main optimizations that constitute λ -PPO, it is essential to clarify how we interpret the cost function for the evaluation. Following the insights presented in the large-scale analysis by Corsi et al. [9], we believe that generating a policy that achieves an actual “zero cost” is almost impossible. For this reason, in this paper, we treat the cost function as a budget function, and our goal is to minimize the expected probability of encountering unwanted situations above the given required threshold rather than completely avoiding these behaviors.

4.1 Optimized Lagrangian-PPO

In Sect. 2 we proposed to relax the Lagrangian constrained optimization problem into an unconstrained, *min-max* version thereof. By taking the gradient of

Eq. 2.2 and applying some algebraic manipulation, we derive the following two simultaneous problems:

$$\nabla_{\theta} \mathcal{L}(\pi, \lambda) = \nabla_{\theta}(J_R(\pi) - \sum_K \lambda_k J_{C_k}(\pi)) \quad (4.2)$$

$$\forall k, \quad \nabla_{\lambda_k} \mathcal{L}(\pi, \lambda) = -(J_{C_k}(\pi) - d_k) \quad (4.3)$$

However, the naïve application of this approach has shown strong instability and the proclivity to optimize only the cost in our experiments, limiting the exploration and resulting in poorly performing agents. To overcome these problems, we introduce two key optimizations that proved crucial in obtaining the results we present in the next section and a set of improvements to stabilize the training process.

Reward Multiplier. The standard update rule for the policy in a Lagrangian method is given in Eq. 4.2. However, as mentioned above, it often fails to maximize the reward. To overcome this failure, we introduce a new parameter α , which we term *reward multiplier*, such that $\alpha \geq \sum_K \lambda_k$. This parameter is used as a multiplier for the reward objective:

$$\nabla_{\theta} \mathcal{L}(\pi, \lambda) = \nabla_{\theta}(\alpha \cdot J_R(\pi) - \sum_K \lambda_k J_{C_k}(\pi)) \quad (4.4)$$

Lambda Bounds and Normalization. Theoretically, the only constraint on the Lagrangian multipliers is that they are non-negative. However, when solving numerically, the value of λ_k can increase quickly during the early stages of the training, causing the optimizer to focus primarily on the cost functions (Eq. 4.2), potentially not pushing the policy towards high-performance regions. To overcome this, we introduced dynamic constraints on the multipliers (including the reward multiplier α), such that $\sum_K \lambda_k + \alpha = 1$. In order to also enforce the previously mentioned upper bound for α , we clipped the values of the multipliers such that $\sum_K \lambda_k \leq \frac{1}{2}$. Formally, we perform the following normalization over all the multipliers:

$$\forall k, \quad \lambda_k = \frac{\tilde{\lambda}_k}{2(\sum_K \tilde{\lambda}_k)} \quad \alpha = 1 - \sum_K \lambda_k \quad (4.5)$$

4.2 State Space Expansion and Convergence

In the previous sections, we explained the benefits of using SBP for encoding trajectory-based properties to effectively characterize intricate behaviors and their combinations. To achieve this objective, it is crucial to combine actions, states, and connections between them in a time-dependent manner. For example, if we need to encode the behavior of never turning left four times in a row, we can use a scenario-based property. However, optimizing the cost function generated by this scenario without caution may result in violating the properties of the

Markov Decision Process (MDP), where transitions between states should only depend on the previous state and action. We leave it for future work to address that tension in order to allow the use of arbitrary SBP behaviors.

We note that our approach builds on well-established algorithms such as PPO and other constrained reinforcement learning methods (e.g., CPO, LAG-PPO), which are known to have theoretical convergence guarantees under certain conditions, as discussed in previous work [1, 44, 47]. However, since our method relies on gradient-based optimization, the convergence guarantees are inherently limited by the stochasticity involved in the training process. Therefore, while our approach inherits the general convergence properties of the underlying framework, the practical learning process may vary during stochastic execution, especially given the complex, multi-step nature of our constraints.

5 Evaluation

We performed the training on a distributed cluster of HP EliteDesk machines running at 3.00 GHz with 32 GB RAM. We collected data from more than 100 seeds for each algorithm, reporting the mean and standard deviation for each learning curve, following the guidelines for a fair comparison of Colas et al. [7].

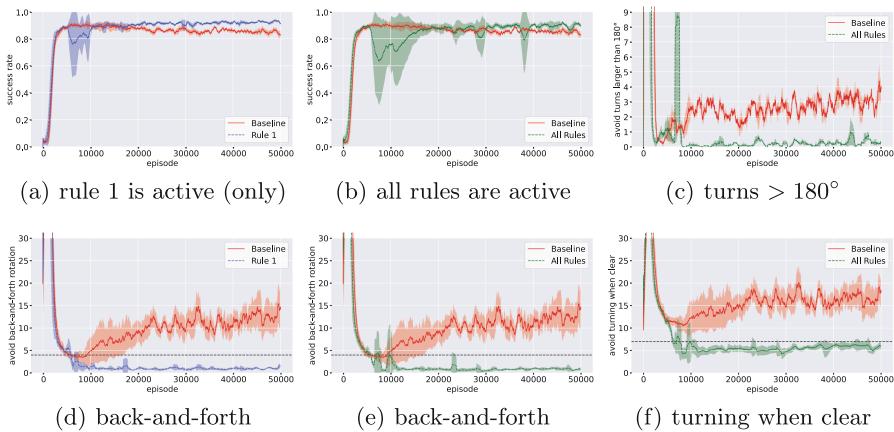


Fig. 5. A comparison between the baseline policies to policies trained using our approach. The black dotted line states the threshold (d_k) we considered for the k^{th} rule.

Figure 5 depicts a detailed comparison between policies trained with a standard end-to-end PPO [47] (the baseline), and those trained using our constrained method with the injection of rules. In particular, (a) and (d) show the results of policies trained with only *avoid back-and-forth rotation* added as a constraint, while (b), (c), (e), and (f) show the result with all the rules active.

Subfigure (a) shows that the success rate of the baseline stabilizes at around 87%, while the success rate of our improved policies stabilizes at around 95%. Subfigure (d) compares the frequency of undesired behavior occurrences between the baseline and our policies, where the frequency diminishes *almost completely*. Subfigures (c), (e), and (f) compare the frequency of the occurrence of undesired behaviors between the baseline and the policies trained with all rules active. Using the baseline, the frequencies of the three behaviors are respectively around 13, 3, and 17 per episode. The undesired behaviors are removed *almost completely* for the policies trained with our approach. We note that the undesired behavior addressed by the rule *avoid turns larger than 180°* is quite rare in general; and so the statistics reported in (c) were collected over the final 100 training episodes.

The results clearly show that our method is able to train agents that respect the given constraints without damaging the main training objective—the success rate. Moreover, it also highlights the scalability of our method, i.e., performing well when single or multiple rules are applied. Reviewing Fig. 5(b), comparing the baseline’s success rate with our method’s success rate when all rules are applied together with all the optimizations presented in Sect. 4 shows a clear advantage.

Moreover, our approach even led to an improved success rate, suggesting that the contribution of expert knowledge can drive the training to better policies.

5.1 Comparison with Naïve Reward-Engineering Approach

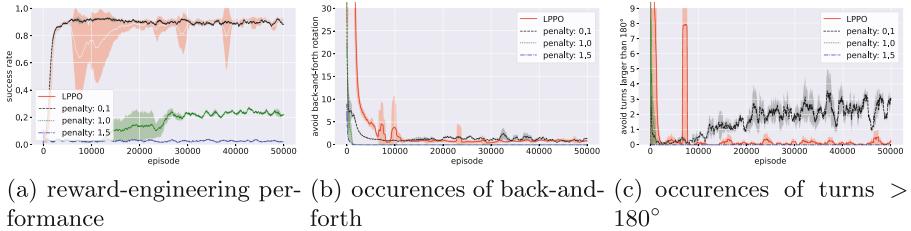


Fig. 6. A comparison between results achieved by our approach, denoted by *LPPO* (red line), with those achieved with a reward-engineering approach. (Color figure online)

To further validate our approach, we performed a comparison against a reward-shaping approach that penalizes the undesired behavior at each occurrence. Figure 6 shows a set of experiments with different reward penalties (e.g., 0.1, 1.0, or 1.5) compared to our approach (the red line). These results highlight the challenge of finding the correct balance for the penalty, as it necessitates non-trivial parameter tuning. In contrast, our constrained approach offers easier tuning and guarantees a more general and flexible solution for such problems. This finding is consistent with the conclusions of Kamran et al. [26] and Roy et al. [45] in their recent studies on the subject.

In Yerushalmi et al. [56], the authors proposed an integration between SBP and DRL using a reward-shaping approach that penalizes the agent via unconstrained optimization when rules are violated. Our approach, based on constrained optimization, provides many advantages compared to the aforementioned work, which results in high-performing agents and fewer rule violations.

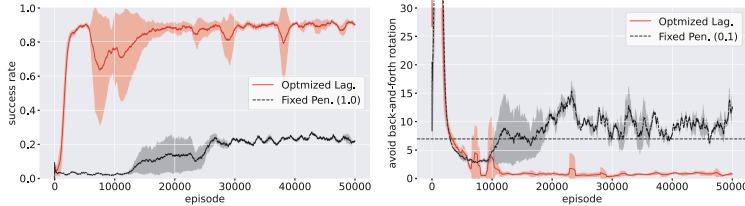


Fig. 7. Comparison between our approach and a reward-shaping method.

In Fig. 7, we provide a further comparison between the results of our approach, labeled ‘Optimized Lag.’, and that of Yerushalmi et al. [56], labeled ‘Fixed Pen.’. When using the ‘Fixed Pen.’ approach with a low penalty value that allows the agent to reach high-performance reward-wise (as seen in Fig. 6(a)), it fails to minimize the cost (e.g., the number of rule violations). In contrast, using a high penalty value reduces the agent’s rule violations but fails to achieve adequate performance in terms of the reward function. Using our approach (‘Optimized Lag.’), the agent was able to achieve similar performances as the best of the ‘Fixed Pen.’ when using a penalty value of 0.1 and reducing the agent’s rule violations as the best of ‘Fixed Pen.’ when using a penalty value of 1.0.

Our approach adopts a constraint-driven DRL framework that differentiates between optimizing the main reward and minimizing the costs. This differentiation affords significant advantages:

1. it allows setting constraint thresholds independently for each rule/property and handling multiple such constraints, unlike reward-shaping methods that only allow a global minimization to zero of the total cost.
2. it separates reward maximization from cost minimization, simplifying the reward engineering task.
3. it automatically balances the focus of the training between the different cost elements and the reward by learning the values of the different multipliers for each cost factor.
4. it introduces novel numerical optimizations to the training phase, resulting in a more stable algorithm and a higher cumulative reward.

Additional results can be found in Appendix B, where we analyze the trained policies using recent results from the field of *neural network verification* to formally show how our approach can improve the reliability of the trained agents.

5.2 Discussion and Limitations

While our method shows strong performance in the scenarios we evaluated, we recognize that there might be limitations to some CMDPs. One potential limitation is CMDPs with highly complex or conflicting constraints, where the cost functions may create trade-offs with rewards that are difficult to optimize. In such cases, finding a feasible policy that satisfies all constraints while optimizing for performance can lead to suboptimal exploration and convergence behavior. This could result in overly conservative policies where the agent prioritizes constraint satisfaction over the reward function, thereby limiting its ability to find high-performing policies. Another challenge may arise in environments with dynamic or non-stationary constraints. Since our approach assumes that constraints remain consistent during training, sudden changes in constraints or their thresholds could disrupt the learning process, requiring the algorithm to adapt to new conditions, which is not explicitly handled by our current method. Finally, our method, like other gradient-based algorithms, may struggle in CMDPs with sparse rewards or constraints that are rarely violated. In such scenarios, the agent may not effectively receive sufficient feedback to balance constraint satisfaction with reward maximization.

6 Related Work

To the best of our knowledge, this is the first work that combines scenario-based programming into training a constrained deep reinforcement learning system—specifically, in a robotic environment. The most closely related prior work is the study by Yerushalmi et al. [56], which proposed the integration of SBP and DRL using a reward-engineering approach. However, this approach inherits the limitations of incorporating a fixed penalty into the reward function. In another recent work on constrained reinforcement learning [45], the authors advocate an optimized version of Lagrangian-PPO. They propose a different approach to balance the constraints and the return based on the softmax activation function without imposing bounds on the values for the multipliers. Additionally, they did not employ a framework specifically designed for constraint encoding, such as SBP. Moreover, previous works focused on game development and synthetic environments, which differ from our robotic domains and present distinct challenges, such as safety and efficiency, are not considered crucial requirements.

In this paper, we adopted Lagrangian PPO as the basic building block for our algorithm. However, it is important to note that adopting SBP as a framework to design the requirements is agnostic to the optimization algorithm of choice. Consequently, we believe that alternative approaches to solving a CMDP should be explored—e.g., the alternative family of algorithms that rely on convex optimization, such as CPO [1], CUP [55], or FOCOPS [59].

An alternative family of approaches to guarantee safety and additional requirements exploits *safety shields*. Approaches from this family try to enforce the constraints via hardcoded shielding methods [48, 53]. Although these approaches guarantee the respect of the requirements by construction, they rely

heavily on prior knowledge and often restrict the agent’s ability to learn original strategies for problem-solving. In the case-study work from Kamran et al. [26], the authors show that restricting the search space often produces over-conservative behavior that can potentially lead to a stalemate of the system.

7 Conclusion

In this paper, we introduce a novel and generic approach that directly incorporates subject-matter-expert knowledge into the DRL learning process. This approach is significant as it enables us to achieve user-defined safety properties and behavioral requirements. We demonstrate how to encode the desired behavior as constraints for the DRL algorithm and enhance a state-of-the-art algorithm with various optimizations. Importantly, we define properties comprehensibly, leveraging scenario-based programming to encode them into the training loop. We apply our method to a real-world robotic problem, specifically mapless navigation, and show that our method can produce policies that respect all the constraints without adversely affecting the main objective of the optimization.

Future work will focus on exploring the scalability of our approach, particularly in systems with a large number of constraints. Although constrained-based methods typically scale better than reward-shaping approaches [45], further investigation is needed to fully understand the computational and performance trade-offs in large-scale environments. Moreover, we plan to extend our work to different robotics environments, including navigation in more complex domains (e.g., air and water), manipulation (e.g., grasping), and medical applications, where safety is a crucial requirement.

A Scenario-Based Programming: Concrete Example

Although SBP is implemented in many high-level languages (e.g., [15, 17, 21]), it is often convenient to think of scenarios as transition systems, where each state corresponds to a synchronization point, and each edge corresponds to an event that could be triggered. Figure 8 uses that representation to depict a simple SB program that controls the temperature and water level in a water tank (borrowed from [19]). The scenarios *add hot water* and *add cold water* repeatedly wait for a WATER LOW event, and then request three times the event Add HOT or Add COLD, respectively. Since these six events may be triggered in any order by the event selection mechanism, a new scenario, *stability*, is added to keep the water temperature stable by alternately blocking Add HOT and Add COLD events. The resulting execution trace is shown in the event log.

SBP is an attractive choice for the incorporation of domain-expert knowledge that expresses safety-related (and other) constraints into a DRL agent training process due to its support of the event-blocking idiom, as well as by being formally defined and fully executable [2, 14]. SBP’s formal semantics contribute to the formal verification of the constraints model. However, the approach presented in this paper is not specific to SBP. It can be easily extended to use

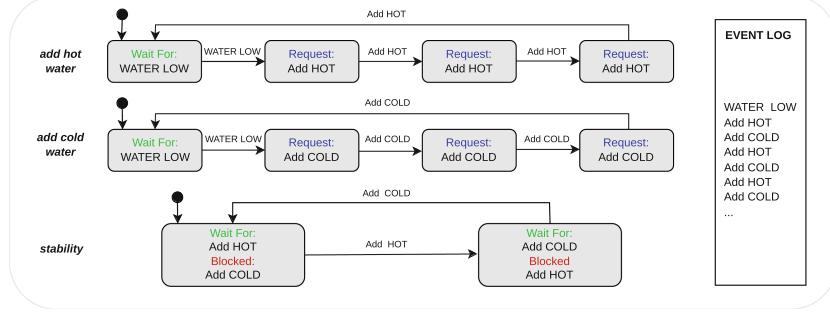


Fig. 8. A scenario-based program for controlling a water tank. The small black circle indicates the initial state.

other frameworks and methods to express safety-related and other constraints, as long as these enable constraint violation monitoring during the agent learning process, such as in the works of Harel [16] and Marron et al. [36], as well as decision trees [39].

B Deep Neural Networks and Formal Verification of DNNs

A deep neural network (DNN) [13] is a computational, directed graph that includes various types of layers—in which each layer includes various nodes (“neurons”). After receiving values through the first layer (the “input” layer), the network propagates the values through the computational layers (the “hidden” layers), until reaching the network’s final layer (the “output” layer).

The outputs generated by the final layer can either be a classification label or a regression value, depending on the DNN in question and its training process. In each hidden layer, the computation is based on the *type* of activation characterizing the neurons of each layer. For example, in the common *ReLU* (*rectified linear unit*) layer, each neuron y calculates the value $y = \text{ReLU}(x) = \max(x, 0)$, for a value x of the matching neuron in the preceding layer. Additional layer types include weighted-sum layers and various layers with non-linear activations (such as *sign* activations, *max-pooling*, and others). Here, we focus on *feed-forward* DNNs, i.e., networks in which each layer is connected exclusively to its subsequent layer. An example of a toy, feed-forward DNN, appears in Fig. 9.

A DNN verification algorithm receives the following inputs [27]: a trained DNN N , a precondition P on the DNN’s inputs, and a postcondition Q on N ’s output. The precondition is used to limit the input assignments to inputs of

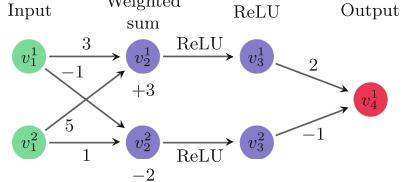


Fig. 9. A toy DNN.

Table 1. Results of the formal verification queries over 120 trained DNNs for each of the three properties in question. The first row shows the results of the 60 baseline policies, and the second row shows the results of the 60 policies trained by our method, with all rules active.

	back-and-forth rotation			turns larger than 180°			turning when clear		
ALGO	SAT	UNSAT	TIMEOUT	SAT	UNSAT	TIMEOUT	SAT	UNSAT	TIMEOUT
Baseline	60	0	0	51	0	9	60	0	0
SBP	22	38	0	0	41	19	9	34	17

interest or to express some assumption the user has regarding the environment (e.g., that an image-recognition DNN will only be presented with certain pixel values). The postcondition typically encodes the *negation* of the behavior we would like N to exhibit on inputs that satisfy P . Then, the verification algorithm searches for an input x' that satisfies the given conditions (i.e., $P(x') \wedge Q(N(x'))$), and returns exactly one of the following outputs:

1. SAT, indicating the query is satisfiable. Due to the postcondition Q encoding the negation of the required property, this result indicates that the desired property is violated in some cases. Modern verification engines also supply a concrete input x' that satisfies the query, and hence, a valid input that triggers a bug, such as an incorrect classification; or
2. UNSAT, indicating that there does not exist such an x' , and thus—that the desired property always holds.

For example, suppose we wish to guarantee that for all non-negative inputs $x = \langle v_1^1, v_1^2 \rangle$, the DNN in Fig. 9 always outputs a value strictly smaller than 40; i.e., that that $N(x) = v_4^1 < 40$. This property can be encoded as a verification query consisting of a precondition that restricts the inputs to the desired range, i.e., $P = (v_1^1 \geq 0) \wedge (v_1^2 \geq 0)$, and by setting $Q = (v_4^1 \geq 40)$, which is the negation of the desired property. In this case, a sound verifier will return SAT, alongside a feasible counterexample such as $x = \langle 2, 3 \rangle$, which produces the output $v_4^1 = 48 \geq 40$ when fed to the DNN. Hence, the property does not always hold. Originally, DNN verification engines were designed to verify the correct behavior of feed-forward DNNs [23, 27, 32, 54]. However, in recent years, the verification community has also designed verification methods tailored for DRL systems [3, 4, 6, 8, 12]. These methods include techniques for encoding multiple invocations of the agent in question when interacting with a reactive environment over multiple time-steps.

Results

As an additional means of proving the effectiveness of our method, we ran formal verification queries relating to the aforementioned undesirable behaviors. In order to conduct a fair comparison, we selected only models that passed our

success cutoff value (85%); and for each of these models we ran three verification queries—each checking whether the model violates a given property (**SAT**), or abides by it for all inputs (**UNSAT**). A verifier might also fail to terminate due to **TIMEOUT** or **MEMOUT** errors. Each query ran with a **TIMEOUT** value of 36 h, and a **MEMOUT** value of 6 GB. Table 1 summarizes the results of our experiments. These results show a *significant* change of behavior between DNNs trained with the baseline algorithm and those trained by our method. Indeed, the latter policies more often abide entirely by the specific rules and are consequently far more reliable.

References

1. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning (ICML) (2017)
2. Alexandron, G., Armoni, M., Gordon, M., Harel, D.: Scenario-based programming: reducing the cognitive load, fostering abstract thinking. In: Proceedings of the 36th International Conference on Software Engineering (ICSE) (2014)
3. Amir, G., Schapira, M., Katz, G.: Towards scalable verification of deep reinforcement learning. In: Proceedings of the 21st International Conference on Formal Methods in Computer-Aided Design (FMCAD), pp. 193–203 (2021)
4. Amir, G., Corsi, D., Yerushalmi, R., Marzari, L., Harel, D., Farinelli, A., Katz, G.: Verifying learning-based robotic navigation systems. In: Proceedings of the 29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), pp. 607–627 (2023)
5. Amsters, R., Slaets, P.: Turtlebot 3 as a robotics education platform. In: Proceedings of the 10th International Conference on Robotics in Education (RiE), pp. 170–181 (2019)
6. Bacci, E., Giacobbe, M., Parker, D.: Verifying reinforcement learning up to infinity. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), pp. 2154–2160 (2021)
7. Colas, C., Signal, O., Oudeyer, P.-Y.: A hitchhiker’s guide to statistical comparisons of reinforcement learning algorithms (2019). Technical Report. <http://arxiv.org/abs/1904.06979>
8. Corsi, D., Marchesini, E., Farinelli, A.: Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In: Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI) (2021)
9. Corsi, D., Amir, G., Katz, G., Farinelli, A.: Analyzing adversarial inputs in deep reinforcement learning (2024). Technical Report. <https://arxiv.org/abs/2402.05284>
10. Damm, W., Harel, D.: LSCs: breathing life into message sequence charts. J. Formal Methods Syst. Des. (FMSD) **19**(1), 45–80 (2001)
11. Du, J.: Understanding of object detection based on CNN family and YOLO. In: Proceedings of the 2nd International Conference on Machine Vision and Information Technology (CMVIT), pp. 23–25 (2018)
12. Eliyahu, T., Kazak, Y., Katz, G., Schapira, M.: Verifying learning-augmented systems. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM) (2021)

13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
14. Gordon, M., Marron, A., Meerbaum-Salant, O.: Spaghetti for the main course? Observations on the Naturalness of Scenario-Based Programming. In: Proceedings of the 17th ACM Conference on Innovation and Technology in Computer Science Education (ITCSE), pp. 198–203 (2012)
15. Greenyer, J., Gritzner, D., Katz, G., Marron, A.: Scenario-based modeling and synthesis for reactive systems with dynamic system structure in ScenarioTools. In: Proceedings of the 19th ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), pp. 16–23 (2016)
16. Harel, D.: STATECHARTS: a visual formalism for complex systems. *Sci. Comput. Program.* **8**(3), 231–274 (1987)
17. Harel, D., Katz, G.: Scaling-up behavioral programming: steps from basic principles to application architectures. In: Proceedings of the 4th SPLASH Workshop on Programming based on Actors, Agents and Decentralized Control (AGERE!), pp. 95–108 (2014)
18. Harel, D., Marely, R.: Come, Let's Play: Scenario-Based Programming using LSCs and the Play-Engine. Springer Science & Business Media, Heidelberg (2003)
19. Harel, D., Katz, G., Marron, A., Weiss, G.: Non-intrusive repair of reactive programs. In: Proceedings of the 17th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 3–12 (2012a)
20. Harel, D., Marron, A., Weiss, G.: Behavioral programming. *Commun. ACM (CACM)* **55**(7), 90–100 (2012)
21. Harel, D., Kantor, A., Katz, G.: Relaxing synchronization constraints in behavioral programs. In: McMillan, K., Middeldorp, A., Voronkov, A. (eds.) LPAR 2013. LNCS, vol. 8312, pp. 355–372. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45221-5_25
22. Harel, D., Katz, G., Marron, A., Weiss, G.: Non-intrusive repair of safety and liveness violations in reactive programs. *Trans. Comput. Collective Intell. (TCCI)* **16**, 1–33 (2014)
23. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: Majumdar, R., Kunčak, V. (eds.) CAV 2017. LNCS, vol. 10426, pp. 3–29. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63387-9_1
24. Juliani, A., et al. Unity: a general platform for intelligent Agents (2018). Technical Report. <https://arxiv.org/abs/1809.02627>
25. Jumper, J., et al.: Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021)
26. Kamran, D., et al.: A modern perspective on safe automated driving for different traffic dynamics using constrained reinforcement learning. In: Proceedings of the IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), (2022)
27. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: ReluPlex: an efficient SMT solver for verifying deep neural networks. In: Majumdar, R., Kunčak, V. (eds.) CAV 2017. LNCS, vol. 10426, pp. 97–117. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63387-9_5
28. Koch, W., Mancuso, R., West, R., Bestavros, A.: Reinforcement learning for UAV attitude control. *ACM Trans. Cyber-Phys. Syst.* **3**, 1–21 (2019)
29. Kulhánek, J., Derner, E., De Bruin, T., Babuška, R.: Vision-based navigation using deep reinforcement learning. In: Proceedings of the European Conference on Mobile Robots (ECMR) (2019)
30. Li, Y.: Deep reinforcement learning: an overview (2017). Technical Report. <http://arxiv.org/abs/1701.07274>

31. Liu, Y., Ding, J., Liu, X.: IPO: interior-point policy optimization under constraints. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI), pp. 4940–4947 (2020)
32. Lyu, Z., Ko, C.Y., Kong, Z., Wong, N., Lin, D., Daniel, L.: Fastened crown: tightened neural network robustness certificates. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI), pp. 5037–5044 (2020)
33. Marchesini, E., Farinelli, A.: Discrete deep reinforcement learning for mapless navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 10688–10694 (2020)
34. Marchesini, E., Corsi, D., Farinelli, A.: Benchmarking safe deep reinforcement learning in aquatic navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021)
35. Marchesini, E., Corsi, D., Farinelli, A.: Exploring safer behaviors for deep reinforcement learning. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI) (2021b)
36. Marron, A., Hacohen, Y., Harel, D., Mülder, A., Terfloth, A.: Embedding scenario-based modeling in Statecharts. In: Proceedings of the 5th International Workshop on Model-driven Robot Software Engineering (MORSE), pp. 443–452 (2018)
37. Mnih, V., et al.: Playing Atari with deep reinforcement learning (2013). Technical Report. <https://arxiv.org/abs/1312.5602>
38. Nandkumar, C., Shukla, P., Varma, V.: Simulation of indoor localization and navigation of Turtlebot 3 using real time object detection. In: Proceedings of the International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON) (2021)
39. Nanfack, G., Temple, P., Frénay, B.: Learning customised decision trees for domain-knowledge constraints. *Pattern Recognit.* **142**, 109610 (2023)
40. Nguyen, H., La, H.: Review of deep reinforcement learning for robot manipulation. In: Proceedings of the 3rd IEEE International Conference on Robotic Computing (IRC) (2019)
41. Pfeiffer, M., et al.: Reinforced imitation: sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. *IEEE Robot. Autom. Lett.* **3**(4), 4423–4430 (2018)
42. Pore, A., et al.: Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4025–4031 (2021)
43. Quigley, M., et al.: ROS: an open-source robot operating system. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2009)
44. Ray, A., Achiam, J., Amodei, D.: Benchmarking safe exploration in deep reinforcement learning (2019). Technical Report. <https://cdn.openai.com/safexp-short.pdf>
45. Roy, J., Girgis, R., Romoff, J., Bacon, P., Pal, C.: Direct behavior specification via constrained reinforcement learning (2021). Technical Report. <https://arxiv.org/abs/2112.12228>
46. Sallab, A., Abdou, M., Perot, E., Yogamani, S.: Deep reinforcement learning framework for autonomous driving. *Electron. Imag.* **19**, 70–76 (2017)
47. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017). Technical Report. <http://arxiv.org/abs/1707.06347>
48. Srinivasan, K., Eysenbach, B., Ha, S., Tan, J., Finn, C.: Learning to be safe: deep RL with a safety critic (2020). Technical Report. <http://arxiv.org/abs/2010.14603>

49. Stooke, A., Achiam, J., Abbeel, P.: Responsive safety in reinforcement learning by Pid Lagrangian methods. In: Proceedings of the 37th International Conference on Machine Learning (ICML), pp. 9133–9143 (2020)
50. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT press (2018)
51. Tai, L., Paolo, G., Liu, M.: Virtual-to-real deep reinforcement learning: continuous control of mobile robots for Mapless navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 31–36 (2017)
52. Tessler, C., Mankowitz, D., Mannor, S.: Reward constrained policy optimization. In: Proceedings 6th International Conference on Learning Representations (ICLR) (2018)
53. Thananjeyan, B., et al.: Recovery RL: safe reinforcement learning with learned recovery zones. *IEEE Robot. Autom. Lett.* **6**, 4915–4922 (2021)
54. Wang, S., Pei, K., Whitehouse, J., Yang, J., Jana, S.: Formal security analysis of neural networks using symbolic intervals. In: Proceedings of the 27th USENIX Security Symposium, pp. 1599–1614 (2018)
55. Yang, L., et al.: Constrained update projection approach to safe policy optimization. In: Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS) (2022)
56. Yerushalmi, R., Amir, G., Elyasaf, A., Harel, D., Katz, G., Marron, A.: Scenario-assisted deep reinforcement learning. In: Proceedings of the 10th International Conference on Model-Driven Engineering and Software Development (MODEL-SWARD), pp. 310–319 (2022)
57. Zhang, J., Springenberg, J., Boedecker, J., Burgard, W.: Deep reinforcement learning with successor features for navigation across similar environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
58. Zhang, L., et al.: Penalized proximal policy optimization for safe reinforcement learning. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI) (2022)
59. Zhang, Y., Vuong, Q., Ross, K.: First order constrained optimization in policy space. In: Proceedings 34th Conference on Neural Information Processing Systems (NeurIPS) (2020)
60. Zhao, W., Queraltà, J., Westerlund, T.: Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI) (2020)
61. Zhu, K., Zhang, T.: Deep reinforcement learning based mobile robot navigation: a review. *Tsinghua Science and Technology* (2021)



Explainable AI in Feature Selection: Improving Classification Performance on Imbalanced Datasets

Shahriar Siddique Ayon¹ , Muhammad Ebrahim Hossain¹ ,
Md Saef Ullah Miah¹ , M. Mostafizur Rahman² ,
and Mufti Mahmud^{3,4,5}

¹ Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh (AIUB), Dhaka 1229, Bangladesh
saeef@aiub.edu

² Department of Mathematics, Faculty of Science and Technology, American International University-Bangladesh (AIUB), Dhaka 1229, Bangladesh
mostafiz.math@aiub.edu

³ Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

⁴ Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

⁵ SDAIA-KFUPM Joint Research Center for AI, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
mufti.mahmud@ntu.ac.uk

Abstract. Particularly in biomedical applications, feature selection plays a critical role in enhancing the interpretability and efficacy of machine learning models. This work examines the performance of Explainable AI (XAI), Information Gain (IG), and Principal Component Analysis (PCA) methods on an imbalanced dataset pertaining to stroke prediction. Data from 4,603 patient records, including 362 instances of stroke from the National Health and Nutrition Examination Survey are used in this investigation. Methodologically, IG is used for feature ranking, PCA is used to reduce dimensionality and XAI techniques are used to improve model transparency. The chosen features are used to assess the performance of several machine learning models, including Random Forest, Support Vector Machine, k-Nearest Neighbours, and Logistic Regression, in terms of classification. Our experimental results show that the combined PCA-IG approach significantly enhances classification accuracy, achieving 91.75%. Furthermore, LIME-based feature selection outperformed in precision, recall, and F1 score, with the highest accuracy at 91.86%. LIME discovered nine positive impact features, highlighting the top contributors in the dataset. We also applied the same feature selection technique to datasets from other domains. These findings highlight the robustness of using PCA-IG and XAI approaches separately to create reliable and understandable machine learning models for healthcare and other applications. By offering insights into the optimal use of PCA, IG, and XAI to enhance the accuracy and practicality of machine learning

models in healthcare and other domains, this paper advances the field of feature selection across all areas of data analysis.

Keywords: Feature Selection · Classification · (Principal Component Analysis) · Imbalanced Data · Explainable AI

1 Introduction

Data classification is the process of classifying data into distinct groups, which is often carried out in two stages: first, developing a model for the class attribute based on other factors, and then applying this model to fresh, previously unknown datasets to determine each record's class [1, 15]. Classification algorithms, which are widely utilized in domains such as health, astronomy, business, biology, and media, frequently come across imbalanced datasets in which one class outnumbers the other(s) [16, 24]. This common occurrence in applications such as medical diagnosis, fraud detection, and spam filtering causes significant issues for machine learning (ML) algorithms, resulting in biased models that favor the majority while ignoring the minority [3, 4].

Traditional feature selection approaches frequently face considerable hurdles when dealing with imbalanced datasets, owing to the overwhelming effect of the majority class [14]. This imbalance may skew feature selection, leading to poor representation of the minority class and biased classification results. As a result, the model fails to generalize well, especially for the minority class, which is frequently the class of greatest interest in practical applications [7]. This work addresses feature selection in imbalanced datasets using advanced techniques like Principal Component Analysis (PCA), Information Gain (IG), and Explainable AI (LIME). These algorithms are employed to identify and select important features, aiming to enhance classification performance [21].

We conducted a series of experiments on an imbalanced dataset using various ML classification algorithms. Feature selection was performed using PCA, IG, and LIME. After applying PCA, a feature selection algorithm was used. The models' performance was then evaluated in terms of accuracy, precision, recall, and F1 score.

This study highlights the role of effective feature selection in enhancing ML model performance on imbalanced datasets. Using LIME to assess both positive and negative feature impacts, we demonstrate how reducing dimensions and selecting meaningful features enhance model accuracy and interpretability. The key contributions of this paper are:

- An extensive analysis of feature selection in the setting of imbalanced data using PCA, IG, and LIME.
- A analysis of how each feature selection approach affects the performance of several classification algorithms.
- Evidence suggests LIME-based feature selection outperforms traditional approaches, especially in terms of accuracy and interpretability.

The rest of this paper is organized as follows: Sect. 2 provides a detailed review of related work. Section 3 discusses the dataset and experimental setup. Section 4 offers results and analyses. Section 5 explains the findings and their implications for society. Finally, Sect. 6 concludes the paper and suggests directions for future research.

2 Related Works

Zebari et al. [26] underline the relevance of accuracy and dimensionality reduction in data mining and machine learning, especially as data dimensions increase. They demonstrate that duplicate features can impede pattern classification. Their study, which included datasets such as medical analysis, ethnicity identification, and emotion recognition, found that feature selection beat other approaches in power quality events, but it still has an overfitting risk.

Khaire et al. [12] propose a novel stability estimator to address the instability of feature selection, emphasizing the importance of stability in feature selection methods across different domains. They explore network algorithms for stable feature selection, such as RBNAs, and use the relative weighted consistency approach to tackle subset-size bias. They point out that instability arises when algorithms are designed without considering the curse of dimensionality and stability.

Spencer et al. [18] demonstrate how machine learning has improved the accuracy of computer-aided diagnosis systems, namely through feature selection strategies in classification algorithms. They experimented using BayesNet, Logistic, Adaboost M1, JRip, Random Forest, and Stochastic Gradient Descent, and achieved 85% accuracy using a BayesNet and Chi-squared combo. Their findings highlight the importance of feature selection for machine learning outcomes.

Filter methods are crucial for feature selection as they can be used with any machine learning model and significantly shorten the execution time of algorithms. Bommert et al. [5] benchmarked selection approaches for high-dimensional datasets, examining 22 filter techniques using 16 high-dimensional classification datasets. Information theoretic, statistical, and random forest test filters performed well, though not every filter technique outperformed others in all scenarios. Their work serves as a reference for applications involving feature filtering.

Zacharias et al. [25] discuss the growing impact of AI systems on humans and the importance of transparency to uphold people’s “right to explanation.” They focus on explainable feature selection in AI systems, evaluating effectiveness, comprehension of feature impact, and justifiability. Their iterative design research cycle aims to create and assess artifacts to enhance the openness and comprehensibility of feature selection tasks.

Dimensionality reduction techniques in feature selection and feature extraction improve data mining and machine learning tasks by reducing redundancy, removing irrelevant data, and enhancing model accuracy and efficiency [26].

Techniques such as Principal Component Analysis (PCA) and t-SNE project features into a lower-dimensional space, preserving crucial data for analysis while improving processing and storage efficiency [11].

Enhancing classification performance across various domains requires careful consideration of feature selection for classification using PCA and Information Gain (IG). Research has shown that hybrid models integrating PCA and IG reduce data dimensions, select suitable features, and improve classification metrics such as accuracy, precision, and recall [13]. PCA significantly reduces pre-processing time while maintaining comparable performance outcomes, particularly in contexts like classifying Influenza-A antiviral resistance [17].

Explainable Artificial Intelligence (XAI) strategies, combined with feature selection methods, are critical for improving the interpretability and transparency of ML models, especially in the biomedical domain [22]. These techniques reduce the number of variables in model explanations and decision-making processes while preserving essential information [19]. Various machine learning models, such as k-nearest neighbors, XGBoost, AdaBoost, logistic regression, random forest, decision tree, Naive Bayes, and SVM, have been combined with ReliefF feature selection to enhance performance [6].

Developing feature selection techniques grounded in explainable artificial intelligence (XAI) addresses the lack of interpretability and transparency in artificial neural networks, particularly in defect diagnostics systems [8]. For applications such as Intrusion Detection Systems (IDS), feature selection enhances both explainability and prediction performance in ML models [2]. In biological applications, balancing explainability, accuracy, and retention rate in feature selection has a substantial impact on final model explanations [22]. Researchers highlight the need for measuring causal strength and optimizing feature selection models for improved interpretability and robustness, using objective functions like relative entropy distance (RED).

3 Methodology

In this section, we outline the steps followed in our study. Our methodology is organized into several key stages: data collection, feature selection using PCA, IG, and XAI. This is followed by machine learning model selection and customization to interpret the model predictions. The proposed methodology is summarized in Fig. 1, providing a clear and concise overview of our strategy.

3.1 Data Collection

The dataset utilized in this study, named “Imbalanced Data-based Prediction and Risk Factor Analysis of Stroke”, is available on Mendeley [23]. It is based on data from the National Health and Nutrition Examination Survey and other national surveys. The dataset includes records for 4,603 patients: 362 (7.86%) with stroke and 4,241 (92.14%) without. It has 36 columns, with the first column

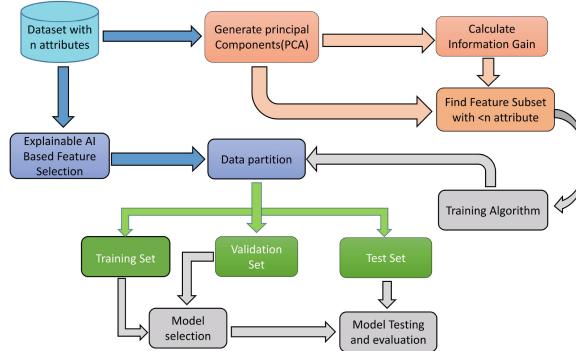


Fig. 1. Proposed Feature Selection Strategy for Imbalanced Data Classification.

indicating stroke status and the remaining columns as predictors. These predictors include gender, age, race, marital status, alcohol consumption, smoking status, sleep disorder, health insurance status, general health condition, depression, sleep time, diabetes, hypertension, high cholesterol, minutes of sedentary activity, coronary heart disease, body mass index (BMI), waist circumference, systolic blood pressure, diastolic blood pressure, high-density lipoprotein (HDL), triglycerides, low-density lipoprotein (LDL), fasting glucose, glycohemoglobin, energy intake, protein intake, carbohydrate intake, dietary fiber intake, total fat intake, total saturated fatty acids intake, total monounsaturated fatty acids intake, total polyunsaturated fatty acids intake, potassium intake, and sodium intake. This extensive dataset, which is entirely composed of numerical data, is used to study the prediction and analysis of stroke risk factors. We have three additional imbalanced datasets for testing our proposed method: IT Customer Churn (20 features), Bank Marketing (17 features), and Car Insurance Claim (25 features), all available on Kaggle [9, 10, 20]. The IT Customer Churn dataset has 1,869 churn and 5,174 non-churn records. The Bank Marketing dataset includes 5,289 deposit and 5,873 non-deposit records. And the Car Insurance Claim dataset comprises 2,746 claim and 7,556 non-claim records. To ensure robust model performance, we divided the dataset into 80% training and 20% testing, with an additional 20% remained aside as a validation subset. Table 1 shows the data partitioned specifically for model training.

Table 1. Train, Test, and Validation split of the employed datasets.

Dataset Name	Target Column	Number of Samples	Train	Test	Valid
Imbalanced Stroke	stroke	4603	2946	921	736
IT Customer Churn	Churn	7043	4508	1409	1126
Bank Marketing	deposit	11162	7132	2232	1798
Car insurance claim	CLAIM_FLAG	10302	6593	2060	1649

3.2 Principal Component Analysis (PCA) for Feature Reduction

PCA is an effective dimensionality reduction technique that transforms correlated variables into a set of uncorrelated principal components, which is useful for managing large, high-dimensional, and imbalanced datasets. PCA computes the covariance matrix of centered data, performs eigenvalue decomposition to obtain eigenvectors and eigenvalues, and selects principal components based on the largest eigenvalues. Given a dataset X with N samples and d features, PCA aims to find a transformation matrix W such that,

$$X_{new} = X \cdot W \quad (1)$$

PCA converts the original data matrix X (size $n*d$) into a reduced-dimensional space using W (size $d*k$), where K is the number of principal components. The results X_{new} (size $n*k$), captures the dataset's variance with the most significant orthogonal components.

3.3 Feature Selection Techniques

After performing PCA for feature reduction, we applied four different feature selection techniques. They are briefly described below.

Correlation-Based Feature Selection Evaluation (CfsSubsetEval). CfsSubsetEval selects feature subsets by maximizing predictive accuracy while minimizing redundancy, based on their correlation with the target variable and among features. The merit $M(S)$ of a subset S of features is calculated as follows,

$$M(S) = \frac{k \cdot \text{cor}(\mathbf{X}_i, \mathbf{y})}{\sqrt{k + \frac{k \cdot (k-1)}{2} \cdot \text{avgCor}(\mathbf{X})}} \quad (2)$$

In CfsSubEval, K is the subset size, $\text{Cor}(X_i, y)$ is feature target correlation, and $\text{avgCor}(X)$ is the average feature correlation within S , crucial for feature selection.

Gain Ratio Attribute Evaluation (GainRatioAttributeEval). A feature selection technique called GainRatioAttributeEval assesses attributes according to their gain ratio, which quantifies how well an attribute classifies instances in relation to the inherent information of classes.

The gain ratio $GR(S)$ for a subset S of attributes is calculated as follows,

$$GR(S) = \frac{IG(S)}{IV(S)} \quad (3)$$

In GainRatioAttributeEval, $IG(S)$ is the information gain and $IV(S)$ is the intrinsic of subset S , crucial for attribute evaluation.

ReliefF Attribute Evaluation (ReliefFAttributeEval). ReliefFAttributeEval measures a feature's importance by evaluating how well it distinguishes between nearby examples and adjusts weights based on differences between an instance and its neighbors. The weight $W[A]$ of an attribute A is updated based on the following equation.

$$W[A] = W[A] - \frac{1}{m} \sum_{i=1}^m (\text{diff}(A, \mathbf{x}_i, \text{nearest hit}) - \text{diff}(A, \mathbf{x}_i, \text{nearest-miss})) \quad (4)$$

where $W[A]$ is the weight of attribute A , m is the number of iterations, and \mathbf{x}_i is the sampled instance. The terms $\text{diff}(A, \mathbf{x}_i, \text{nearest-hit})$ and $\text{diff}(A, \mathbf{x}_i, \text{nearest-miss})$ denote differences with the nearest same-class and different-class neighbors, respectively.

Principal Component Analysis Information Gain (PCA-IG). PCA-IG combines PCA with Information Gain to select the most relevant features from a high-dimensional dataset. After applying PCA, information gain is used on the transformed features to assess their relevance to the target variable. Equation (1) describes PCA, while Information Gain for a feature A is defined as,

$$IG(A) = H(Y) - H(Y|A) \quad (5)$$

where $H(Y)$ is the entropy of the target variable and $H(Y|A)$ is the conditional entropy of the target variable given the feature A .

This hybrid approach is ideal for large, complex datasets where traditional feature selection methods struggle with high dimensionality.

3.4 Explainable AI (XAI)-Based Feature Selection

We utilized LIME as XAI in selecting the important features from the dataset. The approach is briefly described in the following section.

Local Interpretable Model-Agnostic Explanations (LIME). LIME simplifies complex model behaviors into interpretable forms, offering local interpretability to explain specific predictions. It identified features with both positive and negative impacts and we utilized random forest-based LIME feature selection for analysis. The LIME explanation model is represented as follows,

$$\hat{g}(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_{x'}) + \Omega(g) \quad (6)$$

In LIME, the explanation model $\hat{g}(x)$ approximates the local behavior of a complex model using a set G of potential models, often linear. A loss function $L(f, g, \pi_{x'})$ is employed to optimize the simplicity and interpretability of the perturbed explanation model.

3.5 ML Model Selection and Hyper-Parameter Tuning

After performing PCA and feature selection, we identified top-performing ML algorithms: Random Forest Classifier (RFC), Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Logistic Regressor (LR), XGBoost (XGB) Classifier, and Extra Trees Classifier for classifying the results in our dataset. After selecting these ML models, we initially evaluated them using their default settings. Subsequently, we adjusted the hyperparameters to enhance the models' accuracy. Table 2 presents the hyperparameter configurations for each model.

Table 2. Hyperparameter settings for each machine learning model.

Model	Hyperparameters Setting
RFC	n_estimators: 100, 200, 300, max_depth: None, 10, 20, 30, min_samples_split: 2, 5, 10, min_samples_leaf: 1, 2, 4, max_features: ‘auto’, ‘sqrt’, ‘log2’
SVC	C: 1.0, kernel: ‘rbf’, gamma: ‘scale’, degree: 3
KNN	n_neighbors: 5, weights: ‘uniform’, algorithm: ‘auto’, leaf_size: 30, p: 2
LR	penalty: ‘l2’, C: 1.0, solver: ‘lbfgs’, ‘liblinear’, ‘saga’, max_iter: 100
XGB Classifier	n_estimators: 100, random_state: 42
Extra Tress Classifier	n_estimators: 100, random_state: 42

In Table 2, RFC is tuned with hyperparameters like n_estimators (100, 200, 300), max_depth (None, 10, 20, 30), min_samples_split (2, 5, 10), min_samples_leaf (1, 2, 4), and max_features (‘auto’, ‘sqrt’, ‘log2’). These parameters influence the forest's complexity, impacting both its generalization ability and computational efficiency. SVC optimizes classification using key hyperparameters: C for regularization strength (default 1.0), kernel (default ‘rbf’), gamma (default ‘scale’), and degree (default 3 for ‘poly’ kernels). KNN classifier relies on a single hyperparameter, n_neighbors, which specifies the number of neighbors to consider during classification (default is 5). LR uses hyperparameters like penalty (default ‘l2’) for regularization type, C (default 1.0) for regularization strength inversely proportional to regularization, and solver (‘lbfgs’, ‘liblinear’, ‘saga’) for optimization method. The n_estimators parameter specifies the number of trees in boosting, while random_state ensures reproducibility for XGB and Extra Trees classifiers. Optimal tuning enhances model performance, boosting robustness and accuracy across various applications.

4 Experimental Results and Analysis

In our experimental setup, we used Google Colab for developing ML models in Python, utilizing PCA, IG for feature reduction, and LIME for explainability.

Colab provided reliable resources for extended testing and improved collaboration, with Scikit-Learn handling all machine learning tasks. In this section, we present an analysis of the prediction outcomes from various models, employing both the full feature sets and the selected features identified through PCA, IG and XAI methods. A comprehensive examination of these results is detailed below.

4.1 Performance Comparison Using All Features (Without Feature Selection)

These findings summarize the classification experiment performed on the dataset before to feature selection, and include metrics such as precision, recall, F1 score, and accuracy. Table 3 shows a full summary of these findings.

Table 3. Performance of Classification Methods with All Features.

Dataset Name	Algorithm	Precision	Recall	F1-Score	Accuracy(%)
Imbalanced Stroke	RFC	0.84	0.91	0.87	91.74
	SVC	0.84	0.91	0.87	91.74
	KNN	0.84	0.91	0.87	91.42
	LR	0.84	0.91	0.84	91.74
IT Customer Churn	RFC	0.65	0.45	0.54	79.20
	LR	0.65	0.54	0.59	80.19
	SVC	0.41	0.59	0.48	66.78
Bank Marketing	RFC	0.80	0.86	0.83	83.29
	XGB Classifier	0.81	0.85	0.83	83.47
	LR	0.79	0.76	0.77	78.63
Car Insurance Claim	KNN	0.29	0.44	0.35	55.70
	Extra Trees Classifier	0.96	0.93	0.94	97.33
	LR	0.50	0.002	0.004	73.07

Table 3 presents the performance results of the best machine learning models across various datasets with all features considered. For the stroke dataset, RFC, SVC, and LR all achieved a precision, recall, and F1 score of 91.74% accuracy. In the IT customer dataset, LR achieved the highest accuracy at 80.19%. The bank marketing dataset saw the XGB classifier attaining the top accuracy of 83.47%. Lastly, the car insurance claim dataset was best modeled by the Extra Trees classifier, which reached an accuracy of 97.33%.

4.2 Performance After Feature Selection

Further experiments were conducted across all datasets using a variety of feature selection techniques. These methods included Cfs Subset Evaluation, Gain

Ratio Attribute Evaluation, ReliefF Attribute Evaluation, PCA in combination with IG, and LIME-based feature selection. For each dataset, PCA was utilized with cross-validation and cumulative importance methods to identify crucial features. Here, we present detailed results for the imbalanced stroke dataset, where the number of selected features varied between nine (9) and twenty-two (22). Subsequently, the selected features were evaluated using the four classification algorithms mentioned earlier. The results of these experiments are presented in Table 4 below.

Table 4. Classification Performance with Feature Selection in Stroke Dataset.

Classification Algorithm	Feature Selection	Precision	Recall	F1-Score	Accuracy (%)
RFC	CfsSubsetEval	0.84	0.91	0.87	91.74
	GainRatioAttributeEval	0.84	0.91	0.87	91.74
	ReliefAttributeEval	0.84	0.91	0.87	91.74
	PCA-IG	0.84	0.91	0.88	91.75
	LIME	0.93	0.92	0.88	91.86
SVC	CfsSubsetEval	0.84	0.91	0.87	91.74
	GainRatioAttributeEval	0.84	0.91	0.87	91.74
	ReliefAttributeEval	0.84	0.91	0.87	91.74
	PCA-IG	0.84	0.92	0.88	91.75
	LIME	0.84	0.92	0.88	91.75
KNN	CfsSubsetEval	0.84	0.91	0.87	91.53
	GainRatioAttributeEval	0.84	0.90	0.87	91.66
	ReliefAttributeEval	0.86	0.92	0.87	91.53
	PCA-IG	0.84	0.91	0.87	91.09
	LIME	0.85	0.91	0.88	91.21
LR	CfsSubsetEval	0.84	0.91	0.87	91.74
	GainRatioAttributeEval	0.84	0.91	0.87	91.74
	ReliefAttributeEval	0.84	0.91	0.87	91.74
	PCA-IG	0.84	0.92	0.88	91.75
	LIME	0.84	0.92	0.88	91.75

In Table 4, the results of four classification algorithms, each evaluated with five feature selection techniques, are presented. In KNN, ReliefF achieved the best precision and recall scores, around 0.86 and 0.92, respectively. The combined PCA-IG method performed better in three classification models, with precision at 0.84, recall at 0.92, F1 score at 0.88, and an accuracy of 91.75%. LIME-based feature selection outperformed all other methods across various metrics. For SVC, KNN, and LR models, LIME and PCA-IG techniques produced similar results. However, LIME-based feature selection using a RFC achieved the highest

overall performance, with a precision of 0.93, recall of 0.92, F1 score of 0.88, and accuracy of 91.86%. Overall, LIME-based feature selection with the RFC classifier provided the best results, indicating that XAI techniques are more effective than PCA in handling this imbalanced dataset.

The PCA-IG feature selection model resulted in 14 final features, while the LIME-based feature selection model identified 9 final features. These 9 features include ‘Race’, ‘Alcohol’, ‘High Cholesterol’, ‘Coronary Heart Disease’, ‘Waist Circumference’, ‘Diastolic Blood Pressure’, ‘Total Fat’, ‘Total Saturated Fatty Acids’, and ‘Total Monounsaturated Fatty Acids’. Among them, ‘Coronary Heart Disease’, ‘Total Saturated Fatty Acids’, and ‘Total Fat’ are the top 3 positive contributors as identified by LIME.

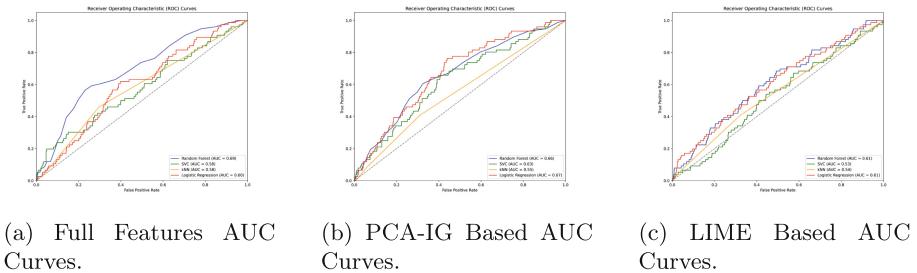


Fig. 2. AUC Curve Comparison Across Multiple Classifier Models Using Three Feature Scenarios in the Imbalanced Stroke Dataset.

Figure 2 illustrates AUC curves for all classification models across three different feature selection scenarios in the imbalanced stroke dataset. In 2a, utilizing full features, RFC had the greatest AUC of 0.69, followed by LR at 0.60. In 2b, utilizing PCA-IG-selected features, LR had the greatest AUC of 0.67, whereas RFC scored 0.66. In 2c, utilizing LIME-selected features, RFC and LR scored 0.61, whereas SVC and KNN performed similarly across scenarios. These findings show that each feature influences AUC performance, with PCA-IG and LIME-based selections generally producing better outcomes, albeit slightly lower overall AUC values.

4.3 Testing of Feature Selection Methods on Different Datasets

We tested our proposed method on an imbalanced stroke dataset and are now evaluating it on three more imbalanced datasets: IT customer churn, bank marketing, and car insurance claims. We first identify the best ML model for each dataset, then use LIME to pinpoint key features and analyze the results. Table 5 provides detailed results of our analysis.

In Table 5, the IT customer churn dataset shows the LR model performing the best, with an accuracy of 80.19% using all features. We then used PCA, IG, and XAI for feature selection. Both PCA-IG and LIME-based selections

Table 5. Feature Selection Performance Across Datasets.

Data set	ML Algorithm	Feature Selection	Precision	Recall	F1-Score	Accuracy (%)	AUC Curve
IT Customer	LR	Full Features	0.65	0.54	0.59	80.19	0.85
		CfsSubSetEval	0.66	0.52	0.58	80.41	0.84
		Gain Ratio AttributeEval	0.62	0.30	0.40	76.65	0.81
		Relief AttributeEval	0.64	0.52	0.57	79.84	0.84
		PCA-IG	0.69	0.52	0.59	81.05	0.84
Churn		LIME	0.67	0.57	0.62	81.19	0.86
		Full Features	0.81	0.85	0.83	83.47	0.91
		CfsSubSetEval	0.72	0.57	0.64	69.46	0.75
		Gain-Ratio-AttributeEval	0.72	0.45	0.56	65.65	0.70
		Relief-AttributeEval	0.81	0.85	0.83	83.78	0.90
Bank Marketing	XGB Classifier	PCA-IG	0.81	0.85	0.83	83.21	0.91
		LIME	0.82	0.85	0.83	83.66	0.91
		Full Features	0.96	0.93	0.94	97.33	1.00
		CfsSubSetEval	0.99	0.98	0.99	99.56	1.00
		Gain Ratio AttributeEval	0.98	0.97	0.98	98.98	1.00
Car Insurance	Extra Trees Classifier	Relief-AttributeEval	1.00	0.98	0.99	99.61	1.00
		PCA-IG	0.99	0.98	0.98	99.32	1.00
		LIME	0.99	0.99	0.99	99.66	1.00

performed well. PCA-IG achieved a precision of 0.69 and accuracy of 81.05%. However, our proposed LIME-based method outperformed in all criteria, with an accuracy of 81.19%. In the bank marketing dataset, the XGB classifier has the maximum accuracy across all features, scoring 83.47%. Using feature selection techniques, two models stand out: PCA-based ReliefAttributeEval and LIME-based selection. The ReliefAttributeEval model achieves an accuracy of 83.78%. However, our proposed LIME-based model outperforms the full feature model, achieving a precision of 0.82 and an accuracy of 83.66%. In the car insurance claim dataset, the Extra Trees classifier performs best with all features, achieving 97.33% accuracy. After feature selection, all methods improved accuracy, but our proposed LIME-based method outperformed them all. It achieved a precision, recall, and F1 score of 0.99, an accuracy of 99.66%.

The AUC curves of the top-performing models with LIME-based feature selection are displayed in Fig. 3. In Fig. 3a, the Customer Churn dataset with an LR model achieved an AUC of 0.86. Figure 3b depicts the Bank Marketing dataset with an XGB classifier, achieving an AUC of 0.91. Figure 3c illustrates the Car Insurance Claim dataset with an Extra Trees classifier, reaching an AUC

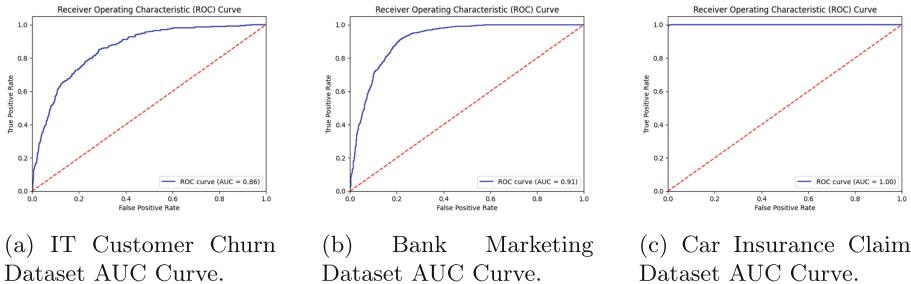


Fig. 3. AUC Curve for Best LIME-based Model Across Tested Datasets.

of 1.00. Across all datasets tested, our suggested LIME-based feature selection method consistently produced the best results for imbalanced datasets.

5 Discussion

This section reviews our results, highlighting experiments with LIME and RFC, along with other classifiers, on the imbalanced stroke dataset. Using all features led to lower accuracy, as unnecessary data can reduce algorithm efficiency.

After feature selection and training, the PCA-IG and LIME models outperformed the other methods. Table 4 shows that PCA-IG performs better than basic PCA. However, LIME attained the maximum accuracy, with the PCA-IG model scoring 91.75% and the LIME model scoring 91.86% with the RFC classifier. We analyzed an imbalanced stroke dataset to determine the most influential factors affecting strokes. The LIME feature selection technique considers both positive and negative influences when determining the relevance of each feature. Following LIME-based feature selection, we identified nine features that had a positive impact on the model. The top three contributions are ‘Coronary Heart Disease’, ‘Total Saturated Fatty Acids’, and ‘Total Fat’. In the AUC study of the imbalanced stroke dataset, the full feature set yielded the highest AUC values. Feature selection with PCA, Information Gain, and LIME led to a slight drop in AUC scores but improved overall accuracy.

We also tested our results on three other imbalanced datasets from different domains. For each, we identified the best ML model and found that combining it with LIME-based feature selection achieved higher accuracy than using the full feature set, PCA, or PCA-IG techniques. Notably, the AUC scores improved with LIME-based feature selection across all tested datasets. This feature selection strategy enhances medical and other domain analyses by providing greater insights and aiding decision-making. By reducing data dimensions and selecting more meaningful features, we observed improvements in precision and recall, demonstrating that effective feature selection enhances machine learning efficiency. These results underscore the effectiveness of LIME in feature selection, leading to more accurate classifications.

6 Conclusion and Future Work

This study explored advanced feature selection techniques, including PCA, IG, and XAI (LIME), for classifying imbalanced data. Results showed that using all features reduced accuracy, while PCA-IG and LIME effectively reduced dimensionality, enhancing model performance and feature relevance. Among these, LIME feature selection stood out, achieving the highest accuracy with the RFC on the imbalanced stroke dataset. In other tested datasets, the LIME-based feature selection technique resulted in the highest accuracy for the LR, XGB classifier, and Extra Trees classifier across various domain scenarios. These results show LIME's effectiveness in identifying key features and the potential of XAI to improve classification in imbalanced datasets.

Future research will look into additional feature selection approaches and combinations to improve classification performance. Genetic algorithms, mutual information, and ensemble approaches will be discussed. Furthermore, using imbalanced data management strategies like SMOTE, ADASYN, and ensemble learning methods may improve model robustness. Another focus will be on investigating the applicability of these techniques in various domains and the use of deep learning technologies. The goal is to provide more adaptable and efficient feature selection frameworks for a variety of classification jobs.

Data and Code Availability. The codes and data employed in this study can be found in the following GitHub repository. https://github.com/ShahriarAyon63/feature_selection_pca.

References

1. Aggarwal, C.C., Aggarwal, C.C.: Data Classification. Springer (2015)
2. Akintade, S., Kim, S., Roy, K.: Explaining machine learning-based feature selection of IDs for IoT and CPS devices. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 69–80. Springer (2023)
3. Bhowmik, A., Noor, N.M., Miah, M., Karmaker, D.: Aspect-based sentiment analysis model for evaluating teachers' performance from students' feedback. AIUB AJSE **22**(3), 287–294 (2023)
4. Bin Sulaiman, R., Schetinin, V., Sant, P.: Review of machine learning approach on credit card fraud detection. Hum.-Centric Intell. Syst. **2**(1), 55–68 (2022)
5. Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., Lang, M.: Benchmark for filter methods for feature selection in high-dimensional classification data. Comput. Stat. Data Anal. **143**, 106839 (2020)
6. Elmannai, H., et al.: Polycystic ovary syndrome detection machine learning model based on optimized feature selection and explainable artificial intelligence. Diagnostics **13**(8), 1506 (2023)
7. Feng, F., Li, K.C., Shen, J., Zhou, Q., Yang, X.: Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification. IEEE Access **8**, 69979–69996 (2020)

8. Guo, C., Shang, Z., Ren, J., Zhao, Z., Wang, S., Chen, X.: Instance-wise causal feature selection explainer for rotating machinery fault diagnosis. In: 2022 International Conference on Sensing, Measurement & Data Analytics in the Era of Artificial Intelligence (ICSMD), pp. 1–6. IEEE (2022)
9. Guy, R.: Bank marketing (2018). <https://www.kaggle.com/datasets/rouseguy/bankbalanced>
10. Jh, M.: Car insurance claim (2021). <https://www.kaggle.com/datasets/mukuljh2/car-insurance-claim/data>
11. Jindal, P., Kumar, D.: A review on dimensionality reduction techniques. Int. J. Comput. Appl. **173**(2), 42–46 (2017)
12. Khaire, U.M., Dhanalakshmi, R.: Stability of feature selection algorithm: a review. J. King Saud Univ.-Comput. Inf. Sci. **34**(4), 1060–1073 (2022)
13. Lingwal, S., et al.: Info_PCA: a hybrid technique to improve accuracy by dimensionality reduction. Inf. Technol. Ind. **9**(2), 458–466 (2021)
14. Matharaarachchi, S., Domaratzki, M., Muthukumarana, S.: Assessing feature selection method performance with class imbalance data. Mach. Learn. Appl. **6**, 100170 (2021)
15. Meem, S.M., Hossain, M.T., Chowdhury, J.K., Miah, M.S.U., Monir, M.F.: Understanding the dynamics of dengue in Bangladesh: EDA, climate correlation, and predictive modeling. In: TENCON 2023-2023 IEEE Region 10 Conference (TENCON), pp. 1309–1314. IEEE (2023)
16. Sen, S., Agarwal, S., Chakraborty, P., Singh, K.P.: Astronomical big data processing using machine learning: a comprehensive review. Exp. Astron. **53**, 1–43 (2022). <https://doi.org/10.1007/s10686-021-09827-4>
17. Shaltout, N., Moustafa, M., Rafea, A., Moustafa, A., El-Hefnawi, M.: Comparing PCA to information gain as a feature selection method for *Influenza-a* classification. In: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), pp. 279–283. IEEE (2015)
18. Spencer, R., Thabtah, F., Abdelhamid, N., Thompson, M.: Exploring feature selection and classification methods for predicting heart disease. Digital Health **6**, 2055207620914777 (2020)
19. Tasnim, N., Al Mamun, S.: Comparative performance analysis of feature selection for mortality prediction in ICU with explainable artificial intelligence. In: 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–6. IEEE (2023)
20. Tehranipour, S.: It customer churn (goal: imbalanced dataset) (2020). <https://www.kaggle.com/datasets/soheiltehranipour/it-customer-churn>
21. Thakkar, A., Lohiya, R.: A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif. Intell. Rev. **55**(1), 453–563 (2022)
22. Wang, H., Doumard, E., Soulé-Dupuy, C., Kemoun, P., Aligon, J., Monsarrat, P.: Explanations as a new metric for feature selection: a systematic approach. IEEE J. Biomed. Health Inform. **27**(8), 4131–4142 (2023)
23. Wang, P.: Imbalanced data-based prediction and risk factor analysis of stroke (2024). <https://doi.org/10.17632/xggs239bnw.1>
24. Xing, W., Bei, Y.: Medical health big data classification based on KNN classification algorithm. IEEE Access **8**, 28808–28819 (2019)

25. Zacharias, J., von Zahn, M., Chen, J., Hinz, O.: Designing a feature selection method based on explainable artificial intelligence. *Electron. Mark.* **32**(4), 2159–2184 (2022)
26. Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J.: A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J. Appl. Sci. Technol. Trends* **1**(1), 56–70 (2020)



Enhancing Industrial Energy Efficiency with Predictive Analytics and Fuzzy Logic: A Case Study of Renewable Energy Management in the Meat Processing Industry

Mostafa Pasandideh¹(✉), Jason Kurz², Martin Atkins³, and Mark Apperley¹

¹ Ahuora—Centre for Smart Energy Systems, School of Computing and Mathematical Sciences, University of Waikato, Hamilton 3240, New Zealand

sp290@students.waikato.ac.nz, m.apperley@waikato.ac.nz

² Department of Mathematics, University of Waikato, Hamilton 3240, New Zealand
jason.kurz@waikato.ac.nz

³ Ahuora—Centre for Smart Energy Systems, School of Engineering, University of Waikato, Hamilton 3240, New Zealand

martin.atkins@waikato.ac.nz

Abstract. With growing environmental concerns and the urgent need to mitigate global warming, there is a significant push towards adopting renewable energy sources, such as solar and wind power, which are crucial for reducing reliance on fossil fuels. However, the inherent variability of these sources presents substantial challenges for effective energy management, especially in the industrial sector. This research, focusing on the meat processing industry, adopts a two-pronged approach to tackle these issues. It starts by using a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN) to analyze open-source weather data, aiming to predict the impact of weather variations on renewable energy production. This predictive effort is crucial for enhancing the reliability of renewable sources in industrial applications. Despite the advancements in forecasting, the variable nature of energy supply necessitates efficient management strategies. Therefore, the study implements a Fuzzy Logic system to manage electricity consumption based on real-time energy availability and demand within the meat processing industry. Chosen for its robustness in handling uncertainty, Fuzzy Logic enables more informed decision-making under ambiguous conditions, thereby reducing reliance on conventional energy grids and improving energy use efficiency. This dual strategy aims to foster more sustainable and environmentally friendly industrial operations, addressing both the variability of renewable energy sources and the challenges in energy consumption management.

Keywords: Renewable Energy Management · Industrial Energy Efficiency · Predictive Analytics · Fuzzy Logic · Meat Industry · Artificial Neural Networks (ANNs) · Energy Consumption Management · Weather Impact Forecasting

No academic titles or descriptions of academic positions should be included in the addresses. The affiliations should consist of the author's institution, town/city, and country.

1 Introduction

Environmental concerns, particularly the emission of greenhouse gases, coupled with the ever-growing demand for energy, stand as pivotal reasons for the global shift towards adopting policies aimed at achieving net zero emission. New Zealand stands at the forefront of this initiative, with a bold commitment to achieving net zero carbon dioxide emissions by the year 2050 [1–3]. A key strategy in reaching this ambitious goal involves the utilisation of renewable energy sources. However, transitioning to renewable energy is not without its challenges.

Key among these are the dependency on variable weather conditions, the generally lower efficiency of renewable energy sources compared to traditional fuels, and their limited capacity for electricity production [4]. To navigate these obstacles, extensive research has been undertaken to enhance the predictability and efficiency of renewable energy sources. A cutting-edge approach in this realm is the application of artificial intelligence technologies. These technologies aim to accurately predict the availability of renewable energy sources and the quantity of electricity that can be generated, thereby paving the way for more effective and efficient use of renewable resources [5].

Beyond the challenge of accurately predicting outputs from renewable energy systems, the issue of energy management stands out as a significant concern, often marked by uncertainty. This concern becomes especially pronounced when dealing with a system composed of diverse components, where determining the optimal energy source for use during specific periods is crucial.

In situations laden with uncertainty, employing Fuzzy Logic becomes particularly effective due to its straightforwardness and the absence of a need for intricate mathematical models to support energy management strategies [6]. This study enhances the field by integrating ANN with Fuzzy Logic, offering a dual approach that not only improves prediction accuracy but also addresses the inherent variability and uncertainty of renewable energy sources. The ANN component models complex, non-linear relationships in weather data, which is crucial for accurate energy production forecasting.

Fuzzy Logic, on the other hand, enables adaptive and informed decision-making, especially in scenarios where precise mathematical models are impractical. The effectiveness of this dual approach is further enhanced by the careful selection of fuzzification and defuzzification parameters, significantly improving the performance quality of the control mechanisms [7]. This integration of ANN and Fuzzy Logic offers a substantial improvement over traditional methods, which often treat prediction and management as separate challenges, by providing a comprehensive solution for effective energy management.

This research is directed towards managing the electricity consumption within the meat processing industry, identified as the central case study. The methodology unfolds across two key phases. Initially, a neural network is utilized to predict the solar energy radiation at the designated site, laying the groundwork for the subsequent phase involving the formulation of specific rules and the consideration of forthcoming constraints aimed at optimising resource allocation. The application and implications of the fuzzy model used in this context are thoroughly discussed. With the structure of the study in mind, this paper is organized as follows:

- The background section engages in a detailed examination and analysis of prior research in this area, focusing specifically on the management of electricity consumption by industries.
- The following section explores the case study with greater detail, offering an in-depth exploration of the meat industry's specific segments and the renewable resources leveraged for electricity management.
- Section four presents the results and conclusions, encapsulating the outcomes of the research and the effectiveness of the strategies employed for managing electricity consumption in the meat industry.

Through this structured approach, the study seeks to offer significant insights and methodologies for enhancing electricity consumption efficiency in the meat industry. This is achieved by leveraging predictions of renewable energy availability and implementing strategic management based on fuzzy logic models.

2 Background

Traditional energy management systems face significant challenges in dealing with the variability and unpredictability of renewable energy sources, such as solar and wind. These energy systems were initially designed for stable and consistent electricity generation, which does not align well with the fluctuating nature of renewable energy. This mismatch can cause imbalances in the grid, leading to inefficiencies and potential energy wastage [8]. As a result, there is a growing need for more advanced and adaptive approaches that can better handle these complexities, ensuring more effective integration and utilization of renewable energy sources.

Methodologies in energy management, focusing on the use of Multi-Layer Perceptron (MLP) and fuzzy logic control systems, are outlined in this section. MLPs were chosen for their ability to model non-linear relationships effectively, making them suitable for solar irradiance forecasting, where input-output relationships are complex but stable. Unlike Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), which are more appropriate for spatial or sequential data, MLPs provide an efficient solution without adding unnecessary complexity to the model. By adapting input weights based on data quality and collection intervals, MLPs enhance predictive accuracy.

Fuzzy logic control systems optimize energy usage in scenarios where data precision is limited, employing a structured approach to process imprecise data through defined rules and fuzzy inference. The integration of MLP and fuzzy logic offers a balanced and adaptive approach, improving both prediction accuracy and energy management efficiency. Together, these techniques can play a crucial role in enhancing the flexibility and effectiveness of modern energy systems.

2.1 Artificial Neural Network

Neural networks, a subset of machine learning and fundamental to Deep Learning (DL) algorithms, mimic the structure and functioning of the human brain [9]. They derive their name and architecture from the interconnectedness observed in biological neurons.

Multi-Layer Perceptron (MLP) stands as an original form for neural networks, and in this research, the technique will be explored as a means to forecast solar irradiance.

As depicted in Fig. 1 the operation of a Multilayer Perceptron (MLP) begins with the inputs being multiplied by their respective weights, which indicate the influence of each input. These weighted inputs are then summed. If the resultant sum exceeds the neuron's threshold, the neuron activates and produces an output response.

This process illustrates why MLP is classified as a DL method: it enhances performance by adjusting input weights based on the errors observed in the computations. The accuracy of MLP predictions is primarily influenced by the quality of the input data. Among the factors affecting this quality is the time interval of the data received. The research presented in [10] comprehensively discusses how the timing of data input impacts the predictive accuracy of models like MLP.

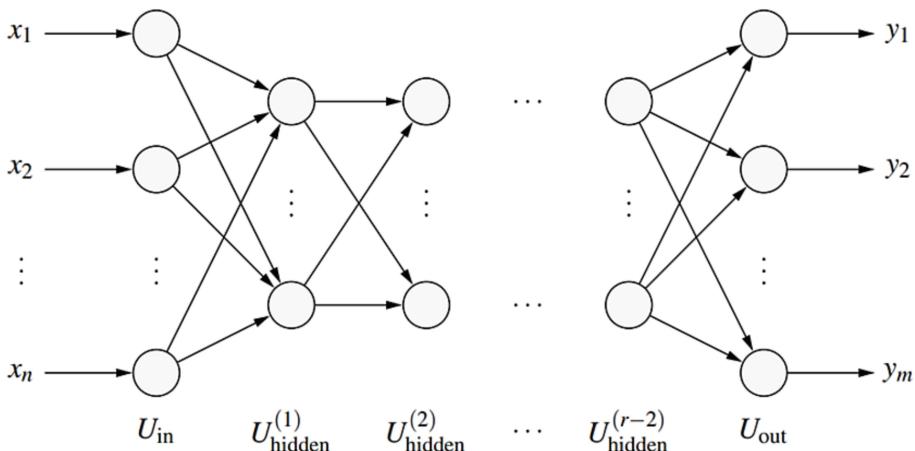


Fig. 1. Typical structure of an r-layered perceptron [11].

A review of the literature has indicated that Table 1 can be used to classify forecasts based on the time intervals of data collection.

Table 1. Time horizons for prediction of solar parameters.

Time horizon	Range
Short-term	Few minutes to 30 min
Medium-term	30 min to 1 day
Long-term	1 day to a month

The impact of data collection intervals on the accuracy of predictive models has been clearly established, particularly in the field of solar energy forecasting [12]. Additionally, the choice and combination of input parameters are critical for accurate predictions of

solar irradiation and the subsequent energy output from solar panels. The review focused on recent studies to explore the methodologies employed, examining how different data collection frequencies and diverse sets of parameters are used in these models.

The analysis aimed to provide insights into the current trends and practices in the modeling of solar energy systems. Table 2 catalogs these studies, offering a systematic overview by the method used, the time intervals of the input data, and the specific parameters considered.

Table 2. Comparison of previous research done on the prediction of solar photovoltaic parameters.

#	Reference	Method	Input Parameters	Time horizon
1	[13]	MLP	Solar Irradiation, Wind Speed, Ambient Temperature, Humidity, Precipitation, Atmospheric Pressure and Wind Direction.	Short-term
2	[14]	MLP	Meteorological data such as sky images, solar irradiance, temperature, humidity, and wind speed.	Medium-term
3	[15]	ANN	Latitude, longitude, altitude, months of a year, maximum temperature, minimum temperature, humidity, wind velocity, and the sunshine hour.	Short-term
4	[16]	ANN	Horizontal Irradiation (GHI), Diffuse Horizontal Irradiance (DHI), temperature, wind speed, and rainfall.	Short-term
5	[17]	ANN & Multi-Linear Regression (MLR)	Space inhabitant emission, daily thermal information, and corresponding moisture.	Medium-term
6	[18]	ANN	Solar irradiation data, ambient temperature, relative humidity, wind speed, wind direction, sunshine duration, atmospheric pressure, extraterrestrial horizontal irradiation, solar declination, and Zenith angle.	Short-term
7	[19]	ANN	Thermal information (min and max), air velocity, moisture, condensation and sunlight time.	Short-term

2.2 Fuzzy Logic Control Systems

Rule-based fuzzy logic control systems are increasingly employed to enhance the management and optimization of energy consumption and production across a variety of applications, including industrial facilities, smart grids, and systems harnessing renewable energy [20]. Utilizing fuzzy logic, which adeptly simulates human decision-making by processing ambiguous or uncertain data, these systems are particularly well-suited for complex operational environments. In such contexts, precise and exact input data are often difficult to secure, making traditional control systems less effective. Fuzzy logic's ability to operate within these constraints allows for more adaptive and responsive control strategies, leading to improvements in efficiency and operational reliability.

The structure of fuzzy logic control systems, as shown in Fig. 2, is essential for developing fuzzy rule-based strategies for energy management. The process starts with identifying and categorizing important input and output variables. These variables are grouped into fuzzy sets, with each set defined by specific membership functions. Next, rules that connect the input variables to the outputs are created using "IF-THEN" statements. These rules are then applied in a fuzzy inference process, which evaluates the input values against the fuzzy sets to produce a fuzzy output. This output is made precise through defuzzification methods, converting it into actionable commands.

The final step involves implementing these commands to effectively manage energy usage, allowing the system to adapt to different situations using the flexibility of fuzzy logic. This method provides a practical way to handle uncertain or imprecise data in energy control systems [21].

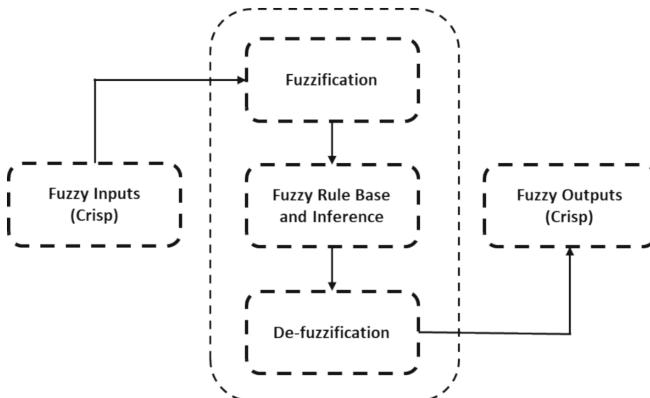


Fig. 2. Fuzzy rule base control system structure.

3 Resources and Techniques

This section explores the industry selected as the case study for this research, focusing on its electricity and energy consumption patterns. It also reviews the input data for the system and the model employed for managing energy consumption.

3.1 Description of an Industrial Process

The case study focuses on the economic integration of renewable solar energy into a meat processing facility, which operates as a centralised factory microgrid. The electricity produced by this system is crucial for powering a variety of equipment including machines, chillers, freezers, and boilers. This setup features a significant solar installation with a capacity of 64 megawatts, supported by an energy storage system with a capacity of 16 gigawatt-hours. The configuration and operational framework of the case study are comprehensively detailed in Fig. 3, which also shows the potential for community integration in the microgrid. This section aims to explore the practical aspects of employing solar energy in industrial applications.

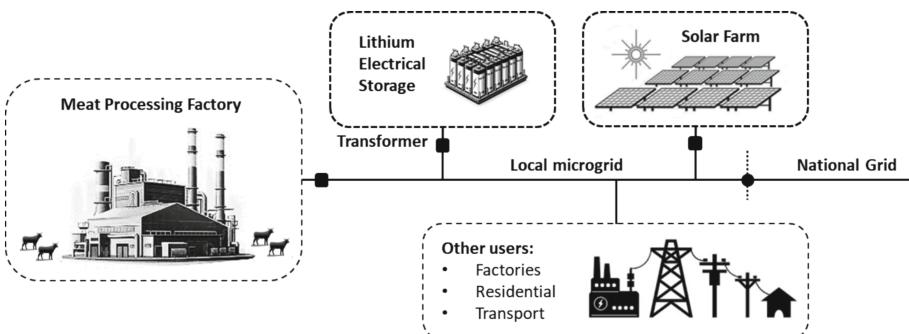


Fig. 3. Integrating a low-carbon factory microgrid with local and national energy networks.

3.2 Description of Data and Models

In this study, two distinct data series have been employed to develop a comprehensive model for managing the electricity consumption within the system. The first dataset comprises weather-related information sourced from the New Zealand open-source National Institute of Water and Atmospheric Research database (NIWA) [22], which will inform predictions of the solar farm's production capacity.

The second dataset pertains to the power usage across various sectors of the meat processing facility, which, when combined with the forecasted data, is utilized to construct a fuzzy logic model aimed at optimizing the utilization of renewable energy resources. While the two datasets differ in their time scales, necessary adjustments have been made to ensure cohesion within the data model. As part of the preprocessing, the time scales of the two datasets were aligned to ensure consistency in the analysis. The collated data encompasses a full-year cycle for 2022, segmented into 30-min intervals.

Numerous options are available for selecting the input data for the initial model aimed at predicting solar global irradiance. These data undergo two distinct filtering processes before the selection of the most relevant variables for the model's input. Initially, based on prior research, variables such as Temperature, Relative Humidity, Wind Speed, and Azimuth are identified as influencing factors on solar radiation. In the subsequent stage,

correlation analysis is employed as a sensitivity analysis tool [13, 18]. For this purpose, Eq. (1) is utilized to compute the correlation between two variables, X and Y.

$$\text{corr}(X, Y) = \frac{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sigma_X \times \sigma_Y} \quad (1)$$

This mathematical formula generates values ranging from -1 to $+1$. Ultimately, variables that exhibit an absolute correlation value greater than 0.4 with the ratio of solar global irradiance are selected as inputs for the MLP model. As illustrated in Fig. 4, Temperature, Wind Speed, and Relative Humidity possess these characteristics and are thus chosen as model inputs.

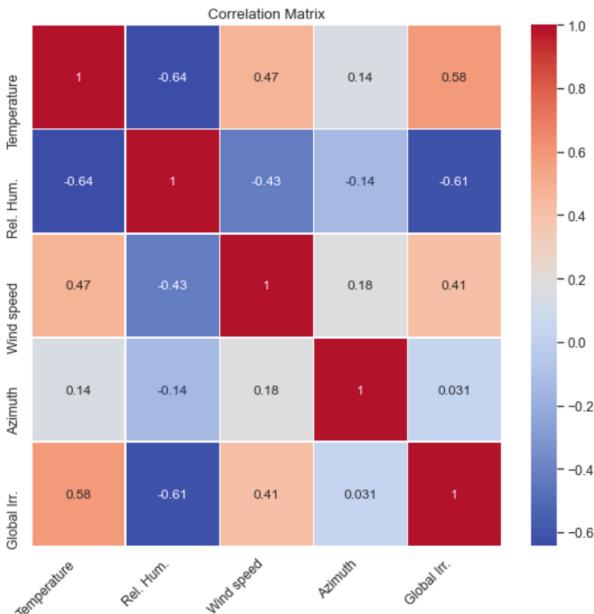


Fig. 4. Correlation matrix of input data for the prediction model.

3.3 MLP Model and Solar Radiation Prediction

Once the inputs for the model are established, the next step involves selecting the appropriate method and architecture for predicting solar radiation. ANN is capable of modeling both linear and nonlinear relationships within systems [18]. Among various ANN approaches, MLP utilizing feed-forward back-propagation has been employed extensively in previous research to estimate solar radiation [23–25].

A challenge in this stage involves specifying the optimal number of hidden layers and the number of neurons within each layer, collectively referred to as hyperparameters. To address this issue, various algorithms have been explored. Notably, Random Search has demonstrated superior efficacy in optimizing these parameters when compared to Manual Search and Grid Search [26], owing to its ability to explore a broader range of potential solutions more efficiently. This methodology has been adopted in this research to fine-tune the MLP model.

The outcomes of this modeling approach, which significantly enhance the predictive accuracy of solar radiation estimations, are illustrated in Fig. 5. In the context of neural networks, "dense" refers to fully connected layers where each neuron is connected to every neuron in the previous layer, enabling complex feature extraction and pattern recognition.

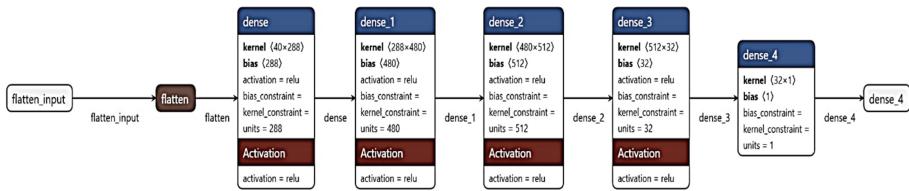


Fig. 5. MLP architecture for prediction of solar radiation.

The optimal configuration identified comprises four hidden layers with the following neuron allocations: 288 neurons in the first layer, 480 in the second, 512 in the third, and 32 in the fourth. This structure emerged as the most suitable for handling the complexities of the dataset used. The efficacy of this model configuration is substantiated by the Mean Squared Error (MSE), recorded at 0.00439, which indicates a high level of precision in solar radiation prediction. The MSE is calculated using Eq. (2).

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (2)$$

Here “ \hat{y}_i ” represents the predicted values, “ y_i ” represents the actual values, and “N” is the total number of observations. A lower MSE value, as seen here, reflects a model that closely aligns with the actual data, thus demonstrating its effectiveness in predicting solar radiation.

3.4 Rule-Based Fuzzy Logic and Electricity Consumption Management

In the fuzzy logic system described, membership functions for solar generation, electrical demand, and storage difference are established based on the maximum observed values from a dataset. Specifically, the maximum solar generation observed is 66,107 kW, and the maximum electrical demand is 19,198 kW. For each of these parameters, an Antecedent variable is created with a range extending from zero to the maximum value, or in the case of storage difference, from a negative to a positive range encompassing

possible fluctuations. As is shown in Fig. 6, solar generation and electrical demand variables are automatically categorized into three fuzzy sets ('Low', 'Medium', and 'High') using triangular membership functions, which are simple and effective for partitioning the input space. In contrast, the storage difference employs trapezoidal membership functions to define 'negative' and 'positive' states, designed to capture the extremes of surplus or deficit in storage relative to production and usage, thus facilitating more precise control actions in response to varying storage conditions.

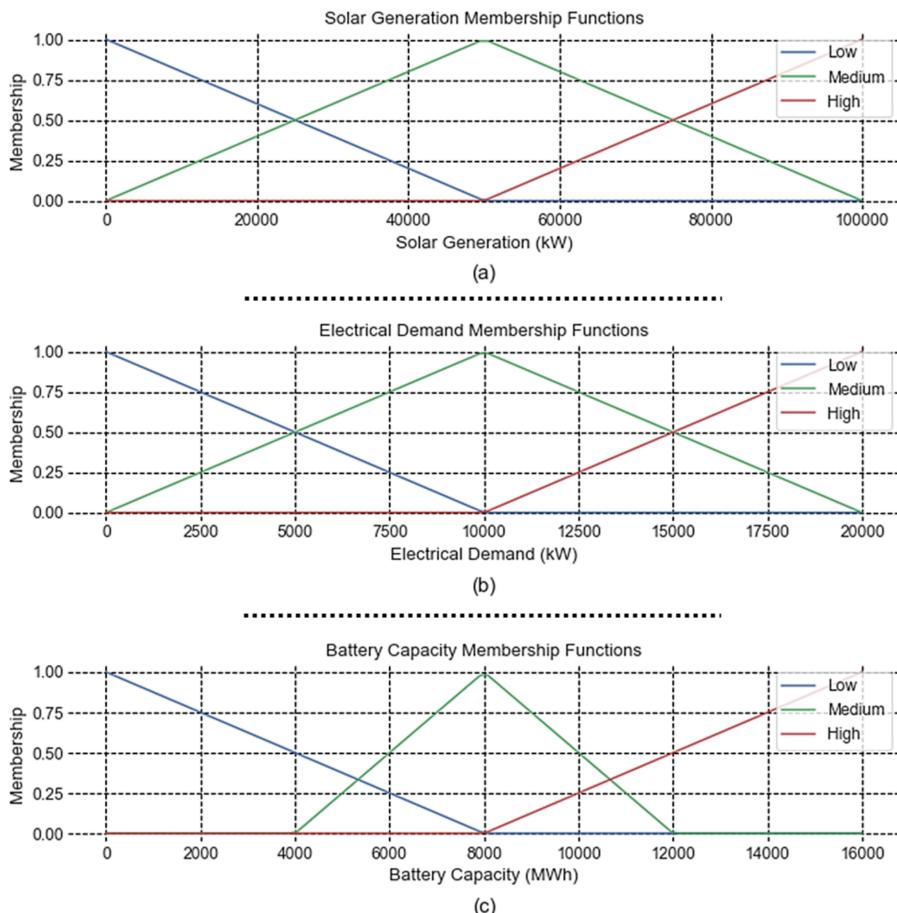


Fig. 6. Membership functions for solar generation, electrical demand, and battery capacity.

After definition of membership functions, fuzzy rules are first established based on predicted solar radiation values, which are then used to calculate the output electricity of the solar farm. The electricity generated, alongside electricity demand from the grid and stored electricity, serves as inputs to the fuzzy system.

These inputs determine the selection of power sources and the potential electrical storage of solar energy in batteries. Initially identifying 27 decision states based on three potential input values—high, medium, or low—these are then consolidated into 12 distinct scenarios through the merging of similar states, as detailed in Table 3. This system prioritizes solar electricity utilization, followed by storage and grid usage, with battery charging occurring predominantly when solar output is high, and electricity demand is low. The outcomes of this approach will be further explored in the following section.

To better understand Table 3 and the system's decisions illustrated in the Fig. 7, a visual representation is utilized. In Fig. 7, each of the three rows in the chart initially indicates the level of solar energy production, followed by detailed information on the rule number, the chosen energy source for that situation, and the final decision on whether to charge the battery. Vertically, each of the three columns represents the electrical load level. Additionally, three different colors are used to depict the amount of electricity stored in the battery.

Table 3. Fuzzy electrical management systems rules.

	Inputs: Status of solar farm, electrical load, and electrical storage			Output: Decision about power source & battery state	
Rule	Solar Generation	Electrical Demand	Battery Capacity	Power Source	Charging Battery
1	Low	Low/Medium	Low	Grid	No
2	Low	Low/Medium	Medium/High	Battery	No
3	Low	High	Low/Medium	Grid	No
4	Low	High	High	Battery	No
5	Medium	Low	Low	Solar	Yes
6	Medium	Low	Medium	Battery	Yes
7	Medium	Low	High	Battery	No
8	Medium	Medium/High	Low/Medium	Solar	No
9	Medium	Medium/High	High	Battery	No
10	High	Low	Low/Medium /High	Solar	Yes
11	High	Low	High	Battery	Yes
12	High	Medium/High	Low/Medium /High	Solar	No

Electrical Load										
L			M			H				
Solar Generation	1	2	2	1	2	2	3	3	4	
	G	B	B	G	B	B	G	G	B	
	-	-	-	-	-	-	-	-	-	
	5	6	7	8	8	9	8	8	0	
M	S	B	B	S	S	B	S	S	B	
	✓	✓	-	-	-	-	-	-	-	
H	10	10	11	12	12	12	12	12	12	
	S	S	B	S	S	S	S	S	S	
	✓	✓	✓	-	-	-	-	-	-	

Rule number

 Energy source:
 • G Grid
 • B Battery
 • S Solar
 Charge Battery?
 • ✓ Yes
 • - No

 : Low Battery State of Charge
 : Medium Battery State of Charge
 : High Battery State of Charge

Fig. 7. The energy source, and battery charging decisions, derived from Table 3, focusing on solar generation, Overall load, and battery state of charge.

4 Results and Discussion

Upon completing the development of both the prediction model and the electricity consumption management system, the outcomes are shown in Fig. 8. Figure 8a displays the decision-making process over the course of a year, illustrating the various strategies adopted based on solar radiation levels: utilizing the electricity grid, directly employing solar energy from the solar farm, using the solar farm to charge the battery, or consuming the stored energy from the battery directly. This variability highlights the adaptive use of different energy sources in response to solar availability.

Figure 8b quantifies the extent of reliance on each source, showing that the national grid was independently used to supply electricity 38.95% of the time, the solar farm was used to charge batteries 38.48% of the time and provided direct electricity for 13.48% of the time, with the remaining 9% of energy consumption coming directly from battery storage.

To further evaluate the effectiveness of the system, Fig. 9 examines a one-week period, focusing on how solar power generation aligns with its use for battery charging and direct supply to industry. The graph demonstrates a well-coordinated management system, efficiently integrating solar production with consumption demands, thereby ensuring that the industry's power needs are met with optimal use of renewable resources. This synchronization not only reflects the system's capacity to manage fluctuating energy inputs but also its ability to maximize the local consumption of locally produced energy and sustainably meet industrial electricity requirements.

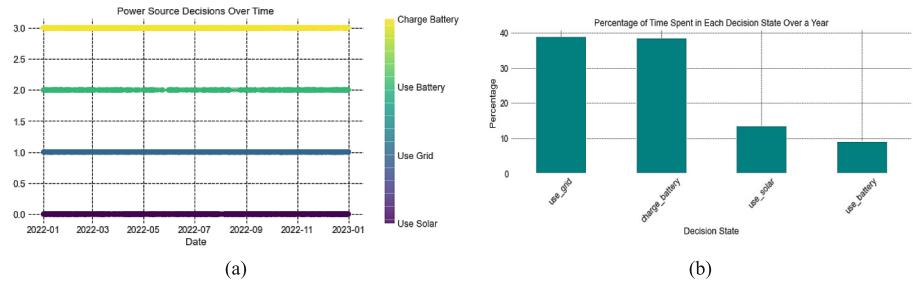


Fig. 8. Evaluating adaptive electricity strategies: (a) Decision dynamics across a yearly cycle (b) Comparative usage analysis of energy sources.

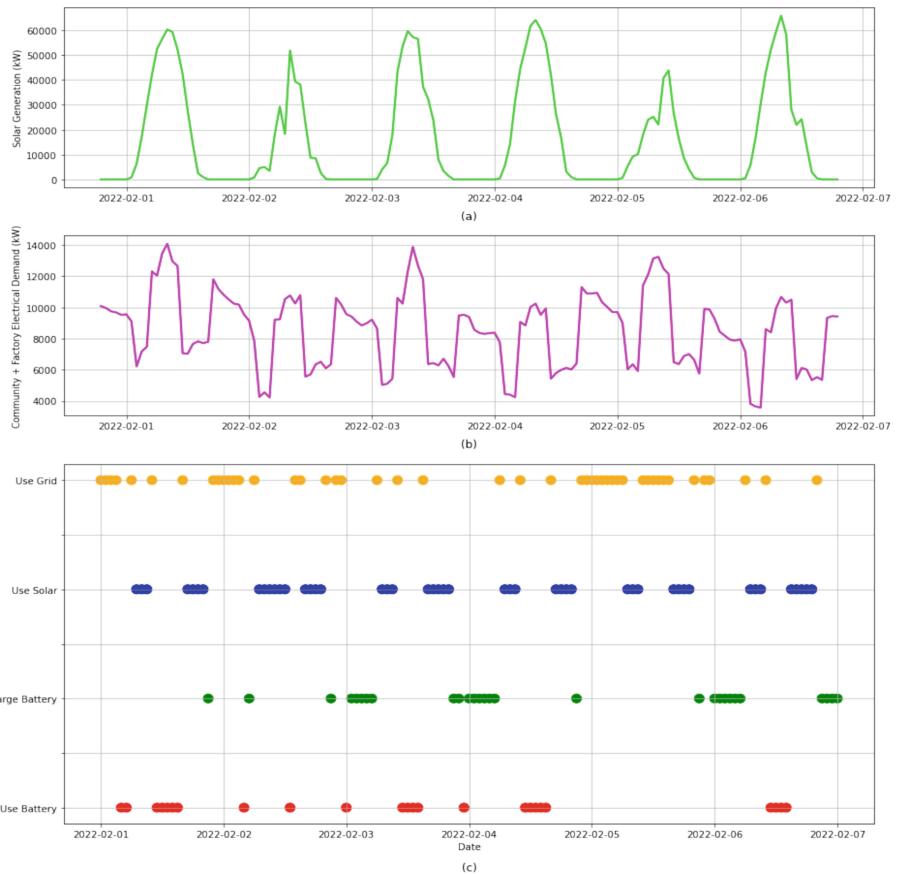


Fig. 9. Evaluating Fuzzy Rule-Based system decisions considering solar generation and electrical demand: (a) Weekly Solar Energy Production (kW) (b) Weekly Electrical Demand (kW) (c) Operational decisions on electricity source and storage charging by the fuzzy system.

5 Conclusion

This study has explored the integration of Artificial Neural Networks (ANNs) and Fuzzy Logic to address the inherent variability of renewable electricity sources, specifically in the meat industry. By utilizing ANNs to predict the impact of weather variations on solar irradiance, and incorporating Fuzzy Logic to refine energy consumption decisions in real-time, the research aimed to reduce reliance on conventional energy grids. The strategies developed focused on optimizing the use of solar-generated electricity, dynamically adjusting between direct consumption, storage, and grid supplementation according to solar availability.

The key findings demonstrate that the combined use of ANNs for solar irradiance prediction and Fuzzy Logic for adaptive energy management results in a more efficient approach to handling renewable electricity within industrial operations. This method successfully improves energy utilization by aligning electricity consumption with solar generation, reducing the facility's dependence on non-renewable grid energy. Although this approach shows promise in efficiently managing electricity sources, the results indicate potential areas for further improvement. The systems implemented represent an initial step towards enhancing sustainable practices within industrial settings. Further refinements to the model, including the improvement of prediction accuracy and energy management strategies, are necessary to fully realize its potential in reducing carbon emissions and advancing environmental sustainability.

Practical implications of this research suggest that the developed model can be applied to other industrial sectors with similar energy demands, provided that relevant data, such as weather and electricity consumption patterns, are available. The ability to switch dynamically between energy sources based on real-time predictions makes this model adaptable to diverse energy management scenarios, particularly in industries striving for renewable electricity integration. For future research, several areas require exploration to enhance the model's performance. First, comparative studies with alternative energy management techniques, such as Model Predictive Control (MPC), would further substantiate the model's effectiveness across different industrial contexts. In addition, exploring the scalability of the system to larger and more complex industrial facilities could provide insights into its broader applicability. Performing sensitivity analysis on input parameters and incorporating additional variables, such as electricity price, could also lead to improvements in both prediction accuracy and management efficiency. Finally, as AI technologies continue to evolve, future work could integrate advanced machine learning methods to further optimize the model's adaptability and performance in managing renewable electricity sources.

References

1. Tito, S.R., Walmsley, M., Apperley, M.: Impact of thermal energy storage on optimal sizing of a grid connected wind-photovoltaic-battery system for an industry edged community. In: 2023 IEEE Fifth International Conference on DC Microgrids (ICDCM), pp. 1–6 (2023). <https://doi.org/10.1109/ICDCM54452.2023.10433591>.

2. Greenhouse gas emissions targets and reporting, Ministry for the Environment. <https://environment.govt.nz/what-government-is-doing/areas-of-work/climate-change/emissions-reductions/emissions-reduction-targets/greenhouse-gas-emissions-targets-and-reporting/>. Accessed 8 Sept 2024
3. Emissions Reduction Plan | Ministry of Business, Innovation & Employment. <https://www.mbie.govt.nz/building-and-energy/energy-and-natural-resources/low-emissions-economy/emissions-reduction-plan>. Accessed 8 Sept 2024
4. Maradin, D.: Advantages and disadvantages of renewable energy sources utilization. *Int. J. Energy Econ. Policy* **11**(3), Art. no. 3 (2021)
5. Necula, S.-C.: Assessing the potential of artificial intelligence in advancing clean energy technologies in europe: a systematic review. *Energies* **16**(22), Art. no. 22 (2023). <https://doi.org/10.3390/en16227633>
6. Suganthi, L., Iriyan, S., Samuel, A.A.: Applications of fuzzy logic in renewable energy systems – a review. *Renew. Sustain. Energy Rev.* **48**, 585–607 (2015). <https://doi.org/10.1016/j.rser.2015.04.037>
7. Zangeneh, M., Aghajari, E., Forouzanfar, M.: A survey: fuzzify parameters and membership function in electrical applications. *Int. J. Dynam. Control* **8** (2020). <https://doi.org/10.1007/s40435-020-00622-1>
8. Eltigani, D., Masri, S.: Challenges of integrating renewable energy sources to smart grids: a review. *Renew. Sustain. Energy Rev.* **52**, 770–780 (2015)
9. What are Neural Networks? | IBM. <https://www.ibm.com/topics/neural-networks>. Accessed 30 Dec 2022
10. Obiora, C.N., Hasan, A.N., Ali, A.: Predicting solar irradiance at several time horizons using machine learning algorithms. *Sustainability* **15**(11), Art. no. 11 (2023). <https://doi.org/10.3390/su15118927>.
11. Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., Steinbrecher, M.: Multi-layer Perceptrons. In: Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., Steinbrecher, M. (eds.) Computational Intelligence: A Methodological Introduction, Texts in Computer Science, pp. 53–124. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-42227-1_5.
12. El-Amarty, N., Marzouq, M., El Fadili, H., Bennani, S.D., Ruano, A.: A comprehensive review of solar irradiation estimation and forecasting using artificial neural networks: data, models and trends. *Environ. Sci. Pollut. Res.* **30**(3), 5407–5439 (2023). <https://doi.org/10.1007/s11356-022-24240-w>
13. Qadir, Z., et al.: Predicting the energy output of hybrid PV–wind renewable energy system using feature selection technique for smart grids. *Energy Reports* **7**, 8465–8475 (2021). <https://doi.org/10.1016/j.egyr.2021.01.018>
14. Rafati, A., Joorabian, M., Mashhour, E., Shaker, H.R.: High dimensional very short-term solar power forecasting based on a data-driven heuristic method. *Energy* **219**, 119647 (2021). <https://doi.org/10.1016/j.energy.2020.119647>
15. Choudhary, A., Pandey, D., Bhardwaj, S.: Global solar radiation estimation modeling using artificial neural network: a case study on metro cities of India. In: Sekhar, G.T.C., Behera, H.S., Nayak, J., Naik, B., Pelusi, D. (eds.) Intelligent Computing in Control and Communication, pp. 479–489. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8439-8_39
16. Sun, L., Sun, Y.: Photovoltaic power forecasting based on artificial neural network and ultraviolet index. *Int. J. Comput.* 153–158 (2022). <https://doi.org/10.4783/ijc.21.2.2583>.
17. Antonopoulos, V.Z., Papamichail, D.M., Aschonitis, V.G., Antonopoulos, A.V.: Solar radiation estimation methods using ANN and empirical models. *Comput. Electron. Agric.* **160**, 160–167 (2019). <https://doi.org/10.1016/j.compag.2019.03.022>
18. Notton, G., Voyant, C., Fouilloy, A., Duchaud, J.L., Nivet, M.L.: Some applications of ANN to solar radiation estimation and forecasting for energy applications. *Appl. Sci.* **9**(1), Art. no. 1 (2019). <https://doi.org/10.3390/app9010209>

19. Kumar, N., Sinha, U.K., Sharma, S.P., Nayak, Y.K.: Prediction of daily global solar radiation using neural networks with improved gain factors and RBF networks. *Int. J. Renew. Energy Res.* **7**(3), Art. no. 3 (2017)
20. Acun, F., Çunkaş, M.: Low-cost fuzzy logic-controlled home energy management system. *J. Electric. Syst. Inform. Technol.* **10**(1), 31 (2023). <https://doi.org/10.1186/s43067-023-00100-6>
21. Alghassab, M.A.: Fuzzy-based smart energy management system for residential buildings in Saudi Arabia: a comparative study. *Energy Reports* **11**, 1212–1224 (2024). <https://doi.org/10.1016/j.egyr.2023.12.039>
22. Solarview. <https://solarview.niwa.co.nz/>. Accessed 23 Apr 2024
23. Manyala, R.: Solar Collectors and Panels: Theory and Applications. BoD – Books on Demand (2010)
24. Mellit, A., Pavan, A.M.: A 24-h forecast of solar irradiance using artificial neural network: application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Solar Energy* **84**(5), 807–821 (2010). <https://doi.org/10.1016/j.solener.2010.02.006>
25. Yildiz, N.: Layered feedforward neural network is relevant to empirical physical formula construction: a theoretical analysis and some simulation results. *Phys. Let. A* **345**(1), 69–87 (2005). <https://doi.org/10.1016/j.physleta.2005.06.116>
26. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(10), 281–305 (2012)



A Dual-Branch Riemannian Learning Network for EEG Speech Imagery Decoding

Liying Zhang¹, Peiliang Gong¹, Qianru Sun¹, Yueying Zhou², Qi Zhu¹,
and Daoqiang Zhang^{1(✉)}

¹ College of Artificial Intelligence and the Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

dqzhang@nuaa.edu.cn

² School of Mathematics Science, Liaocheng University, Liaocheng 252000, China

Abstract. Decoding speech imagery (SI) from electroencephalogram (EEG) signals holds immense promise for individuals with severe speech production deficits. However, existing methods struggle with harnessing the diverse information inherent in EEG signals and mitigating their non-stationary nature. To address these limitations, this paper proposes a dual-branch riemannian network (DBRNet), which integrates parallel Riemannian manifold structures into a deep learning framework to simultaneously capture time-frequency and spatial features of EEG data. First, feature extraction decomposes the input sequence into feature maps representing time-frequency and spatial information. Next, Riemannian structures in the manifold space for each feature type are extracted independently. Then, multi-stage CNN-Transformer modules with feature fusion are employed for deep feature extraction and integration across stages, culminating in the final decoding results. Experimental results on the ASU dataset demonstrate that DBRNet significantly outperforms baseline methods in SI-EEG decoding, underscoring its potential as an advanced approach for this domain.

Keywords: Brain-computer interface · Speech imagery · Deep learning · Riemannian geometry

1 Introduction

Verbal speech is a fundamental mode of human interaction, essential for transmitting information, emotions, and thoughts [1]. However, impairments caused by mental disorders, diseases, accidents, vocal abuse, or brain trauma can significantly diminish one's quality of life and lead to social isolation. A promising alternative for individuals with speech impairments is the Brain-Computer Interface (BCI), which establishes a direct communication channel between the brain and external devices, bypassing the need for vocalization [2,3]. Among

various BCI applications, electroencephalography (EEG) stands out due to its non-invasiveness, high temporal resolution, cost-effectiveness, portability, and versatility [4, 5].

A highly promising BCI paradigm is Speech Imagery (SI), where individuals engage in first-person motor imagery of speaking without moving any articulators [6]. Due to its intuitiveness, SI facilitates more natural control of external devices, thereby improving the usability of BCI systems. By decoding imagined speech activities from EEG signals, researchers can extract linguistic information such as vowels, words, and sentences. This approach offers alternative communication methods for patients with speech expression difficulties while also enhancing our understanding of speech-related EEG signals [7]. These advancements pave the way for improved human-machine interactions and innovative communication solutions.

Current research on SI decoding predominantly falls into two main categories: machine learning-based methods and deep learning-based methods. Machine learning approaches typically leverage spatial-temporal and frequency characteristics to extract meaningful patterns from EEG signals [8, 9]. These approaches often employ classifiers such as Support Vector Machines (SVM) [10], Extreme Learning Machines (ELM) [11], and K-Nearest Neighbors (KNN) [12] for decoding. For example, Kusuma Mohanchandra et al. [10] captured EEG signals during silent reading, compressed the data using subset selection methods, and employed multi-class SVM for classification. Similarly, another study utilized a sparse regression model for feature selection to reduce the dimensionality of EEG feature vectors, followed by the application of the ELM algorithm for decoding [11]. These foundational techniques in feature extraction and classification establish the basic framework for SI decoding. However, they cannot fully adapt to the highly stochastic nature of EEG signals, which are easily affected by factors such as emotional changes and psychological states [13], making it difficult to improve decoding performance.

To address the non-stationarity issue of EEG signals, feature extraction methods based on Riemannian geometry have been proposed. These methods mitigate the impact of the linear mix effect [14] in EEG sequences during covariance distance calculations and counteract the inflation effect produced by the Symmetric Positive Definite (SPD) features of these matrices [15, 16], which can negatively affect classification accuracy. Moreover, the effectiveness of Riemannian geometry in separating covariance feature distances has been extensively validated in numerous studies [2, 17]. For example, Nguyen et al. [2] designed a description strategy based on the covariance matrices in Riemannian manifolds. They mapped the autocorrelation matrices of EEG sequences into the tangent space as features and employed SVM for classification. Another study [17] calculated the Riemannian distance of the Correntropy Spectral Density of EEG signals as classification features and used the KNN for classification. However, most Riemannian geometry-based methods are embedded within a machine learning framework, necessitating the utilization of manual feature selection. This may limit the model's ability to uncover high-order speech imagery features in EEG signals.

Deep learning-based decoding methods typically offer more flexibility and automation in feature extraction and classification. These methods include foundational models such as Convolutional Neural Networks (CNN) [18], sequential networks like Gated Recurrent Units [19] and Long Short-Term Memory (LSTM) [20], hybrid models like CNN-LSTM [21], and complex ensemble structures such as attention modules [22] and Transformers [23]. For example, Saha et al. [21] proposed an EEG speech imagery recognition method that independently trains autoencoders, CNNs, and LSTMs, then combines them for inference. Lee et al. [23] designed a specialized EEG-Transformer model for speech imagery decoding. Despite the significant progress gained by these approaches, it's worth noting that the geometric structure of EEG signals does not fully adhere to the Euclidean space typically assumed in conventional deep learning models [15, 24]. This discrepancy can lead to reduced performance in these models. In related areas of EEG signal processing fields, such as motion imagery classification, researchers have incorporated Riemannian geometry into existing deep learning frameworks to address similar challenges, such as TensorCSPnet [25] and RGDDANet [13].

While these approaches can be directly applied to SI-decoding tasks, the EEG signals elicited by these two tasks are significantly different. Specifically: (1) The primary mechanisms for motion imagery signals are event-related desynchronization and event-related synchronization, which are more readily identifiable [26]. (2) The frequency bands where features are concentrated differ between the two modalities [27]. Moreover, the integration of Riemannian structures into feature extraction is often relatively simplistic and lacks effectiveness in handling many types of features (such as temporal, spatial, and frequency) in these models. This limitation may reduce the potential of Riemannian structures in stabilizing learning across different types of features. Consequently, these models do not exhibit superior decoding performance in SI-decoding tasks and still show a performance gap compared to dedicated SI-decoding models (see Table 2 below).

To address the aforementioned challenges, we propose the Dual-Branch Riemannian Network (DBRNet) for SI decoding. DBRNet integrates Riemannian manifold structures into the deep learning framework to manage the nonstationarity of EEG speech imagery signals in Euclidean space. It enhances decoding performance by applying Riemannian structures in parallel branches to simultaneously learn the time-frequency and spatial features. Specifically, we first preprocess the input sequence to decompose it into feature maps containing time-frequency features and spatial features. Then, Riemannian structures perform feature learning in the manifold space for each feature type separately. To couple local and global features, we incorporate CNN-Transformer modules with feature fusion for deep feature extraction at each stage. CNN excels at capturing local spatial patterns, while Transformer effectively models long-range dependencies and global context. This combination enhances the network's ability to understand intricate patterns in the data, ultimately yielding more accurate and robust decoding results. Furthermore, we conducted benchmark and abla-

tion experiments on the Arizona State University (ASU) dataset [2] to assess the performance of the DBRNet method.

This paper highlights the following significant contributions:

- (1) We propose a Dual-Branch Riemannian Deep Learning Network named DBRNet for SI decoding to address the non-stationarity in speech imagery EEG signals. To the best of our knowledge, this is the first time that Riemannian manifolds are integrated with deep learning to decode SI.
- (2) DBRNet incorporates parallel Riemannian manifold structures within the network, demonstrating their efficient performance in handling both time-frequency and spatial EEG features.
- (3) Experimental results on the ASU dataset for EEG SI decoding show that our DBRNet has superior performance.

The rest of this manuscript is organized as follows: Sect. 2 details the design of the proposed DBRNet. Section 3 describes our experiments on relevant datasets and analyzes the results. Section 4 concludes the manuscript.

2 Methodology

In this section, we propose a novel DBRNet to leverage the stabilizing properties of the Riemannian manifold for different types of features in EEG speech imagery signals. DBRNet consists of two branches, each comprising four main components: feature extraction (time-frequency feature and spatial feature), Riemannian embedding and propagation, a parallel CNN-Transformer processing module, and a classification head. The overall structure of DBRNet is shown in Fig. 1.

2.1 Feature Extraction

Considering that initial EEG signals generally contain both time-frequency and spatial features, we have designed two processing methods to separately extract the time-frequency and spatial features of the EEG signal sequences. The specific processing methods are illustrated in Fig. 2.

Time-Frequency Feature: In this module, we aim to present features for different frequency bands and time segments in more detail. We segment the input sequence in both time domain and frequency bands. The input sequence is characterized by a shape of (C, S) , where C denotes the number of channels and S represents the number of sampling points. First, we use a Chebyshev Type II filter bank to segment the input sequence into different frequency bands. The total frequency range is 0–40 Hz, divided equally into M segments, with M set to 10 by default. For time domain segmentation, we retain the middle segment of the sampled data from each trial, discarding the first and last 10% of the length. This sequence is then divided into N equal parts, where N is set to 8 by default. The final processed sequence shape is $[N * M, C, T/N]$, where T is the middle segment.

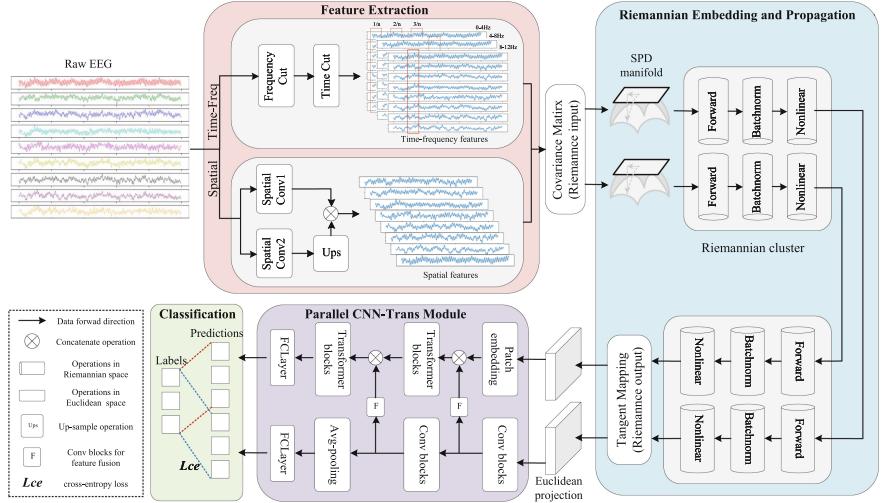


Fig. 1. Overview of DBRNet. The raw EEG signals are directed into a dual-branch feature extraction module, with one branch extracting time-frequency features and the other focusing on spatial features. Subsequently, Riemannian geometry is applied to both branches for feature learning within the manifold space. Then, a parallel CNN-Transformer module conducts nuanced feature fusion, enabling systematic, stage-by-stage deep feature exploration and culminating in precise decoding outcomes.

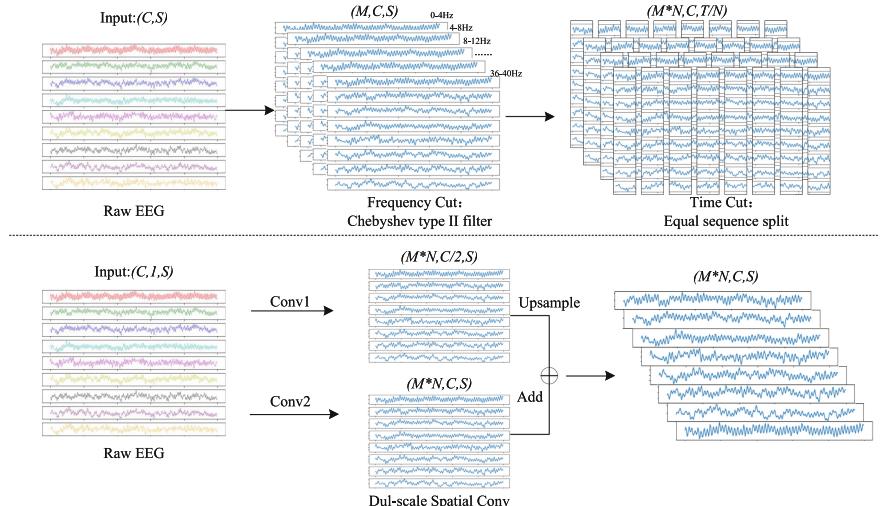


Fig. 2. Preprocessing Methods for Acquiring Time-Frequency and Spatial features.

Spatial Feature: In this module, we aim to represent EEG signals features considering their positions on the cerebral cortex (electrode points). The Common

Spatial Pattern (CSP) algorithm [13] is a typical method for extracting spatial features through spatial filters. While effective, the CSP algorithm is restricted to binary analysis and trained independently of the main network [13]. Another common approach is spatial convolution [16, 28], which participates in the training process of the main network and extracts spatial features through backpropagation updates. Conventional spatial convolution methods set the total number of channels as the convolution kernel size, introducing a large number of learnable parameters and potentially neglecting localized spatial features. To address these issues, we use two different convolution kernel sizes for spatial convolution simultaneously, resize their results to the same shape, and then fuse to obtain the output of this module.

2.2 Feature Embedding in Riemannian Structure

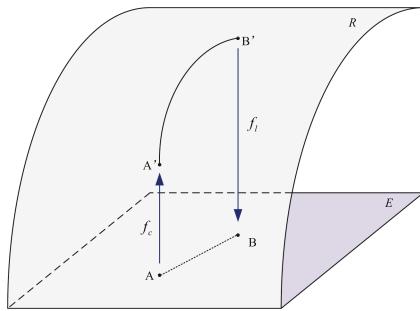


Fig. 3. Concept of Riemannian Manifold.

Figure 3 illustrates the basic principles of the Riemannian structure. Following the work of [25, 29], we project features from Euclidean space E onto the Riemannian manifold R, perform forward propagation computations within this manifold, finally projecting the results back into Euclidean space. This process involves five layers:

a. Input Layer: Project the feature layer from Euclidean space to the manifold in Riemannian space by computing the covariance matrix of the feature layer:

$$X_o = f_i(X_i) = \frac{1}{L} X_i X_i^T \quad (1)$$

where f_c is the definition of the input layer, X_i is the input feature sequence, and L is the sequence length.

b. Linear Forward Layer: Use the bilinear transformation to generate a new SPD matrix that simulates the forward propagation of nodes in a conventional neural network, as follows:

$$X_o = f_l(X_i; W_k) = W_k X_i W_k^T \quad (2)$$

In the equation, f_l represents the linear forward layer, and W is the transformation parameter in the bilinear mapping analogous to the weights in conventional neural layers.

c. Nonlinear Layer: Apply the Reig nonlinearity $\max(I, \Sigma)$ [29] to the eigenvalues obtained from the eigendecomposition of the covariance matrix, calculated as follows:

$$X_i = U \cdot \Sigma \cdot U^T \quad (3)$$

$$X_o = f_{nl}(X_i) = U \cdot \max(\epsilon I, \Sigma) \cdot U^T \quad (4)$$

where f_{nl} represents the nonlinear layer, U is the symmetric matrix obtained from the eigendecomposition, Σ is the diagonal matrix of eigenvalues, and I is the identity matrix.

d. Normalization Layer: Referencing the work [25], normalization is achieved by left and right multiplying the covariance matrix by the matrix B and matrix G , respectively. B is the weighted Riemannian mean of all sample points in the minibatch, representing the central or average point of the batch data. G is learned during the training process and is used to adjust and compensate for scale transformations [25]. This operation can be described as:

$$X_o = f_{nm}(X_i) = G^{1/2} \cdot B^{-1/2} \cdot X_i \cdot B^{-1/2} \cdot G^{1/2} \quad (5)$$

where f_{nm} represents the normalization layer.

e. Output Layer: Apply the logarithm to the eigenvalues obtained from the eigendecomposition of the input sequence features [29], projecting the results from Riemannian layers back to Euclidean space, as:

$$X_o = f_o(X_i) = \log(X_i) = U \log(\Sigma) U^T \quad (6)$$

where f_o represents the output layer.

We use the Riemannian layer in both Time-Frequency Branch and Spatial Branch to learn time-frequency features and spatial features, respectively. In the Time-Frequency Branch, we use two successive Riemannian layers in the following order $a \rightarrow bcd \rightarrow bcd \rightarrow e$. In the Spatial Branch, we use a single Riemannian layer in the order: $a \rightarrow bcd \rightarrow e$.

2.3 Parallel CNN-Trans Module

The parallel CNN-Transformer module comprises two branches, Branch-A and Branch-B, which together progress through three stages with specific parameters as shown in Table 1. Branch-A primarily processes time and frequency features by leveraging convolutional networks on the time channel to exploit local advantages [28]. In contrast, Branch-B focuses on a broader range of feature types. At each stage, Branch-B receives outputs from both Branch-A and Branch-B of the preceding stage. After merging these outputs, a 1×1 convolution is applied to adjust the number of channels and segment patches, which are then fed into

a multi-block Transformer. The self-attention mechanism in Branch-B performs weight adjustment and filtering for multiple feature types. Additionally, we construct convolutional network clusters for feature fusion, which process the results from each stage of Branch-A and subsequently pass them to Branch-B.

Table 1. Main parameters of the CNN-Trans parallel module (Using the ASU dataset as an example)

Stage	Layer name	Settings	In/Output size	Describe
1	Path embedding	Kernal size: (32,1)	(b,c0,32,32) (b,c1,32)	Conv layers for dividing patch
1	Transformer block	Embedding size: 32	(b,c1,32) (b,c1,32)	Transformer blocks in Branch-B
1	Conv cluster	Kernal size: (1,2) Stride: (1,2)	(b,c0,32,32) (b,c1,32,16)	Conv, relu and batchnorm layers in Branch-A
1	Fusionconv cluser	Kernal size: (3,3)	(b,c1,32,16) (b,c1,32,16)	Conv, relu and batchnorm layers in feature fusion
1	Fusion pooling	Global average pooling	(b,c1,32,16) (b,c1,32)	Pooling layer in feature fusion
2	Transformer block	Embedding size: 32	(b,c1,32) (b,c1,32)	Transformer blocks in Branch-B
2	Conv cluster	Kernal size: (1,2) Stride: (1,2)	(b,c1,32,32) (b,c1,32,16)	Conv, relu and batchnorm layers in Branch-A
2	Fusion conv cluser	Kernal size: (3,3)	(b,c1,32,16) (b,c1,32,16)	Conv, relu and batchnorm layers in feature fusion
2	Fusion pooling	Global average pooling	(b,c1,32,16) (b,c1,32)	Pooling layer in feature fusion
3	Transformer block	Embedding size: 32	(b,c1,32) (b,c1,32)	Transformer blocks in Branch-B
3	Conv cluster	Kernal size: (1,2) Stride: (1,2)	(b,c1,32,32) (b,c1,32,16)	Conv, relu and batchnorm layers in Branch-A

2.4 Classification Heads and Loss Functions

In DBRNet, both Time-Frequency Branch and Spatial Branch use independent classifiers to produce results and jointly compute the loss. The classifier for

Time-Frequency Branch consists of a global average pooling layer followed by two Multilayer Perceptron (MLP) layers with dimensions adjusted to $(16, n_c)$, where n_c is the number of classes. The classifier for Spatial Branch flattens the feature layer and then inputs it into three successive MLP layers, with dimensions adjusted to $(b, 32)$, $(b, 32)$, and (b, n_c) respectively, where b is the batch size. For both branches, ReLU layers are used to introduce nonlinearity in all MLP layers except the last one.

The loss for each branch is calculated using the cross-entropy loss function, which is defined as:

$$L_{ba} = - \sum_i^{n_c} y_i \cdot \log(\hat{y}_{a,i}) \quad (7)$$

$$L_{bb} = - \sum_i^{n_c} y_i \cdot \log(\hat{y}_{b,i}) \quad (8)$$

where L_{ba} and L_{bb} are the loss functions of Branch-A and Branch-B, respectively, and $\hat{y}_{a,i}$ and $\hat{y}_{b,i}$ are the outputs of Branch-A and Branch-B, respectively. y_i is the true label.

The loss function of the entire DBRNet is the weighted sum of the two branches, with weighting coefficients of α and $1-\alpha$, respectively. This is expressed as:

$$L = \alpha L_{ba} + (1 - \alpha) L_{bb} \quad (9)$$

3 Experiment

3.1 Experimental Settings

Dataset: Our experiments were conducted on the ASU dataset [2]. It records various types of speech imagery signals collected from 15 subjects, including long words (“cooperate” and “independent”), short words (“in”, “out”, and “up”), and vowels (“/a/”, “/i/”, and “/u/”). The recording process of this dataset includes both formal trial sessions and reset sessions. The trial sessions consist of several periods. In each period, subjects are instructed complete speech imagery tasks and the signals from 64 EEG electrodes are recorded. For long words, the trial period length is 1.4 s, with 6 subjects collecting 1200 trials. Short words and vowels are presented for a shorter trial period of 1 s, with 6 subjects collecting 1120 trials (for short words) and 8 subjects collecting 2400 trials (for vowels), respectively. We remove the reset sessions from the dataset. Then, the remaining portion is divided, with 70% used for training and 30% for testing.

Evaluation Metrics: We use Accuracy (Acc) and Precision (Pre) to evaluate our model.

Implementation Details: We constructed the DBRNet under the PyTorch deep learning framework, using an RTX3090 GPU for the task. The model was trained for a total of 200 epochs, using the Riemannian Adam optimizer from geoopt [30] with an initial learning rate of 0.001. We also applied a learning rate

decay strategy, where the learning rate was reduced to a quarter of its initial value after every 50 epochs of training. The weighting coefficient α of the loss function is set to 0.3. Additionally, the machine learning methods compared in this manuscript were implemented in Matlab, while other deep learning methods were configured according to their open-sourced codes.

Methods Compared: In our experiments, we compared our approach with four types of methods to highlight the advantages of DBRNet.

- **a.** Machine learning methods: such as SVM [10], LDA [31].
- **b.** Generic EEG deep learning methods, such as LSTM [20], EEGConformer [28], Tception [32].
- **c.** Works combining deep learning with Riemannian manifold in other EEG domains, such as TensorCSPnet [25], GraphCSPnet [33].
- **d.** Recent representative works in EEG SI-decoding, like work [27].

3.2 Experimental Result

Comparison with Other Methods: We conducted comparative experiments between DBRNet and other methods on three categories of speech data (vowels, short words, and long words). Table 2 and Table 3 illustrate the results of the comparative experiments.

Table 2. Accuracy comparison of DBRNet and other methods.

	Vowels (%)	Short Words (%)	Long Words (%)
SVM [10]	38.30	37.89	58.33
LDA [31]	37.00	35.67	56.67
LSTM [20]	38.67	38.44	58.00
EEGconformer [28]	42.83	44.66	61.50
TSeption [32]	44.11	43.18	62.67
TensorCSPnet [25]	57.33	63.16	64.27
GraphCSPnet [33]	49.93	55.17	62.06
Work [27]	-	-	70.20
DBRNet(ours)	65.38	69.95	77.60

As shown in Tables 2 and 3, DBRNet has a certain advantage over several other methods. The Acc and Pre of DBRNet for vowels, short words and long words reached 65.38%, 69.95%, 77.6% and 73.34%, 71.52%, and 77.84%, respectively. This significantly outperforms classical machine learning methods (SVM, LDA) as well as general EEG deep learning methods (LSTM, EEGConformer). For TensorCSPnet and GraphCSPnet, which also use a combination of deep learning and Riemannian manifold similar to DBRNet, their performance is

Table 3. Precision comparison of DBRNet and other methods.

	Vowels (%)	Short Words (%)	Long Words (%)
SVM [10]	38.57	36.00	56.35
LDA [31]	37.02	35.50	57.18
LSTM [20]	40.54	38.67	58.12
EEGConformer [28]	43.03	44.97	60.95
TSeption [32]	44.11	43.18	62.67
TensorCSPnet [25]	57.38	64.40	64.36
GraphCSPnet [33]	50.96	63.89	64.10
DBRNet(ours)	73.34	71.52	77.84

suboptimal due to not being well-suited for speech imagery tasks and lacking fine-tuning. We also compare our work with a recent study in the field that introduced additional multimodal data to take advantage of complementary benefits. In contrast, DBRNet does not use any additional data information but still has a certain performance advantage. This can be attributed to DBRNet’s effective extraction and utilization of the inherent time-frequency-spatial information within the signals.

For a better demonstration, we utilized t-distributed stochastic neighbor embedding (t-SNE) [34] to visualize the outputs of the final layers from EEG-Conformer, TensorCSPnet, and our model. T-SNE visualizes high-dimensional data in lower dimensions to reveal its characteristics. In Fig. 4, yellow, green, red, and purple dots represent the short words “in”, “out”, and “up”, respectively. In Fig. 4(a) and (b), the lack of a clear classification boundary makes it challenging to discriminate between the samples of different classes. In contrast, Fig. 4(c) for DBRNet shows distinct clustering of features for each class with minimal overlap, demonstrating superior feature extraction and discrimination capabilities.

To further illustrate the underlying characteristics learned from the data, we employed Class Activation Topography (CAT) [28]. CAT is a novel visualization method that combines EEG topography with Class Activation Mapping (CAM) by multiplying normalized EEG data with normalized CAM. This method highlights the spatial distribution of the most informative features learned by our model. As shown in Fig. 5, during speech imagery tasks, CAT analysis revealed significant activations in electrodes corresponding to Broca’s area (F7, F5, and F3) and Wernicke’s area (T7, TP7, and P7), which are essential for language production and comprehension. These activations provide insights into the specific brain regions engaged during the task, aligning with established neuroscientific theories of speech imagery [35].

Ablation Study: To further explore the role of different structures in DBRNet, we made the following modifications to the original DBRNet: (1) retaining only Spatial Branch, (2) retaining only Time-Frequency Branch, (3) replacing

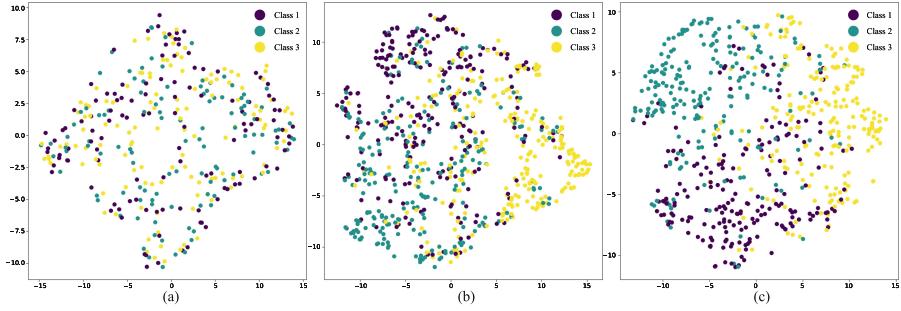


Fig. 4. Comparison of t-SNE visualization with other methods: (a) EEGConformer, (b) TensorCSPnet, (c) DBRNet.

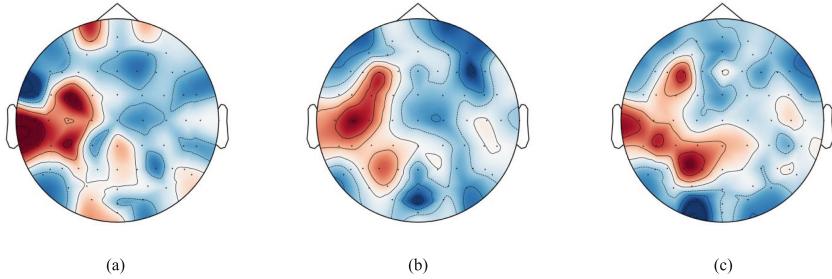


Fig. 5. Topography maps of CAM-weighted EEG on ASU dataset.: (a) Long words, (b) Short words, (c) Vowels.

Table 4. Accuracy of DBRNet under different ablation experiments.

Method	(1)	(2)	(3)	(4)	ACC (%)
Spatial-only	✓	-	-	✓	54.26
Time-Frequency-only	-	✓	-	✓	51.94
Riemannian-removed	-	-	✓	-	56.72
Fusion-removed	-	-	-	✓	62.25
DBRNet	✓	✓	✓	✓	65.38

the Riemannian layer with convolutional layers, (4) removing the feature fusion layer in the middle of the CNN-Trans module. The performance of the modified DBRNet on the vowel data is shown in the Table 4 below:

As shown in Table 4, in experiments (1) and (2), directly removing the composite branch caused the model to lose its ability to process multiple types of features, resulting in significant performance degradation. In Experiment (3), replacing the Riemannian layer with convolutional layers completely eliminated DBRNet's ability to learn features on the Riemannian manifold, which also caused significant performance degradation. In contrast, the changes in exper-

iments (4) and (5) had a relatively small impact on DBRNet, suggesting that the model is not sensitive to the stacking of cnn-trans stages. Experiment (6), which removed the information interaction between the two branches, weakened the feature selection role of the self-attention mechanism in Branch-B, thereby affecting performance.

4 Conclusion

This study presents DBRNet, an advanced deep-learning architecture tailored for decoding EEG-based speech imagery. It introduces the Riemannian manifold into the conventional deep learning framework to address the non-stationarity problem of EEG speech imagery signals in non-Euclidean space. DBRNet incorporates a Riemannian structure composed branch and completes the decoding process through a multi-stage CNN-Trans module with feature fusion. Experimental results on the ASU dataset show that DBRNet has a significant performance advantage over various types of methods. Looking ahead, the exceptional performance of our method in SI decoding is an important step towards better BCI application and neuroscience research.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Nos. 62276130, 62136004), by the National Key R&D Program of China (Grant No. 2023YFF1204803), and also by the Key Research and Development Plan of Jiangsu Province (No. BE2022842).

References

- Blank, S.C., Scott, S.K., Murphy, K., Warburton, E., Wise, R.J.: Speech production: Wernicke, Broca and beyond. *Brain* **125**(8), 1829–1838 (2002)
- Nguyen, C.H., Karavas, G.K., Artemiadis, P.: Inferring imagined speech using EEG signals: a new approach using Riemannian manifold features. *J. Neural Eng.* **15**(1), 016002 (2017)
- Sereshkeh, A.R., Trott, R., Bricout, A., Chau, T.: EEG classification of covert speech using regularized neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **25**(12), 2292–2300 (2017)
- Li, B., Cheng, T., Guo, Z.: A review of EEG acquisition, processing and application. *J. Phys. Conf. Ser.* **1907**, 012045 (2021)
- Orban, M., Elsamanty, M., Guo, K., Zhang, S., Yang, H.: A review of brain activity and EEG-based brain-computer interfaces for rehabilitation application. *Bioengineering* **9**(12), 768 (2022)
- Martin, S., Iturrate, I., Millán, J.D.R., Knight, R.T., Pasley, B.N.: Decoding inner speech using electrocorticography: progress and challenges toward a speech prosthesis. *Front. Neurosci.* **12**, 367292 (2018)
- Alderson-Day, B., Fernyhough, C.: Inner speech: development, cognitive functions, phenomenology, and neurobiology. *Psychol. Bull.* **141**(5), 931 (2015)
- Wang, L., Zhang, X., Zhong, X., Zhang, Yu.: Analysis and classification of speech imagery EEG for BCI. *Biomed. Sig. Process. Control* **8**(6), 901–908 (2013)

9. Kim, J., Lee, S.-K., Lee, B.: EEG classification in a single-trial basis for vowel speech perception using multivariate empirical mode decomposition. *J. Neural Eng.* **11**(3), 036010 (2014)
10. Mohanchandra, K., Saha, S.: A communication paradigm using subvocalized speech: translating brain signals into speech. *Augmented Hum. Res.* **1**(1), 3 (2016)
11. Min, B., Kim, J., Park, H.J., Lee, B., et al.: Vowel imagery decoding toward silent speech BCI using extreme learning machine with electroencephalogram. *BioMed Res. Int.* **2016**, 2618265 (2016)
12. Hashim, N., Ali, A., Mohd-Isa, W.N.: Word-based classification of imagined speech using EEG. In: *Computational Science and Technology: 4th ICCST 2017*, Kuala Lumpur, Malaysia, 29–30 November, 2017, pp. 195–204. Springer (2018)
13. Liu, W., Guo, C., Gao, C.: A cross-session motor imagery classification method based on Riemannian geometry and deep domain adaptation. *Expert Syst. Appl.* **237**, 121612 (2024)
14. Khadem, A., Hosseini-Zadeh, G.-A.: Quantification of the effects of volume conduction on the EEG/MEG connectivity estimates: an index of sensitivity to brain interactions. *Physiol. Meas.* **35**(10), 2149 (2014)
15. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.* **29**(1), 328–347 (2007)
16. Zhang, G., Etemad, A.: Spatio-temporal EEG representation learning on Riemannian manifold and Euclidean space. *IEEE Trans. Emerg. Top. Comput. Intell.* **8**(2), 1469–1483 (2023)
17. Bakhshali, M.A., Khademi, M., Ebrahimi-Moghadam, A., Moghimi, S.: EEG signal classification of imagined speech based on Riemannian distance of correntropy spectral density. *Biomed. Sig. Process. Control* **59**, 101899 (2020)
18. Cooney, C., Korik, A., Folli, R., Coyle, D.: Evaluation of hyperparameter optimization in machine and deep learning methods for decoding imagined speech EEG. *Sensors* **20**(16), 4629 (2020)
19. Jiménez-Guarneros, M., Gómez-Gil, P.: Standardization-refinement domain adaptation method for cross-subject EEG-based classification in imagined speech recognition. *Pattern Recogn. Lett.* **141**, 54–60 (2021)
20. Gasparini, F., Cazzaniga, E., Saibene, A.: Inner speech recognition through electroencephalographic signals. arXiv preprint [arXiv:2210.06472](https://arxiv.org/abs/2210.06472) (2022)
21. Saha, P., Fels, S., Abdul-Mageed, M.: Deep learning the EEG manifold for phonological categorization from active thoughts. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2762–2766. IEEE (2019)
22. Lee, D.H., Kim, S.J., Lee, K.W.: Decoding high-level imagined speech using attention-based deep neural networks. In: *2022 10th International Winter Conference on Brain-Computer Interface (BCI)*, pp. 1–4. IEEE (2022)
23. Lee, Y.E., Lee, S.H.: EEG-transformer: self-attention from transformer architecture for decoding EEG of imagined speech. In: *2022 10th International Winter Conference on Brain-Computer Interface (BCI)*, pp. 1–4. IEEE (2022)
24. Kalunga, E.K., Chevallier, S., Barthélémy, Q., Djouani, K., Hamam, Y., Monacelli, E.: From Euclidean to Riemannian means: information geometry for SSVEP classification. In: *Geometric Science of Information: Second International Conference, GSI 2015, Palaiseau, France, October 28–30, 2015, Proceedings 2*, pp. 595–604. Springer (2015)

25. Ju, C., Guan, C.: Tensor-CSPNet: a novel geometric deep learning framework for motor imagery classification. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(12), 10955–10969 (2022)
26. Gert Pfurtscheller, Ch., Neuper, D.F., Pregenzer, M.: EEG-based discrimination between imagination of right and left hand movement. *Electroencephalogr. Clin. Neurophysiol.* **103**(6), 642–651 (1997)
27. Ahn, H.-J., Lee, D.-H., Jeong, J.-H., Lee, S.-W.: Multiscale convolutional transformer for EEG classification of mental imagery in different modalities. *IEEE Trans. Neural Syst. Rehabil. Eng.* **31**, 646–656 (2022)
28. Song, Y., Zheng, Q., Liu, B., Gao, X.: EEG conformer: convolutional transformer for EEG decoding and visualization. *IEEE Trans. Neural Syst. Rehabil. Eng.* **31**, 710–719 (2022)
29. Huang, Z., Van Gool, L.: A Riemannian network for SPD matrix learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31 (2017)
30. Kochurov, M., Karimov, R., Kozlukov, S.: Geoopt: Riemannian optimization in PyTorch. *arXiv preprint arXiv:2005.02819* (2020)
31. Subasi, A., Ismail Gursoy, M.: EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Syst. Appl.* **37**(12), 8659–8666 (2010)
32. Ding, Y., et al.: TSception: a deep learning framework for emotion detection using EEG. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE (2020)
33. Ju, C., Guan, C.: Graph neural networks on SPD manifolds for motor imagery classification: a perspective from the time–frequency analysis. *IEEE Trans. Neural Netw. Learn. Syst.* (2023)
34. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11) (2008)
35. Cooney, C., Folli, R., Coyle, D.: Neurolinguistics research advancing development of a direct-speech brain-computer interface. *IScience* **8**, 103–125 (2018)



ROSAL: Semi-supervised Active Learning with Representation Aggregation and Outlier for Endoscopy Image Classification

Xiaocong Huang¹, Guoheng Huang^{1(✉)}, Guo Zhong², Xiaochen Yuan³, Xuhang Chen⁴, Chi-Man Pun⁵, and Jianwu Chen⁶

¹ Guangdong University of Technology, Guangzhou, China

kevinwong@gdut.edu.cn

² Guangdong University of Foreign Studies, Guangzhou, China

³ Macao Polytechnic University, Macao, China

⁴ Huizhou University, Huizhou, China

⁵ University of Macao, Macao, China

⁶ Fujian Medical University Union Hospital, Fujian, China

Abstract. The classification of endoscopy images is vital for early detection and prevention of Colorectal Cancer (CRC). However, manual annotation of these images is expensive. Semi-supervised Active Learning (SAL) can help reduce costs, but issues with the accuracy of pseudo-labels and the tendency to over-select outliers remain. To address these, we introduce ROSAL, a new SAL framework featuring Representational Correlation-based Pseudo-label Training (RCPT) and Outlier-based Hybrid Querying (OHQ). RCPT employs a pseudo-label contrastive loss to enhance agreement among unlabeled data representations and reduce discord. The pseudo-label generator in RCPT leverages this correlation for more precise labeling. OHQ introduces a distance factor to minimize outlier selection through a hybrid querying strategy. Experimental results demonstrate that ROSAL outperforms other active learning methods, achieving 71.46% and 90.79% accuracy on a publicly available endoscopic dataset and a publicly available natural image dataset, respectively, using only 40% and 20% of the labeled data.

Keywords: Semi-supervised learning · Active learning · Contrast learning · Endoscopy image classification

1 Introduction

Colorectal Cancer (CRC) is one of the most common cancers affecting human health worldwide and one of the leading causes of cancer-related deaths [30]. Artificial intelligence (AI) models are now widely used in the medical imaging field [5, 9, 13, 15, 20, 21, 31, 37, 38]. Endoscopic examinations are the gold standard for the investigation of the gastrointestinal tract. In fact, the production

of medical images with annotations is costly, especially for endoscopic images, as endoscopy requires expensive equipment resources and trained operators, as well as considerations of patient privacy and, finally, multiple physicians with extensive expertise in image annotation.

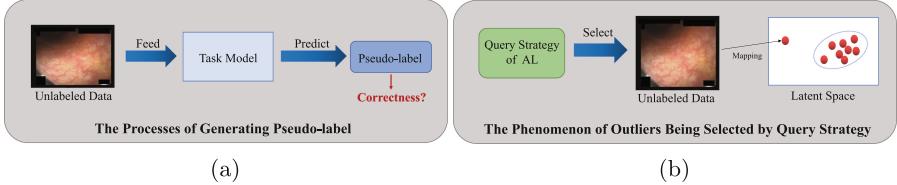


Fig. 1. Visualization of motivation. (a) The processes of generating pseudo-label. The practice of solely relying on prediction results as pseudo-labels for unlabeled data is not sufficiently accurate. (b) The phenomenon of outliers being selected by query strategy. This phenomenon is caused by the nature of outliers and ultimately affects the performance of the model.

To address the high cost of medical image annotation, researchers are exploring Semi-supervised Learning (SSL) and Active Learning (AL). SSL leverages unlabeled data alongside limited labeled data, often employing pseudo-label training to enhance model learning. AL, on the other hand, uses query strategies to select the most informative unlabeled data for expert labeling, which are then integrated into the training set. AL strategies include uncertainty-based, which prioritizes data with high uncertainty, and perturbation-based, which assesses data value by consistency of predictions post-perturbation. Combining SSL and AL into a semi-supervised active learning framework allows for efficient model training and data selection, fostering a synergistic improvement across training rounds.

Compared with normal natural images, endoscopic images of the gastrointestinal tract have highly similar structures and rich details. It is necessary to pay attention to fine details for classification. CNNs, which primarily capture local features, have limitations in this regard. In contrast, Vision Transformers (ViT) [4, 7, 10, 16, 22–24, 39] excel at capturing global context and detailed features by modeling relationships between image patches. Recognizing ViT's advantages in medical imaging [6], we will utilize it as our model backbone in upcoming experiments.

With the strong support of ViT, semi-supervised active learning plays a certain role in endoscopic image classification tasks with limited labeled data. However, there are still issues that need to be addressed in the practical application of the method. As shown in Fig. 1a, in conventional pseudo-label training, unlabeled data relies solely on the predictions of the task model as their pseudo-labels. However, due to the scarcity of endoscopic images, this approach fails to guarantee the correctness of pseudo-labels for unlabeled endoscopic images. As shown in Fig. 1b, due to external factors and inter-individual variations, certain

endoscopic images exhibit significant differences in features compared to others, thus rendering them as outliers. Owing to the characteristics of outliers, active learning methods tend to favor these data. This will lead to a deterioration in the performance of the endoscopic image classification model, which may affect the judgment of the physician judgment of the gastrointestinal tract condition of the patient. Therefore, we need to further consider additional information to optimize the generation of pseudo-labels. Second, the probability of outliers being selected by the query strategy needs to be reduced.

In order to address the aforementioned issues, we propose a novel semi-supervised active learning framework called ROSAL. ROSAL is composed of Representational Correlation-based Pseudo-label Training (RCPT) and Outlier-based Hybrid Querying (OHQ). The purpose of RCPT is to train the model and optimize the generation of pseudo-labels by our designed pseudo-label generator. When generating pseudo-labels by pseudo-label generator, a higher demand is placed on the discrimination between representations. Therefore, we design a pseudo-label contrast loss to further improve the discrimination between representations. The pseudo-label contrast loss constrains the model to aggregate representations of unlabelled data that are consistent with predictions in latent space and to alienate representations of unlabelled data that are inconsistent with predictions in latent space. This improves the discrimination between representations. In terms of representation, our proposed pseudo-label generator further considers the similarity magnitude between the representation of the unlabeled data and the representative representation of each category. It combines the prediction results to generate more accurate pseudo-labels for the unlabeled data. The purpose of OHQ is to select unlabeled data with high annotation values and reduce the probability of the outliers being selected. It includes a distance factor and a hybrid query strategy based on uncertainty and instability. The distance factor is able to indirectly identify outliers, while the hybrid query strategy determines the value of the annotation of the unlabeled data by calculating its uncertainty and instability.

To summarize, the contributions of this paper are as follows:

- (1) We propose a novel semi-supervised active learning framework called ROSAL, which achieves better performance compared to other active learning baselines.
- (2) Our proposed RCPT introduces a contrastive loss for pseudo-labels that groups consistent unlabeled data in the latent space and separates inconsistent ones, enhancing the accuracy of the generated pseudo-labels.
- (3) Our proposed OHQ defines a distance factor to reduce the probability of outliers being selected by the query strategy and integrates a hybrid query strategy that assesses both uncertainty and instability to identify data that most benefits model performance.
- (4) Our proposed ROSAL achieves 71.46% accuracy on LIMUC using only 40% of the data, proving the effectiveness of ROSAL on medical tasks. ROSAL achieves 90.79% accuracy on CIFAR100 using only 20% of the data, proving the generalizability of ROSAL.

2 Related Work

2.1 Semi-supervised Learning

Semi-supervised learning aims to train models using unlabeled data and limited labeled data. Traditional Semi-supervised learning method include pseudo-label training, which uses model predictions to label unlabeled data, and consistency regularization, which enforces stable predictions across data perturbations. Many approaches combine these methods, often applying consistency checks post-augmentation [29, 35]. SoftMatch [3] is a recent development that improves this by applying Gaussian weight functions to pseudo labels, balancing their quantity and quality without a fixed threshold.

However, these methods ignore the correlation between unlabeled and labeled data, resulting in pseudo-labels that are not yet accurate enough. Therefore, our method evaluates the magnitude of similarity between different representations to improve the accuracy of pseudo-labels.

2.2 Active Learning

With a tight annotation budget, active learning chooses key samples to enhance model performance. Common strategies include uncertainty-based [8, 11, 18], diversity-based [2, 27], change-based [1, 14, 32, 33], and adversarial [28, 34] approaches. Hybrid strategies, like those by Zhang [36] and Hu [12], combine these methods, assessing data value through uncertainty, instability, and gradient changes.

However, current active learning methods often overlook outliers, which can be either beneficial or detrimental to model performance. Karamchetti et al. [17] noted that selecting too many outliers can impair model effectiveness. Thus, our strategy incorporates outlier impact into the assessment of data's annotation value.

3 Method

3.1 Overview of ROSAL

We assume that there is a data pool X . The initial labeled data set $X^l = \{(x_i^l, y_i^l)\}_{i=0}^{N_l}$ is selected from X , which is annotated by experts. The rest of X as the unlabeled data set $X^u = \{x_i^u\}_{i=0}^{N_u}$, i.e. $X = X^l \cup X^u$, where N_l and N_u denote the number of the labeled and unlabeled data set, respectively. The learner is a deep neural network $f = f_b \odot f_c$ parameterized by $\theta = \{\theta_b, \theta_c\}$. f_b is a backbone that can feed the input data into the backbone to obtain the representation of the data in the latent space, i.e., $r_i = f_b(x_i, \theta_b), x_i \in X$. f_c is a classification that can map the input representation to the corresponding logits, which are passed through the Softmax layer to become the predicted probability distribution, i.e., $p_i = \text{Softmax}(f_c(r_i, \theta_c))$. The above can be expressed as $p_i = \text{Softmax}(f(x_i, \theta)), x_i \in X$. Index_{\max} is a function that can transform

the probability distribution p_i into the category corresponding to the element with the largest probability. In general, all data were fed into the model with a weak augmentation strategy, including random level reversal and random cropping. Our goal is to train the proposed ROSAL using both X^l and X^u . For our proposed ROSAL, the overall process is demonstrated in Fig. 2.

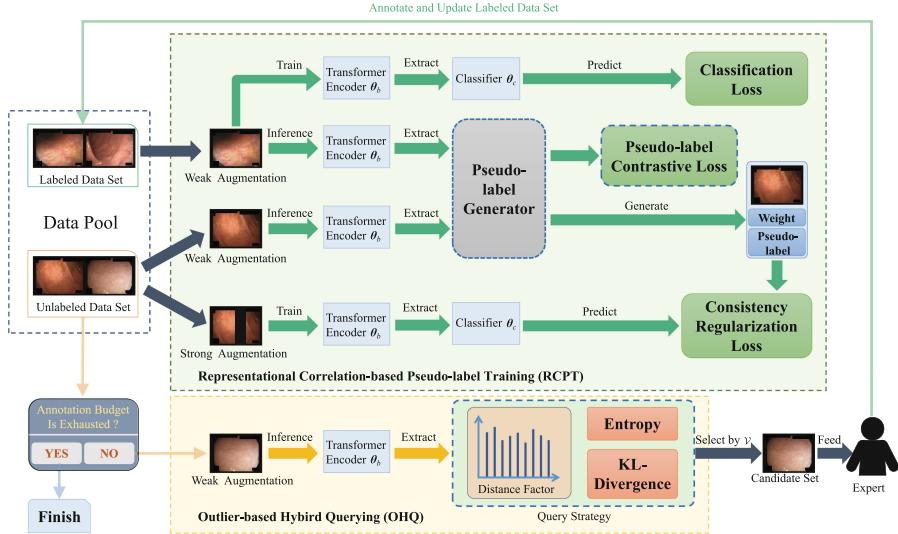


Fig. 2. Overview of ROSAL. We use ViT [7] as the backbone of our model. Weak Augmentation and Strong Augmentation denote the data is processed by weak and strong augmentation strategies, respectively.

3.2 Representational Correlation-Based Pseudo-label Training

Pseudo-label Contrastive Loss. In the latent space, representations hold valuable information for model training. Yet, endoscope images have similar structures, making it difficult to differentiate their representations. To address this, we've adapted the triplet loss to create a pseudo-label contrastive loss. This new loss encourages the model to group similar unlabeled representations and separate dissimilar ones in the latent space, enhancing inter-category discrimination. Pseudo-label contrastive loss \mathcal{L}_{pc} is defined as:

$$\mathcal{L}_{pos} = \frac{1}{N_{pos}} \sum_{i,j=0}^{N_{pos}} \log \left(e^{D(r_i^u, r_j^u)} \right) \quad (1)$$

$$\mathcal{L}_{neg} = -\frac{1}{N_{neg}} \sum_{i,k=0}^{N_{neg}} \log \left(1 + \frac{1}{e^{-D(r_i^u, r_k^u)}} \right) \quad (2)$$

$$\mathcal{L}_{pc} = \frac{1}{2} \times (\mathcal{L}_{pos} + \mathcal{L}_{neg}) \quad (3)$$

where $r_i^u = f_b(x_i^u, \theta_b)$, $x_i^u \in X^u$. r_j^u is a representation in which the category predicted after the classifier θ_c is consistent with r_i^u , and r_z^u is a representation in which the category predicted after the classifier θ_c is inconsistent with r_i^u . D is a non-negative distance function. N_{pos} denotes the number of representations of the same prediction category and N_{neg} denotes the number of representations of different prediction categories.

The final loss \mathcal{L}_{ROSAL} can be formulated as:

$$\mathcal{L}_{ROSAL} = \mathcal{L}_{cls} + \mathcal{L}_{con} + \mathcal{L}_{pc} \quad (4)$$

where \mathcal{L}_{cls} and \mathcal{L}_{con} are classification loss and consistency regularization loss computed by cross-entropy loss.

Pseudo-label Generator. Inspired by SoftMatch [3], we define the weight function of the pseudo-label of each unlabeled data x_i^u as:

$$\lambda(x_i^u) = \begin{cases} \exp\left(-\frac{(\max(p_i^u) - \mu)^2}{2\sigma^2}\right), & \text{if } \max(p_i^u) < \mu, \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

where $p_i^u = \text{Softmax}(f(x_i^u, \theta))$, $x_i^u \in X^u$, $\max(p_i^u)$ denotes the maximum value in the probability distribution p_i^u i.e., the predictive confidence of x_i^u . μ and σ are the two parameters of the Gaussian distribution. μ can be regarded as an alternative dynamic threshold. The weight of the pseudo-label is maximized when the prediction confidence is not less than μ .

Then, we fit μ and θ from the training:

$$\hat{\mu} = \frac{1}{N_u} \sum_{i=1}^{N_u} \max(p_i^u) \quad (6)$$

$$\hat{\sigma}^2 = \frac{1}{N_u} \sum_{i=1}^{N_u} (\max(p_i^u) - \hat{\mu})^2 \quad (7)$$

We use EMA to update parameters:

$$\hat{\mu}_t = m\hat{\mu}_{t-1} + (1-m)\hat{\mu}_{t-1} \quad (8)$$

$$\hat{\sigma}_t^2 = m\hat{\sigma}_{t-1}^2 + (1-m)\frac{N_u}{N_u-1}\hat{\sigma}_{t-1}^2 \quad (9)$$

where m is a hyperparameter of EMA. t denotes t -th iteration.

Finally, the weight function λ is updated as:

$$\lambda(x_i^u) = \begin{cases} \exp\left(-\frac{(\max(p_i^u) - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2}\right), & \text{if } \max(p_i^u) < \hat{\mu}_t, \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

The generation of pseudo-labels cannot only rely on the prediction results of the model but also needs to consider the correlation between unlabeled and labeled data. We utilize the similarity between the representations of the data to measure their relevance. Specifically, we calculate the distance between the representations of all labeled data in each category, and select the representation with the smallest average distance from other representations as the representative representation of each category:

$$r_k^l = \min(D(r_{k,i}^l, r_{k,j}^l)), i, j = 1, 2, \dots, N_k \quad (11)$$

where $k \in C$, C is category set. D is a non-negative distance function. N_k denotes the number of labeled data belonging to the K -th category.

We estimate the similarity between the representation of the unlabeled data and the representative representation of each category by normalizing the similarity to 0–1 to obtain the similarity S between the representation of the unlabeled data and the representation of each category:

$$S(x_i^u) = \frac{1}{2N_C} \sum_k^C \frac{r_i^u \cdot r_k^l}{\|r_i^u\| \cdot \|r_k^l\|} + 0.5 \quad (12)$$

where N_C denotes the number of categories.

We perform an element-wise multiplication operation on $S(x_i^u)$ and p_i^u . Then the dot-multiplication result is calculated by the Softmax function to finally generate the pseudo-label \hat{y} :

$$\hat{y}(x_i^u) = \text{Index}_{\max}(\text{Softmax}(S(r_i^u) \odot p_i^u)) \quad (13)$$

At this point, our pseudo-label generator has generated the pseudo-label and the weight of the pseudo-label for each unlabeled data x_i^u , which will be used for model training.

3.3 Outlier-Based Hybrid Querying

Outliers generally deviate from the majority of the data in the set, and Karamcheti et al. [17] have illustrated that a large number of collective outliers are selected by active learning query strategies. Because these outliers contain information different from the regular data, they are considered effective acquisition targets by the query strategy. However, after a large number of outliers are added to the training set, the model is unable to learn meaningful information, which affects the performance of the model. Therefore, our proposed OHQ considers the impact of this aspect when selecting candidate-labeled data.²

Distance Factor. We want to reduce the probability that outliers within a category are selected by the query strategy. To do this, we feed the unlabeled data x_i^u into the backbone to extract the corresponding representation r_i^u . We estimate the similarity δ of those representations that agree with the predictions

obtained after r_i^u has been run through the model, and then normalize the similarity from 0 to 1:

$$\delta(x_i^u) = \frac{1}{2N_u} \sum_{j=0}^{N_u} \frac{r_i^u \cdot r_j^u}{\|r_i^u\| \cdot \|r_j^u\|} + 0.5 \quad (14)$$

where r_j^u is a representation of the category of predictions obtained after the model is consistent with r_i^u .

The larger δ is, the greater the distance of that data from the same category of data in latent space. For data with larger δ , the higher the probability we think that these data are outliers, the more we need to reduce the value of these data to reduce the probability that the query strategy will select them. In this regard, we define a distance factor to OHQ, which takes similarity as distance and can adapt to adjust its annotation value according to the size of the distance between x_i^u and other unlabeled data in the latent space. The distance factor ω is defined as:

$$\omega(x_i^u) = 1 - \delta(x_i^u) \quad (15)$$

Hybrid Query Strategy Based on Uncertainty and Instability. In the view of query strategy, the uncertainty and instability of data are strongly related. The high instability of the data means that it only takes a small disturbance to add to the data and the result of the data predicted by the model will change dramatically. High uncertainty means that the data contains more information than conventional data. To obtain unlabeled data with high uncertainty, we send the unlabeled data x_i^u to the model and calculate the information entropy of this data as its uncertainty α :

$$\alpha(x_i^u) = - \sum_{i=0}^{N_u} p_i^u \log(p_i^u) \quad (16)$$

To obtain unlabeled data with high instability, we send the unlabeled data x_i^u to the backbone f_b to extract the corresponding representation r_i^u and then feed the representation into the classifier f_c of the model to get its prediction p_i^u . We conduct virtual adversarial training on r_i^u and p_i^u and get the virtual adversarial perturbation r_i^{adv} . r_i^{adv} and r_i^u are combined to obtain virtual perturbation representation \hat{r}_i^u , namely $r_i^{adv} + r_i^u = \hat{r}_i^u$. We feed \hat{r}_i^u into the classifier of model and get its perturbation \hat{p}_i^u . Finally, KL divergence between p_i^u and \hat{p}_i^u is calculated as the instability β of the data. Therefore, we need to calculate the perturbation r_i^{adv} before we can calculate the KL divergence. r_i^{adv} is formulated as:

$$\hat{p}_i^u = \text{Softmax}(f_c(r_i^u + \Delta r, \theta_c)) \quad (17)$$

$$r_i^{adv} = \arg \max_{\Delta r, \|\Delta r\| \leq \epsilon} \text{Div}(p_i^u, \hat{p}_i^u) \quad (18)$$

where ϵ denotes the maximum step length of the disturbance. Div is a non-negative function that measures the similarity between two distributions and we use KL divergence in practice. Δr is the unit vector of random sampling.

In fact, the calculation of r_i^{adv} is very tricky because when $r = 0$, the gradient of Div relative to r_i^{adv} is always 0. Therefore, we use [25] to solve it. Firstly, we approximate r_i^{adv} by repeatedly applying the following formula to update several times:

$$r_i^{adv} \leftarrow \epsilon \overline{\nabla_{\Delta r} Div(p_i^u, \hat{p}_i^u)} \quad (19)$$

One iteration of backpropagation in the neural network is used to compute $\nabla_{\Delta r} Div$, which is then used to solve for r_i^{adv} . We can get the instability β of the unlabeled data x_i^u :

$$\hat{p}_i^u = Softmax(f_c(r_i^u + r_i^{adv}, \theta_c)) \quad (20)$$

$$\beta(x_i^u) = Div(p_i^u, \hat{p}_i^u) \quad (21)$$

Finally, we obtain the annotation value \mathcal{V} of the unlabeled data x_i^u :

$$\mathcal{V}(x_i^u) = \omega(x_i^u) \cdot \alpha(x_i^u) \cdot \beta(x_i^u) \quad (22)$$

To ensure the balance between the categories as much as possible, we select the unlabeled data with the highest annotation value according to the order of the categories to be added to the candidate-labeled data set, which will be labeled by the experts and then updated to the labeled data set for the next round of training.

4 Experiments and Discussion

4.1 Datasets

We evaluated ROSAL on LIMUC [26] and CIFAR100 [19]. **LIMUC** is an ulcerative colitis dataset with 4 categories (“Mayo-0”, “Mayo-1”, “Mayo-2”, and “Mayo-3”) based on the Mayo endoscopic score. Each category was balanced to contain 1200 images, totaling 4800 images. We used 4000 images for training and 800 for testing, with images sized at 352 pixels \times 288 pixels. **CIFAR100** is a natural dataset with 100 categories and 60,000 images (600 per category). We used 50,000 images for training and 10,000 for testing, with images sized at 32 pixels \times 32 pixels.

4.2 Experiments Setting

To ensure fairness, initial labeled data is randomly selected as X^l from X , with the remaining X^u as unlabeled. For LIMUC, the initial labeled data is 8% of X , with subsequent annotation budgets of 8% per round (up to 40% max). For CIFAR100, initial labeled data is 4% of X , with 4% annotation budgets per round (up to 20% max). ViT [7] is used as the backbone with AdamW optimizer. Image sizes are 224 for LIMUC and 32 for CIFAR100, with patch sizes of 16 and 2 respectively, and embedding dimension of 384 for both. Results averaged over three experiments with different seeds on an NVIDIA RTX 3090 GPU.

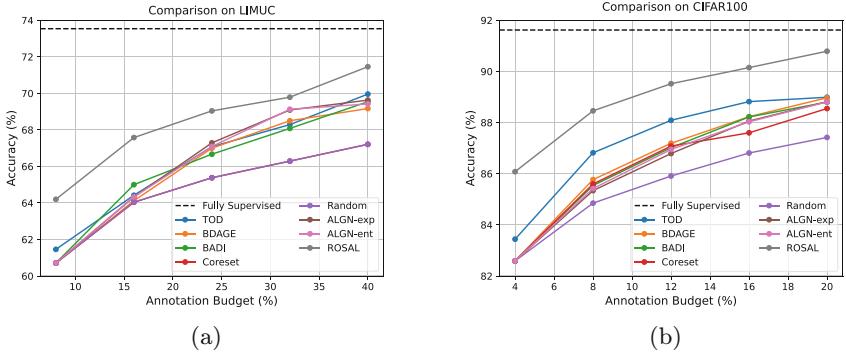


Fig. 3. Results of ROSAL compared to other baselines.

4.3 Baselines and Comparison Experiments

Baselines. To assess the validity of ROSAL, we compare our proposed ROSAL with the following active learning baselines.

- **TOD** [14]: an active learning baseline based on a variance paradigm that evaluates value by the difference in the recurrent output of model.
- **BADGE** [1]: a hybrid baseline that takes the centroids obtained by clustering the gradient embeddings and querying them as candidate-labeled data.
- **BALD** [8]: a Bayesian network-based baseline that uses Bayesian inconsistency to calculate uncertainty.
- **Coreset** [2]: a coresset-based baseline that selects one batch of the most representative data among unlabeled data as a candidate for labeling.
- **ALGN** [32]: a gradient-paradigm-based baseline that treats the gradient-paradigm of the data as annotated values. The baseline has two query strategies, Expected Gradient Paradigm (**ALGN-exp**) and Entropy Gradient Paradigm (**ALGN-ent**).
- **Random**: a baseline with random sampling.

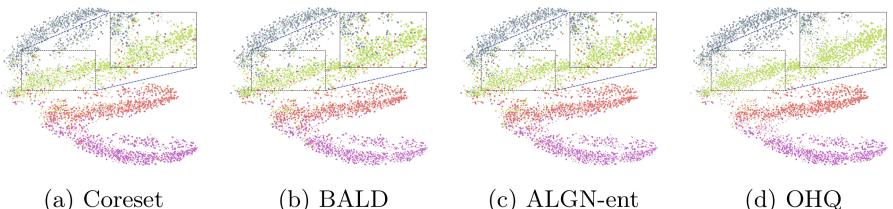
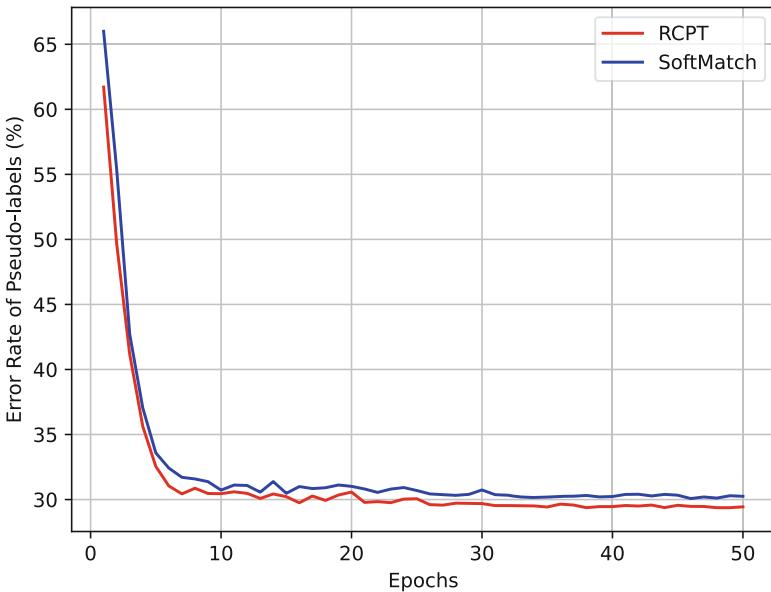


Fig. 4. Visualization of the query process using t-SNE.

Table 1. Performance of different active learning methods.

Methods	Accuracy (%) \pm STD (%)									
	LIMUC					CIFAR100				
	8% Budget	16% Budget	24% Budget	32% Budget	40% Budget	4% Budget	8% Budget	12% Budget	16% Budget	20% Budget
TOD	61.46 \pm 0.83	64.42 \pm 0.50	67.08 \pm 0.59	68.29 \pm 0.52	69.96 \pm 0.38	83.44 \pm 0.49	86.82 \pm 0.51	88.09 \pm 0.30	88.82 \pm 0.18	88.99 \pm 0.16
BALD	60.71 \pm 1.23	65.00 \pm 0.54	66.67 \pm 1.25	68.08 \pm 1.06	69.54 \pm 0.76	82.59 \pm 0.57	85.54 \pm 0.22	87.02 \pm 0.20	88.22 \pm 0.15	88.80 \pm 0.05
BADGE	60.71 \pm 1.23	64.12 \pm 0.13	67.00 \pm 0.58	68.50 \pm 0.25	69.17 \pm 0.32	82.59 \pm 0.57	85.77 \pm 0.28	87.19 \pm 0.34	88.23 \pm 0.23	88.97 \pm 0.16
Coreset	60.71 \pm 1.23	64.91 \pm 0.71	66.04 \pm 0.85	67.79 \pm 0.14	68.63 \pm 0.75	82.59 \pm 0.57	85.60 \pm 0.12	87.08 \pm 0.16	87.60 \pm 0.06	88.55 \pm 0.25
Random	60.71 \pm 1.23	64.04 \pm 0.96	65.38 \pm 0.61	66.29 \pm 0.69	67.21 \pm 0.80	82.59 \pm 0.57	84.85 \pm 0.31	85.91 \pm 0.02	86.81 \pm 0.10	87.42 \pm 0.13
ALGN-exp	60.71 \pm 1.23	64.33 \pm 1.09	67.29 \pm 1.16	69.09 \pm 0.44	69.63 \pm 0.87	82.59 \pm 0.57	85.34 \pm 0.37	86.79 \pm 0.30	88.06 \pm 0.19	88.82 \pm 0.15
ALGN-ent	60.71 \pm 1.23	64.33 \pm 1.09	67.09 \pm 1.35	69.13 \pm 0.50	69.42 \pm 0.71	82.59 \pm 0.57	85.42 \pm 0.31	86.95 \pm 0.02	88.03 \pm 0.10	88.81 \pm 0.13
ROSAL	64.20\pm2.40	67.58\pm0.14	69.04\pm0.52	69.79\pm1.26	71.46\pm0.51	86.08\pm0.30	88.46\pm0.37	89.52\pm0.02	90.15\pm0.06	90.79\pm0.03

Comparison Experiments. Figure 3 demonstrates the results of the comparison experiments between ROSAL and the above baseline on LIMUC and CIFAR100. Specific results are presented in Table 1. In Fig. 3a, we can observe that our proposed ROSAL outperforms all other methods on LIMUC. ROSAL achieves 71.46% accuracy using only 40% of the data. Moreover, ROSAL always achieves the performance advantage under each annotation budget. These results demonstrate the effectiveness and reliability of ROSAL on the task of endoscopic image classification in annotation-limited scenarios. In Fig. 3b, we can observe that ROSAL also outperforms other methods on CIFAR100, and ROSAL achieves 90.79% accuracy using only 20% of the data. Similarly, ROSAL still achieves a performance advantage under each annotation budget. These results demonstrate the generalizability of ROSAL.

**Fig. 5.** Comparison of error rates per epoch.

4.4 Ablation Analysis

We conduct an ablation study to illustrate the effectiveness of each component in ROSAL. We perform three repetitions of the experiment using different random seeds, and the results are averaged over the three experiments. The results of the experiments are presented in Table 2. Row 1 demonstrates the performance when applying SoftMatch [3] only. Row 2 demonstrates the performance when combining SoftMatch, Pseudo-label Contrastive Loss (PCL), and Pseudo-label Generator (PG), i.e., RCPT. Row 3 demonstrates the performance when applying randomly sampled. Row 4 demonstrates the performance when applying a hybrid query strategy based on Uncertainty And Instability (UAI) only. And row 5 demonstrates the performance when combining UAI and Distance Factor (DF), i.e., OHQ. Finally, row 6 demonstrates the performance of our proposed ROSAL.

The results of row 1 and row 2 illustrate the effectiveness of the RCPT, improving the accuracy by 0.29% and 0.30% on LIMUC and CIFAR100, respectively. The results of row 4 and row 5 illustrate the effectiveness of OHQ, improving the accuracy by 0.92% and 0.08% on LIMUC and CIFAR100, respectively. The above results demonstrate the effectiveness of each component in ROSAL.

Table 2. Ablation Study of different modules in ROSAL.

	SSL Selection			AL Selection		Accuracy (%) \pm STD (%)		
	SoftMatch	PCL	PG	Random	UAI	DF	LIMUC	CIFAR100
SoftMatch [3]	✓						69.96 \pm 0.63	89.65 \pm 0.04
RCPT	✓	✓	✓				70.46 \pm 0.19	89.94 \pm 0.07
Random				✓			67.21 \pm 0.80	87.46 \pm 0.23
UAI					✓		69.29 \pm 2.10	88.45 \pm 0.17
OHQ					✓	✓	70.21 \pm 1.45	88.54 \pm 0.39
ROSAL (Ours)	✓	✓	✓		✓	✓	71.46 \pm 0.51	90.79 \pm 0.03

4.5 Visualization Analysis

Visualization of Representational Correlation-Based Pseudo-label Training. To validate RCPT’s performance, we compared its pseudo-label error rates on LIMUC with SoftMatch. By randomly selecting 40% of the data as labeled and the rest as unlabeled, Fig. 5 shows the error rates per epoch. The results indicate that RCPT enhances pseudo-label accuracy.

Visualization of Outlier-Based Hybrid Querying. To validate OHQ’s effectiveness, we used t-SNE to visualize how different query strategies select data from the LIMUC dataset. The visualizations (Fig. 4) compare OHQ (Fig. 4d)

with Coreset (diversity-based, Fig. 4a), BALD (uncertainty-based, Fig. 4b), and ALGN-ent (entropy-based, Fig. 4c) methods. Dark, large dots indicate selected data, while light, small dots represent unlabeled data. The red dots within the black dashed box are likely outliers. OHQ selected fewer of these, suggesting it can effectively reduce outlier selection and maintain model performance.

5 Conclusion

In this paper, we propose a novel semi-supervised active learning framework for endoscopy image classification called ROSAL to address the problem of the accuracy of pseudo-labels and the over-selection of outliers by the query strategy. Our proposed RCPT designs a pseudo-label contrastive loss to improve the discrimination between representations and generates more accurate pseudo-labels by the pseudo-label generator. Our proposed OHQ defines a distance factor to reduce the probability of outliers being selected by the query strategy and selects the data that is more effective in improving the performance of the current model. Experimental results demonstrate that ROSAL outperforms other active learning baselines on LIMUC, approaching the performance of fully supervised learning using only 40% of the labeling budget. This demonstrates the effectiveness of ROSAL in endoscopic image classification tasks. ROSAL outperforms other active learning baselines on CIFAR100, approaching the performance of fully supervised learning using only 20% of the labeling budget, demonstrating the generalizability of ROSAL.

Acknowledgments. This work was supported in part by the Key Areas Research and Development Program of Guangzhou Grant 2023B01J0029, Science and technology research in key areas in Foshan under Grant 2020001006832, the Key-Area Research and Development Program of Guangdong Province under Grant 2018B010109007 and 2019B010153002, the Science and technology projects of Guangzhou under Grant 202007040006, the Guangdong Provincial Key Laboratory of Cyber-Physical System under Grant 2020B1212060069, the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012534, the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011729, and the National Statistical Science Research Project of China under Grant 2022LY096

References

1. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020)
2. Borsos, Z., Mutny, M., Krause, A.: Coresets via bilevel optimization for continual learning and streaming. NeurIPS **33**, 14879–14890 (2020)
3. Chen, H., et al.: SoftMatch: addressing the quantity-quality tradeoff in semi-supervised learning. In: ICLR (2023)
4. Chen, X., Cun, X., Pun, C.M., Wang, S.: ShaDocNet: learning spatial-aware tokens in transformer for document shadow removal. In: ICASSP, pp. 1–5 (2023)

5. Chen, X., Lei, B., Pun, C.M., Wang, S.: Brain diffuser: an end-to-end brain image to brain network pipeline. In: PRCV, pp. 16–26 (2023)
6. Chen, X., Pun, C.M., Wang, S.: MedPrompt: cross-modal prompting for multi-task medical image translation. arXiv preprint [arXiv:2310.02663](https://arxiv.org/abs/2310.02663) (2023)
7. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: ICLR (2021)
8. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: ICML, pp. 1183–1192 (2017)
9. Gong, C., et al.: Generative ai for brain image computing and brain network computing: a review. *Front. Neurosci.* **17**, 1203104 (2023)
10. Guo, X., Chen, X., Luo, S., Wang, S., Pun, C.M.: Dual-hybrid attention network for specular highlight removal. In: ACM MM (2024)
11. Houlsby, N., Huszar, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. arXiv preprint [arXiv:1112.5745](https://arxiv.org/abs/1112.5745) (2011)
12. Hu, W., et al.: Learning from incorrectness: active learning with negative pre-training and curriculum querying for histological tissue classification. *IEEE Trans. Med. Imag.* (2023)
13. Huang, G., Chen, X., Shen, Y., Wang, S.: Mr image super-resolution using wavelet diffusion for predicting Alzheimer’s disease. In: BI, pp. 146–157 (2023)
14. Huang, S., Wang, T., Xiong, H., Huan, J., Dou, D.: Semi-supervised active learning with temporal output discrepancy. In: ICCV, pp. 3447–3456 (2021)
15. Jiang, H., Chen, X., Jin, C., Wang, S.: Structural brain network generation via brain denoising diffusion probabilistic model. In: International Conference on AI in Healthcare, pp. 264–277 (2024)
16. Jiang, Y., Chen, X., Pun, C.M., Wang, S., Feng, W.: MFDNet: multi-frequency Deflare network for efficient nighttime flare removal. *Vis. Comput.*, 1–14 (2024)
17. Karamcheti, S., Krishna, R., Fei-Fei, L., Manning, C.D.: Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In: IJCNLP, pp. 7265–7281 (2021)
18. Kirsch, A., Van Amersfoort, J., Gal, Y.: BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning. *NeurIPS* **32** (2019)
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
20. Li, Z., et al.: Encoding enhanced complex CNN for accurate and highly accelerated MRI. *IEEE Trans. Med. Imag.* (2024)
21. Li, Z., et al.: Complementation-reinforced network for integrated reconstruction and segmentation of pulmonary gas MRI with high acceleration. *Med. Phys.* **51**(1), 378–393 (2024)
22. Li, Z., Chen, X., Guo, S., Wang, S., Pun, C.M.: WavEnhancer: unifying wavelet and transformer for image enhancement. *J. Comput. Sci. Technol.* **39**(2), 336–345 (2024)
23. Li, Z., Chen, X., Pun, C.M., Cun, X.: High-resolution document shadow removal via a large-scale real-world dataset and a frequency-aware shadow erasing net. In: ICCV, pp. 12449–12458 (2023)
24. Luo, S., Chen, X., Chen, W., Li, Z., Wang, S., Pun, C.M.: DeVigNet: high-resolution vignetting removal via a dual aggregated fusion transformer with adaptive channel expansion. In: AAAI (2024)
25. Miyato, T., Maeda, S., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI* **41**(8), 1979–1993 (2019)

26. Polat, G., Kani, H.T., Ergenc, I., Ozen Alahdab, Y., Temizel, A., Atug, O.: Improving the computer-aided estimation of ulcerative colitis severity according to mayo endoscopic score by using regression-based deep learning. *Inflamm. Bowel Dis.* **29**(9), 1431–1439 (2023)
27. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: *ICLR* (2018)
28. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: *ICCV*, pp. 5972–5981 (2019)
29. Sohn, K., et al.: FixMatch: simplifying semi-supervised learning with consistency and confidence. *NeurIPS* **33**, 596–608 (2020)
30. Sung, H., et al.: Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J. Clin.* **71**(3), 209–249 (2021)
31. Tang, H., et al.: RM-UNet: UNet-like mamba with rotational SSM module for medical image segmentation. *Sig. Image Video Process.* **18**(11), 8427–8443 (2024)
32. Wang, T., et al.: Boosting active learning via improving test performance. In: *AAAI*, vol. 36, pp. 8566–8574 (2022)
33. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: *CVPR*, pp. 93–102 (2019)
34. Zhang, B., Li, L., Yang, S., Wang, S., Zha, Z.J., Huang, Q.: State-relabeling adversarial active learning. In: *CVPR*, pp. 8756–8765 (2020)
35. Zhang, B., et al.: FlexMatch: boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS* **34**, 18408–18419 (2021)
36. Zhang, W., et al.: BoostMIS: boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. In: *CVPR*, pp. 20666–20676 (2022)
37. Zheng, F., et al.: SMAFormer: synergistic multi-attention transformer for medical image segmentation. *arXiv preprint arXiv:2409.0034* (2024)
38. Zhou, T., Chen, X., Shen, Y., Nieuwoudt, M., Pun, C.M., Wang, S.: Generative AI enables EEG data augmentation for Alzheimer’s disease detection via diffusion model. In: *ISPCE-ASIA*, pp. 1–6 (2023)
39. Zhou, Z., et al.: DocDeshadower: frequency-aware transformer for document shadow removal. *arXiv preprint arXiv:2307.15318* (2024)



Adaptive Population-Based Incremental Learning for Feature Selection in Leukemia Gene Expression Data

Eranga N. Fernando¹(✉) and Jeremiah D. Deng²

¹ University of Moratuwa, Colombo, Sri Lanka
nuwani.fernando@gmail.com

² School of Computing, University of Otago, Dunedin, New Zealand
jeremiah.deng@otago.ac.nz

Abstract. We present an adaptive PBIL (Population-Based Incremental Learning) algorithm for feature selection in leukemia gene expression data. The proposed adaptive strategy aimed to improve learning rates within the PBIL framework while reducing feature count. Among the tested methods, APBIL-GP (Adaptive PBIL with Gradient-Proportional learning rate adjustment) demonstrated superior performance by achieving the highest Separability Index (SI) value (0.9244) and effectively reducing the feature count down to 3.9%. The selected features led to improved classification performance, particularly with the Support Vector Machine (SVM) and Random Forest (RF) classifiers. t-SNE visualizations validated the efficacy of APBIL-GP-selected features, showing clear boundaries between leukemia subtypes. Further analysis using Jaccard indices and extensive cross-validation confirmed that APBIL-GP explored unique features, reduced redundancy, and achieved the highest mean accuracy (0.9078), significantly outperforming Boruta, RF, and Chi-squared (Chi2). These findings suggest that APBIL-GP is a robust method for feature selection in gene expression data.

Keywords: Feature Selection · Adaptive PBIL · Gradient-Proportional Learning Rate · Gene Expression Data

1 Introduction

Gene expression profiling has become a crucial tool in the classification and prognosis of leukemia, enabling the identification of specific subtypes [22]. The high-dimensional nature and limited number of samples in gene expression data, as exemplified by the Leukemia datasets, pose significant challenges due to the curse of dimensionality [29]. Effective feature selection is critical for reducing dimensionality while preserving essential biological information [12]. Traditional methods often struggle with high-dimensional datasets, necessitating more sophisticated approaches [27].

In the realm of biomedical research, deciphering complex gene expression patterns is pivotal for understanding and distinguishing diseases such as leukemia. With the advent of high-throughput technologies, vast amounts of genomic data are now available, presenting both opportunities and challenges in extracting meaningful insights [30]. Feature selection plays a crucial role in this process, aiming to identify a subset of genes that are most discriminative across different disease subtypes [19].

This research aims to enhance feature selection using a modified Population Based Incremental Learning (PBIL) algorithm [2], evaluating its performance on a gene expression dataset by improving the Separability Index (SI) [18]. PBIL is an estimation of distribution (EDA) algorithm that iteratively updates a probability vector that represents the likelihood of each gene's contribution to separability among leukemia subtypes.

Unlike traditional approaches, the modified PBIL dynamically adjusts learning rates, thereby enhancing its adaptability to the inherent complexity of datasets [8,9]. While optimizing feature subsets for accurate differentiation of leukemia subtypes, we preferred SI over classification accuracy [21] as the fitness function for PBIL due to its ability to mitigate computational complexity and its parameter-free nature. In essence, our research contributes to the evolving landscape of computational biology by leveraging adaptive learning techniques to unravel the intricate genetic underpinnings of leukemia, thereby paving the way for more targeted and personalized treatment approaches.

2 Literature Review

2.1 Population-Based Incremental Learning (PBIL) and Its Variants

Population-Based Incremental Learning (PBIL) was introduced by Baluja [1,2] as a combination of Genetic Algorithms and competitive learning principles from Artificial Neural Networks. Subsequent research introduced adaptive learning rates to enhance PBIL's performance. Folly [8] implemented APBIL by linearly increasing the learning rate over iterations, demonstrating improved stability and convergence in optimizing power system controller parameters. Bolaños et al. [4] explored four different learning rules (linear, sigmoidal, exponential, and bell-shaped) for APBIL, finding that a sigmoidal adaptation based on entropy yielded the best results.

PBIL has been adapted for various optimization problems, including feature selection in high-dimensional datasets like gene expression data [20,21]. Its adaptive nature allows for efficient exploration of feature spaces and convergence to optimal solutions, making it suitable for complex datasets.

2.2 Feature Selection

Feature selection is critical in gene expression data analysis due to high dimensionality and the need to identify relevant genes for classification and diagnosis

[28]. Begum et al. [3] provided a comprehensive overview of feature selection methods, categorizing them into filter, wrapper, embedded, and hybrid methods. They emphasized the importance of method selection based on the specific application.

Mahendran et al. [16] found that supervised gene selection methods outperform unsupervised and semi-supervised methods in microarray datasets, highlighting the value of labeled data. Machine learning approaches like SVM and RF have been employed for feature selection in gene expression data. SVM uses a margin-based approach for feature selection [25, 31], while RF ranks features based on their importance scores derived from impurity reduction [25].

Recent advancements include deep learning and ensemble methods. Mori et al. [17] proposed a deep learning-based approach for feature selection in gene expression data, demonstrating potential in identifying prognostic factors for pancreatic cancer. Ensemble methods combine multiple learning algorithms to enhance robustness and performance in feature selection [26].

High-dimensional feature selection is also critical in other fields such as text classification and image analysis. Methods like Chi-square testing, Mutual Information, and Information Gain are commonly used for this purpose [23]. In image analysis, high-dimensional feature selection is crucial for tasks such as image classification, segmentation, and object detection [5].

Gene expression data presents unique challenges, such as the high-dimensional nature relative to sample size, biological and technical noise, and the need for biologically relevant and interpretable features [14, 32]. Effective feature selection methods must address these issues to avoid overfitting and ensure accurate classification.

In conclusion, while significant advancements have been made in feature selection for high-dimensional gene expression data, the field remains open for further research. The literature emphasizes the need for specialized techniques tailored to the complexities of gene expression datasets, and no definitive optimal solution has been established, underscoring the need for continued exploration and refinement.

3 Methodology

3.1 Dataset

The publicly available dataset, Leukemia GSE28497 [7], as shown in Table 1 below, consists of 281 samples, 7 different classes, and 22,283 gene expression features, and was utilized for our experiment. This dataset offers a robust foundation for investigating gene expression patterns across various leukemia subtypes. Prior to analysis, we conducted thorough preprocessing to validate the dataset's integrity, ensuring no duplicates or missing values were present. Since our dataset does not resemble a Gaussian distribution according to the p-value of the Shapiro test, we split the dataset into training and testing sets and then applied the Robust Scaler, which is based on the median and interquartile range, for feature scaling.

Table 1. Dataset Details

Leukemia Subtypes	Class ID	No. of samples (percent)
B-CELL_ALL	1	74 (26.33%)
B-CELL_ALL_ETV6-RUNX1	2	53 (18.86%)
B-CELL_ALL_HYPERDIP	3	51 (18.15%)
B-CELL_ALL_T-ALL	4	46 (16.37%)
B-CELL_TCF3-PBX1	5	22 (7.83%)
B-CELL_ALL_HYPO	6	18 (6.41%)
B-CELL_ALL_MLL	7	17 (6.05%)
Total Count	-	281 (100%)

3.2 PBIL Algorithm

The PBIL algorithm serves as our primary framework for optimizing feature selection in the context of leukemia subtype classification. Suppose there are d features in total to be selected. PBIL operates through iterative updates of a d -dimension probability vector \mathbf{p} , which represents the likelihood of selecting each feature. A pool of feature representation vectors, $x_i, i = 1, \dots, n$ are employed to represent stochastic selections of features, which are further evaluated. In the feature selection context, the representation vectors are binary:

$$x_i^j = \begin{cases} 1, & \text{if } \zeta < p^j; \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $j = 1, \dots, d$, and ζ is a random number between 0 and 1. Initially, \mathbf{p} is initialized uniformly with all entries set to 0.5. Throughout each iteration, it is adjusted based on the evaluation of generated population vectors using a chosen fitness function f . If a feature j is present in the best vector, denoted by $\mathbf{b} = \arg_{\mathbf{x}_i} f(\mathbf{x}_i)$, its probability of selection should be increased, and the probability vector is updated by

$$p^j = p^j(1 - \alpha) + \alpha b^j, \quad (2)$$

where α is a (positive) learning rate. Conversely, if a feature j is present in the worst vector but not in the best vector, its probability of selection in \mathbf{p} decreases:

$$p^j = p^j(1 - \eta) + \eta b^j \quad (3)$$

where η is a negative learning rate.

Once \mathbf{p} is updated, it goes through optional mutation. This process involves randomly reducing or increasing the probability of each entry in \mathbf{p} , thereby diversifying the genetic composition of the population [21]:

$$p^j = p^j(1 - s) + \gamma s \quad (4)$$

where s is the mutation shift, and γ is random binary (+1 or -1) that controls the direction of mutation.

3.3 Fitness Function

We followed [21] in using a SI as the fitness function for APBIL, motivated by its computational efficiency, suitability for high-dimensional datasets, and parameter-free nature. Unlike traditional classification accuracy metrics that require training classifiers, the SI evaluates feature subsets based on pairwise distances without the need for extensive model training. This computational advantage significantly reduces processing time and resource requirements, making it feasible to explore a wide range of feature combinations efficiently.

SI quantifies the degree to which samples in a dataset are well-separated based on their class labels. SI is computed as the average proportion of samples for which the nearest neighbor shares the same class label as the sample itself. This metric is crucial for evaluating the effectiveness of a feature selection method or dimension reduction technique in enhancing class discriminability. The process involves three key steps:

Step 1: Identify the nearest neighbor (NN) for each data point \mathbf{x}_i ;

$$\Lambda(\mathbf{x}_i) = \arg_{\mathbf{x}_j} \min_{j \neq i} D(\mathbf{x}_i, \mathbf{x}_j). \quad (5)$$

Step 2: Evaluate Class Consistency (CS) based on whether nearest neighbours agree on their class labels:

$$S(\mathbf{x}_i) = \begin{cases} 1, & \text{if } y(i) = y(\Lambda(\mathbf{x}_i)); \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Step 3: Obtain the Separability Index (SI) as the average of class consistency scores:

$$SI = \frac{1}{n} \sum_{i=1}^n S(\mathbf{x}_i). \quad (7)$$

Given the dataset's high dimensionality with 22,283 gene expression features, computational complexity becomes a critical consideration. The SI allows us to focus directly on the discriminatory power of feature subsets in distinguishing between leukemia subtypes, bypassing the computational overhead associated with classifier training and validation. This approach not only streamlines the optimization process but also ensures that our feature selection strategy remains scalable and practical for large-scale genomic datasets.

3.4 Adaptive Learning Rate Adjustment

All three adaptive methods that we used in this study to adjust the learning rates can be put as

$$\alpha = \alpha_L + (\alpha_H - \alpha_L)\phi, \quad (8)$$

where α_L is the minimum possible value for the learning rate α , α_H is the maximum, and ϕ is the adaptive factor obtained from the following three different methods.

Fixed Learning Rate (Fixed-LR). The Fixed-LR method is used to implement standard PBIL by setting the adaptive factor $\phi = 0$ in Eq. (8). This keeps the learning rate at its minimum value until the termination condition is met.

Linear Learning Rate Adjustments (APBIL-Folly). As proposed by Folly in [8], this method adjusts the learning rate linearly over iterations. The adaptive factor is simply set by

$$\phi_{GP} = \frac{i}{T}, \quad (9)$$

which is a ratio of the iteration count i to the total number of iterations allowed. This results in a learning rate that increases linearly from α_L to α_H according to Eq. (8).

Gradient-Proportional Learning Rate Adjustments (APBIL-GP). The APBIL-GP method adjusts the learning rate based on the magnitude of the gradients of the updated population \mathbf{p} . As in [8], the LR starts from the minimum, but we penalize the increment using a normalized gradient value:

$$\phi_{GP} = 1 - \frac{\|\mathbf{b} - \mathbf{p}\|}{\|\mathbf{p}\| + \epsilon}, \quad (10)$$

The gradient is calculated as the difference between the best population vector \mathbf{b} and the probability vector \mathbf{p} . A large gradient means the exploitation is aggressively at work, so a smaller increment, hence a smaller learning rate, would be desirable to prevent premature convergence.

To further reduce the chance for PBIL to get stuck at local minima, we also introduce a momentum factor μ to give a smoother update of the learning rate:

$$\alpha' = \mu\alpha' + (1 - \mu)\alpha, \quad (11)$$

where α is the learning rate suggested by the GP rule, and α' is the final learning rate updated by α using exponential smoothing.

3.5 Experiment Setup

For our experiments, we utilized the Leukemia GSE28497 dataset, which consists of 281 samples and 22,283 gene expression features. We employed a population size of 1000 individuals within APBIL, where each individual represents a probability vector \mathbf{p} governing the selection of gene expression features. This large population size allowed for comprehensive exploration of the feature space, enhancing the likelihood of identifying robust genetic markers associated with leukemia subtypes.

The optimization process spanned $T = 600$ generations, during which PBIL iteratively updated \mathbf{p} based on the fitness of generated sample vectors. During the process both learning rates and negative learning rates are dynamically adjusted based on different strategies between 0.1 and 0.3 to balance exploration

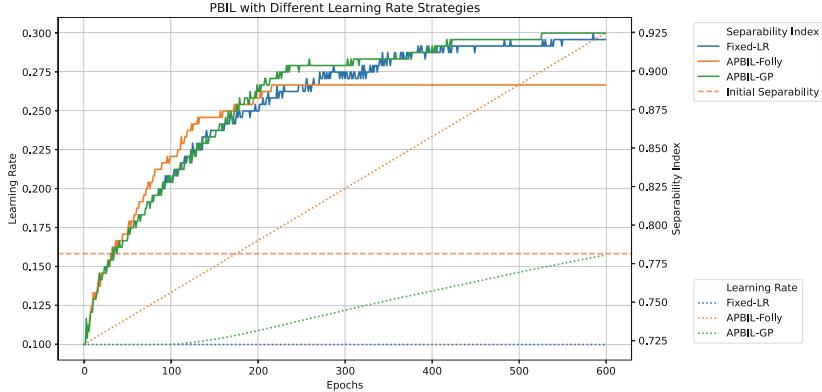


Fig. 1. PBIL learning outcome with different adaptive learning rate strategies. PBIL-GP avoids premature convergence and achieves the highest SI.

and exploitation of the feature space. We apply various schemes to update α , and set the negative learning rate equal to the (positive) learning rate: $\eta = \alpha$. We set $\mu = 0.996$ to maintain a strong momentum for avoiding local minima. And upon updating \mathbf{p} , it undergoes mutation to introduce variability and promote diversity within the probability vectors \mathbf{p} . A mutation probability of 0.02 and a mutation shift of 0.05 were applied.

The evaluation prioritized the SI, which aids in selecting features that enhance inter-class differences while reducing intra-class variance. Although [24], which used the same dataset as ours, suggests that feature selection before or after data splitting does not lead to data leakage, we decided to split the dataset into training and testing sets with a ratio of 85:15. This decision was made to eliminate any potential risk of data leakage. In every generation, if the probability of an instance is greater than 0.5, it is considered a selected feature. The final selected feature count is determined using the final probability vector based on the same criteria. The performance is then assessed using different classification methods. To further evaluate the selected features, we conducted repeated stratified K-fold cross-validation with 10 folds and 50 repeats using four different feature sets selected by different methods.

4 Results and Discussion

4.1 Model Selection and Performance Evaluation

We conducted a comprehensive analysis of three strategies for adapting learning rates within the PBIL algorithm. The results of three different feature selection methods shown in Fig. 1 demonstrate how each strategy converges to its highest SI value within 600 epochs while selecting feature subsets of the Leukemia gene expression dataset. It also illustrates how the learning rates vary across total

Table 2. Performance evaluation of the proposed PBIL variants.

Model	Method	Accuracy	Precision	Recall	F1 score	SFC (percent)
SVM	Chi-squared [24]	0.86	0.78	0.82	0.80	400 (1.8%)
	Fixed-LR	0.86	0.84	0.85	0.85	538 (2.4%)
	APBIL-Folly	0.88	0.86	0.87	0.86	2115 (9.5%)
	APBIL-GP	0.88	0.93	0.87	0.86	880 (3.9%)
RF	Chi-squared [24]	0.88	0.81	0.83	0.82	400 (1.8%)
	Fixed-LR	0.86	0.79	0.82	0.80	538 (2.4%)
	APBIL-Folly	0.84	0.75	0.84	0.79	2115 (9.5%)
	APBIL-GP	0.88	0.80	0.88	0.84	880 (3.9%)

iterations. The initial SI value achieved with the original dataset is considered the baseline for performance comparison.

Within the termination condition (600 epochs), APBIL-GP achieved its highest SI value and reached convergence by the 527th epoch. The highest SI found by the APBIL-Folly method occurred at the 217th epoch. This indicates that APBIL-GP has the ability to conduct a more diverse search than APBIL-Folly does. Since APBIL-Folly converges more quickly, it has a higher chance of reaching a local optimal solution. PBIL with a fixed learning rate achieved a comprehensive SI value but did not converge to an optimal solution. By the termination condition, all three methods excluded original features to a significant extent, as shown by the “SFC” (selected feature counts) column in Table 2.

Considering all these factors, we regard APBIL-GP as the best model among the other two methods because it achieves the highest SI value and its slower convergence allows more exploration for the optimal solutions. Additionally, it reduces the feature count down to nearly 4% of the original feature count.

Classification Performance. We used 85% of the data for training and the remaining part for validation, following the approach in [24]. We trained two classifiers: SVM and RF, which were also employed in the aforementioned study that utilized the same dataset. As highlighted in Table 2, SVM yields better results with both APBIL-GP and APBIL-Folly, while RF performs better with both APBIL-GP and Chi2 method used in [24].

T-SNE Visualization. The feature selection schemes were further visually validated by t-SNE [15]. Figure 2 below illustrates the efficacy of features in distinguishing between different leukemia subtypes before (2-a) and after selection using three methods (2-b, 2-c, and 2-d). Among these methods (Fixed-LR, APBIL-Folly, and APBIL-GP), the features selected by APBIL-GP show significantly clearer and more distinct boundaries, indicating better separation between leukemia subtypes and contributing to improved classification.

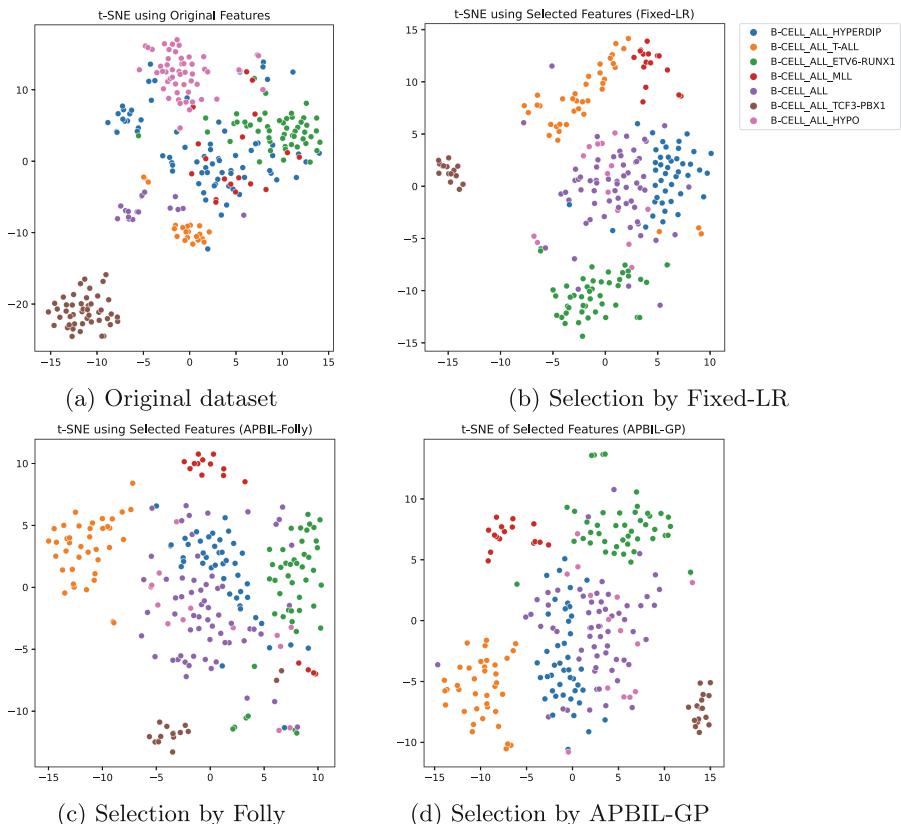


Fig. 2. t-SNE plots of selected features using Different LR adaptation methods with PBIL: (a) Original Dataset, (b) Fixed-LR, (c) Folly's Method and (d) APBIL-GP. Project points are color-keyed by their corresponding class labels.

In essence, the algorithm's key strength lies in its ability to dynamically adapt \mathbf{P} based on the performance of the population vectors, aiming to maximize the SI. By iteratively refining \mathbf{P} , APBIL-GP efficiently explores the feature space, focusing on subsets that enhance the distinguishability of leukemia subtypes. This approach facilitates more accurate separability and provides deeper insights into the genetic markers associated with different disease profiles.

4.2 Detailed Evaluation of Selected Features

The APBIL-GP method selected 880 features when it met its termination condition. To gain further insights into these selected features, we extracted an equal number of features from the original dataset using Boruta [13], Random Forest, and Chi-squared (chi2) test. Boruta was originally designed as a wrapper around an RF classifier. However, in this study, we adapted it to use the SI for eval-

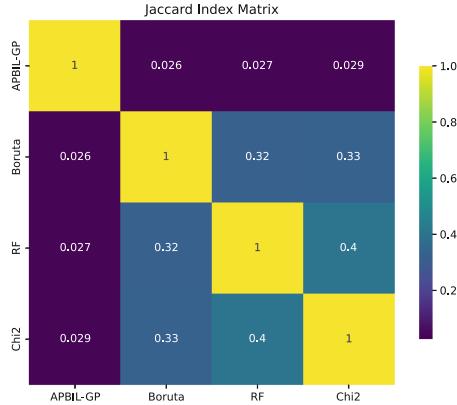


Fig. 3. Matrix representation of Jaccard Index

uating each feature's importance, iteratively removing features based on their individual SI values.

To select the top 880 features based on their importance scores obtained from the RF classifier, we conducted a grid search with 10-fold cross-validation and determined the best hyperparameters: `n_estimators`: 2500, `max_depth`: 25, `min_samples_split`: 10.

Jaccard Indices. We used the Jaccard index to measure the similarity and diversity of four feature sets derived from different methods (APBIL-GP, Boruta, RF and Chi2), each containing the same number of features. Figure 3 indicates the mutual similarities among APBIL-GP and the other three feature sets.

This visualization highlights that APBIL-GP exhibits moderate Jaccard indices compared to Boruta and Chi2 methods. However, it displays the least similarity to the top features from RF. Comparing its results with these methods can provide more comprehensive insights, as each method captures distinct aspects of the data. We can see that APBIL-GP's search approach deviates the most from the RF top-feature set, potentially exploring alternative features that contribute less redundancy while enhancing classification.

Classification Performance of Selected Features. In our study, we employed 50 repeated 10-fold cross-validation. This approach involves performing 10-fold cross-validation 50 times independently, which helps to mitigate the issue of data leakage, especially important given our dataset's relatively small size of 281 samples. By repeatedly splitting the data into different training and validation sets, we ensure that each data point is used in both roles multiple times, reducing the risk of overfitting and providing a more robust estimate of model performance. The choice of 50 repeats was made to balance computational efficiency with the need for reliable and stable performance metrics,

ensuring that our results are not overly dependent on a particular random split of the data.

Table 3 compares the mean accuracy of APBIL-GP, Boruta [11], RF [6], and Chi2 [10] methods, along with their statistical significance as determined by paired t-tests. APBIL-GP achieves a mean accuracy of 0.9078, surpassing Boruta, RF, and Chi2. The statistical analysis reveals significant differences between APBIL-GP and the other methods, indicating that APBIL-GP outperforms them in terms of mean accuracy. This comparison underscores APBIL-GP's competitive performance, particularly in comparison to Boruta, RF, and Chi2.

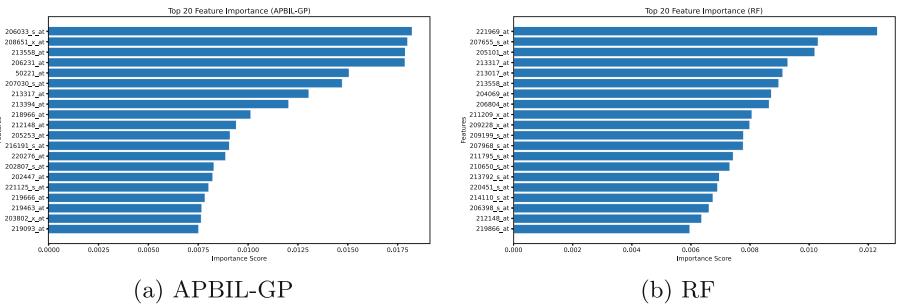


Fig. 4. Feature importance plots of selected features using: (a) APBIL-GP, (b) RF

Table 3. Classification Performance of Selected 880 Features. Asterisks after p-values indicate the significance level: $p < 0.05$ (*), $p < 0.01$ (**).

Method	Accuracy Mean \pm Std	t-statistics (vs APBIL-GP)	p-value
			-
APBIL-GP	0.9078 \pm 0.0499	-	-
Boruta	0.8906 \pm 0.0570	5.0700	0.0000**
RF	0.8994 \pm 0.0539	2.5646	0.0105*
Chi2	0.8841 \pm 0.0519	7.3481	0.0000**

Comparison of Feature Importance Values. As shown in Fig. 4, the top features' importance scores are similar between APBIL-GP and RF, indicating that both methods consistently identify crucial features. However, their most important features may differ, suggesting that although the original dataset contains many similar features, APBIL-GP manages to select less redundant ones that significantly contribute to its performance. Moreover, the Jaccard index reveals

a minimal 3% overlap between feature sets derived from APBIL-GP and RF, highlighting APBIL-GP's ability to prioritize less-important yet potentially less redundant features, thereby enhancing its overall selection performance.

5 Conclusion

In this study, we introduced and evaluated an adaptive PBIL algorithm for feature selection in leukemia gene expression data. Our adaptive strategies aimed to improve the learning rates within the PBIL framework, enhancing the feature selection process. APBIL-GP demonstrated superior performance by achieving its highest SI value (0.9244) by the 527th epoch, indicating its capability for a more extensive and diverse search compared to APBIL-folly and PBIL with a fixed learning rate. APBIL-GP efficiently reduced the original feature count by nearly 4%, leading to improved classification performance, particularly with the SVM and RF classifiers. The t-SNE visualizations further validated the efficacy of the features selected by APBIL-GP, showing significantly clearer boundaries between leukemia subtypes.

Further analysis of the same number of selected features with three other feature sets derived from Boruta, RF, and Chi2, using Jaccard indices and extensive cross-validation, showed that APBIL-GP explored unique features, reduced redundancy, and enhanced classification, achieving the highest mean accuracy (0.9078). It significantly outperformed Boruta, RF, and Chi2.

In conclusion, APBIL-GP is a robust method for feature selection in leukemia gene expression data, offering high accuracy, efficient feature reduction, and unique feature exploration capabilities. This adaptive approach to PBIL has proven its potential to enhance the feature selection process, making it a valuable tool for genomics and bioinformatics applications.

Building on the promising results of the Gradient Proportional method, future research will include a more in-depth exploration of the biological significance and potential clinical applications of the selected features. We also acknowledge the need for a comprehensive comparison with other state-of-the-art feature selection methods, which will be addressed in future studies. Additionally, further investigation into the impact of different parameter settings on the performance of adaptive PBIL methods will be conducted. Expanding the evaluation to include a broader range of gene expression datasets and other high-dimensional biomedical datasets will be essential to generalize these findings and validate the robustness of adaptive PBIL methods across diverse biological contexts.

References

1. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning (1994)
2. Baluja, S.: Genetic algorithms and explicit search statistics. In: Advances in Neural Information Processing Systems, pp. 319–325 (1997)
3. Begum, S., Khan, E.S., Chakraborty, D.: A survey of feature selection methods for the analysis of microarrays data in cancer. *Int. J. Intell. Syst. Appl. Eng.* **11**(10s), 472–482 (2023)
4. Bolanos, F., Aedo, J.E., Rivera, F.: Comparison of learning rules for adaptive population-based incremental learning algorithms. In: Proceedings of the 2012 International Conference on Artificial Intelligence, ICAI 2012, vol. 1, pp. 244–251 (2012)
5. Bolón-Canedo, V., Remeseyro, B.: Feature selection in image analysis: a survey. *Artif. Intell. Rev.* **53**(4), 2905–2931 (2020)
6. Díaz-Uriarte, R., Alvarez de Andrés, S.: Gene selection and classification of microarray data using random forest. *BMC Bioinf.* **7**(1), 3 (2006)
7. Feltes, B.C., Chandelier, E.B., Grisci, B.I., Dorn, M.: CuMiDa: an extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. *J. Comput. Biol.* **26**(4), 376–386 (2019)
8. Folly, K.A.: Population-based incremental with adaptive learning rate strategy. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012. LNCS, vol. 7331, pp. 11–20. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30976-2_2
9. Folly, K.A., Venayagamoorthy, G.K.: Effects of learning rate on the performance of the population based incremental learning algorithm. In: Proceedings of the International Joint Conference on Neural Networks, pp. 861–868 (2009)
10. Jin, X., Xu, A., Bie, R., Guo, P.: Machine learning techniques and Chi-square feature selection for cancer classification using SAGE gene expression profiles. In: Li, J., Yang, Q., Tan, A.-H. (eds.) BioDM 2006. LNCS, vol. 3916, pp. 106–115. Springer, Heidelberg (2006). https://doi.org/10.1007/11691730_11
11. Kavitha, K.R., Sajith, S., Variar, N.H.: An efficient Boruta-based feature selection and classification of gene expression data. In: 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), pp. 1–6 (2022)
12. Kouser, K., Lavanya, P.G., Lalitha, R., Acharya, K.K.: Effective feature selection for classification of promoter sequences. *PLoS ONE* **11**(12) (2016)
13. Kursa, M.B., Jankowski, A., Rudnicki, W.R.: Boruta-a system for feature selection. *Fund. Inform.* **101**(4), 271–285 (2010)
14. Lu, Y., Han, J.: Cancer classification using gene expression data. *Inf. Syst.* **28**(4), 243–268 (2003)
15. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008)
16. Mahendran, N., Durai Raj Vincent, P.M., Srinivasan, K., Chang, C.Y.: Machine learning based computational gene selection models: a survey, performance evaluation, open issues, and future research directions. *Front. Genet.* **11** (2020)
17. Mori, Y., et al.: Deep learning-based gene selection in comprehensive gene analysis in pancreatic cancer. *Sci. Rep.* **11**(1), 1–9 (2021)
18. Mthembu, L., Marwala, T.: A note on the separability index (2008)
19. Omara, H., Lazaar, M., Tabii, Y.: Effect of feature selection on gene expression datasets classification accuracy. *Int. J. Electr. Comput. Eng.* **8**(5), 3194–3203 (2018)

20. Osakidetza, M.M., et al.: Feature subset selection by population-based incremental learning. A case study in the survival of cirrhotic patients treated with TIPS (1970)
21. Perez, M., Rubiny, D.M., Marwala, T., Scottz, L.E., Stevenszx, W.: A population-based incremental learning approach to microarray gene expression feature selection. In: 2010 IEEE 26th Convention of Electrical and Electronics Engineers in Israel (IEEEEI), pp. 10–14 (2010)
22. Richards, L.: Gene-expression profiling in leukemia-a valuable diagnostic tool. *Nat. Rev. Clinical Oncol.* **7**(8), 422 (2010)
23. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: International Conference on Information and Knowledge Management, Proceedings, pp. 659–661 (2002)
24. Rupapara, V., Rustam, F., Aljedaani, W., Shahzad, H.F., Lee, E., Ashraf, I.: Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model. *Sci. Rep.* **12**(1), 1000 (2022)
25. Saengsiri, P., Wichian, S.N., Meesad, P.: Efficient feature selection model for gene expression data. *Appl. Mech. Mater.* **110–116**, 1948–1952 (2012)
26. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008. LNCS (LNAI), vol. 5212, pp. 313–325. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87481-2_21
27. Thudumu, S., Branch, P., Jin, J., Singh, J.J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. *J. Big Data* **7**(1), 1–30 (2020)
28. Vanjimalar, S., Ramyachitra, D., Manikandan, P.: A review on feature selection techniques for gene expression data. In: 2018 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2018. Institute of Electrical and Electronics Engineers Inc. (2018)
29. Wang, Y., Miller, D.J., Clarke, R.: Approaches to working in high-dimensional data spaces: gene expression microarrays. *Br. J. Cancer* **98**(6), 1023–1028 (2008)
30. Xue, B., Khoroshhevskyi, O., Gomez, R.A., Sheffield, N.C.: Opportunities and challenges in sharing and reusing genomic interval data. *Front. Genet.* **14**, 1155809 (2023)
31. Zhang, Y., Deng, Q., Liang, W., Zou, X.: An efficient feature selection strategy based on multiple support vector machine technology with gene expression data. *BioMed Res. Int.* **2018**, 7538204 (2018)
32. Zheng, C.H., Huang, D.S., Kong, X.Z., Zhao, X.M.: Gene expression data classification using consensus independent component analysis. *Genomics Proteomics Bioinf.* **6**(2), 74–82 (2008)

Author Index

A

- A. Jain, Hardik 165
Amir, Guy 284
Apperley, Mark 319
Atkins, Martin 319

B

- Baćić, Boris 72
Beheshti, Amin 165, 243
Budhraja, Sugam 48
Burgess, Tony 61

C

- Chen, Hua 258
Chen, Jianwu 350
Chen, Xuhang 350
Chen, Zhenxiang 32
Cheng, Haoran 227
Corsi, Davide 284

D

- Deng, Jeremiah D. 365
Doborjeh, Maryam 48, 61
Doborjeh, Zohreh 48, 61
Du, Tongchun 227

E

- Ebrahim Hossain, Muhammad 303

F

- Farinelli, Alessandro 284
Feng, Chengwei 72
Fernando, Eranga N. 365
French, Julie 61
Fujiwara, Kaisei 180

G

- Gandomi, Amir H. 116
Gong, Peiliang 335

- Grasso, Floriana 133
Guo, Shanqing 32

H

- Harel, David 284
Hasegawa, Tatsuhito 1
Heym, Nadja 61
Hirate, Takahiro 211
Hu, Jiayu 258
Huang, Guoheng 350
Huang, Haocai 149
Huang, Lin-Lin 87
Huang, Xiaocong 350
Hunter, Kirsty 61

K

- Kamruzzaman, Joarder 271
Kanneganti, Deepak 243
Kasabov, Nicola 61
Kasabov, Nikola 48
Katz, Guy 284
Krishna, Aneesh 165, 243
Kurz, Jason 319

L

- Lai, Edmund 48
Li, Hongwei 258
Li, Weihua 72
Li, Xuanchen 149
Li, Yinbao 17
Li, Yuan 87
Liu, Chang 17
Liu, Jianwei 271
Liu, Jie 227
Lu, Liangfu 271
Lukasik, Szymon 116

M

- Ma, Yiqun 196
Mahmud, Mufti 303

Meng, Lingru 149
 Mistry, Sajib 165, 243

N

Ni, Wenlong 258

O

Ozawa, Seiichi 180

P

Pang, Shaoning 271
 Park, Chanho 180
 Pasandideh, Mostafa 319
 Patel, Chirayu 165
 Pun, Chi-Man 350

R

R. Payne, Terry 133
 Rahman, M. Mostafizur 303
 Rajakaruna, Sumedha 243
 Raju, Sarvanakumar 116
 Ren, Yong 149
 Roxburgh, Andrew 133

S

Salgotra, Rohit 116
 Sarkar, Mustafa 61
 Scott, Aroha 61
 Sharma, Pankaj 116
 Shen, Pingzhang 32
 Siddique Ayon, Shahriar 303
 Singh, Balkaran 48
 Sumich, Alexander 61
 Sun, Qianru 335

U

Ullah Miah, Md Saef 303
 Umatiya, Riyasatali 165

W

Wang, Bo 227
 Wang, Jingjing 149
 Wang, Siyuan 196
 Wang, Wenrui 196
 Wang, Xiaorui 102

X

Xiong, Jiagui 258
 Xu, Jingzehua 149
 Xu, Jun 32

Y

Yamada, Akira 180
 Yamauchi, Koichiro 211
 Yan, Yuyao 196
 Yang, Xi 196
 Yerushalmi, Raz 284
 Yi, Yugen 102
 Yin, Fei 87
 Yuan, Xiaochen 350

Z

Zeng, Yongming 149
 Zhang, Daoqiang 335
 Zhang, Jintao 149
 Zhang, Liying 335
 Zhang, Yan-Ming 87
 Zhang, Yulei 17
 Zhao, Chuan 32
 Zhao, Shengnan 32
 Zhong, Guo 350
 Zhou, Bin 102
 Zhou, Wei 102
 Zhou, Xinyu 258
 Zhou, Yueying 335
 Zhu, Qi 335
 Zhu, Rui 17