

LNCS 15256

Tianqing Zhu
Jin Li
Aniello Castiglione (Eds.)

Algorithms and Architectures for Parallel Processing

24th International Conference, ICA3PP 2024
Macau, China, October 29–31, 2024
Proceedings, Part VI

6 Part VI



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.

Tianqing Zhu · Jin Li · Aniello Castiglione
Editors

Algorithms and Architectures for Parallel Processing

24th International Conference, ICA3PP 2024
Macau, China, October 29–31, 2024
Proceedings, Part VI

Editors

Tianqing Zhu  
City University of Macau
Macau, China

Jin Li  
Guangzhou University
Guangzhou, China

Aniello Castiglione  
University of Salerno
Fisciano, Italy

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-981-96-1550-6

ISBN 978-981-96-1551-3 (eBook)

<https://doi.org/10.1007/978-981-96-1551-3>

© 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

On behalf of the Conference Committee, we welcome you to the proceedings of the 2024 International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2024), which was held in Macau Special Administrative Region, China from October 29–31, 2024. ICA3PP 2024 was the 24th in this series of conferences (started in 1995) that are devoted to algorithms and architectures for parallel processing. ICA3PP is now recognized as the main regular international event that covers the many dimensions of parallel algorithms and architectures, encompassing fundamental theoretical approaches, practical experimental projects, and commercial components and systems. This conference provides a forum for academics and practitioners from countries around the world to exchange ideas for improving the efficiency, performance, reliability, security, and interoperability of computing systems and applications.

A successful conference would not be possible without the high-quality contributions made by the authors. This year, ICA3PP received a total of 265 submissions from authors in various countries and regions. Based on rigorous peer reviews by the Program Committee members and reviewers, 131 high-quality papers were accepted to be included in the conference proceedings and submitted for EI indexing. In addition to the contributed papers, distinguished scholars were invited to give keynote lectures, providing us with the recent developments in diversified areas in algorithms and architectures for parallel processing and applications.

Among the accepted papers, the papers with the highest weighted review mark in each round received the Best Paper Award. The Best Papers were *Updates Leakage Attack against Private Graph Split Learning*, Hao Yang, Zhuo Ma, Yang Liu, Xinjing Liu, Beiwei Yang, and Jianfeng Ma, *Data-Free Encoder Stealing Attack in Self-supervised Learning*, Chuan Zhang, Xuhao Ren, Haotian Liang, Qing Fan, Xiangyun Tang, Chunhai Li, Liehuang Zhu, and Yajie Wang, and *Federated Meta Continual Learning for Efficient and Autonomous Edge Inference* Bingze Li, Stella Ho, Youyang Qu, Chenhao Xu, Tom H. Luan, and Longxiang Gao. Best Student paper was *DIsFU: Protecting Innocent Clients in Federated Unlearning* Fanyu Kong, Xiangyun Tang, Yu Weng, Tao Zhang, Hongyang Du, Jiawen Kang, and Chi Liu.

We would like to take this opportunity to express our sincere gratitude to the 262 Program Committee members and reviewers for their dedicated and professional service. We highly appreciate the track chairs for their hard work in promoting this conference and organizing the reviews for the papers submitted to their tracks. We are so grateful to the publication chairs, Zuobin Ying and Sheng Wen, for their meticulous work in editing the conference proceedings. We must also say “thank you” to all the volunteers who helped us at various stages of this conference.

Moreover, we would like to extend our appreciation to the following chairs for their invaluable contributions:

Local Chairs

Wenjian Liu, Chris Chu, Max Kuok

Workshop Chairs

Jia Gu, Chaofeng Zhang, Fengshi Jin, Chi Liu

Publicity Chairs

Gengshen Wu, Lefeng Zhang

Registration Chairs

Kaiyao Jiang, Congcong Zhu

We were so honored to have many renowned scholars be part of this conference. Finally, we would like to thank all speakers, authors, and participants for their great contribution to and support for the success of ICA3PP 2024.

October 2024

Wanlei Zhou

Paulo Quaresma

Albert Zomaya

Willy Susilo

Tianqing Zhu

Jin Li

Aniello Castiglione

Organization

General Chairs

Wanlei Zhou

City University of Macau, China

Paulo Quaresma

University of Évora, Portugal

Albert Zomaya

University of Sydney, Australia

Willy Susilo

University of Wollongong, Australia

Program Chairs

Tianqing Zhu

City University of Macau, China

Jin Li

Guangzhou University, China

Aniello Castiglione

University of Salerno, Italy

Local Chairs

Wenjian Liu

City University of Macau, China

Chris Chu

City University of Macau, China

Max Kuok

City University of Macau, China

Publication Chairs

Zuobin Ying

City University of Macau, China

Sheng Wen

Swinburne University of Technology, Australia

Workshop Chairs

Jia Gu

City University of Macau, China

Chaofeng Zhang

Advanced Institute of Industrial Technology,
Japan

Fengshi Jin

City University of Macau, China

Chi Liu

City University of Macau, China

Publicity Chairs

Gengshen Wu
Lefeng Zhang

City University of Macau, China
City University of Macau, China

Registration Chairs

Kaiyao Jiang
Congcong Zhu

City University of Macau, China
City University of Macau, China

Program Committee

Aniello Castiglione

Department of Management and Innovation
Systems

Bangbang Ren

National University of Defense Technology

Bin Wu

Chinese Academy of Sciences

Bo Li

Swinbourn University of Technology

Bo Liu

University of Technology Sydney

Bowen Liu

Nanjing University

Chao Li

Beijing Jiaotong University

Chao Wang

University of Science and Technology of China

Chaokun Zhang

Tianjin University

Chen Zhang

City University of Hong Kong

Chentao Wu

Shanghai Jiao Tong University

Chi Liu

City University of Macau

Chris Chu

City University of Macau

Chuan Zhang

Beijing Institute of Technology

Chuang Hu

Wuhan University

Congcong Zhu

City University of Macau

Daniel Andresen

Kansas State University

Dayong Ye

University of Wollongong

Deze Zeng

China University of Geosciences

Dezun Dong

National University of Defense Technology

En Shao

Institute of Computing Technology

Faqian Guan

China University of Geosciences

Fei Lei

NUDT

Fuliang Li

Northeastern University

Fuyuan Song

Nanjing University of Information Science and
Technology

Geng Sun

Jilin University

Gongming Zhao	University of Science and Technology of China
Guang Wang	Florida State University
Guangwu Hu	Shenzhen Institute of Information Technology
Guo Chen	Hunan University
Guozhu Meng	Institute of Information Engineering
Hai Xue	University of Shanghai for Science and Technology
Haikun Liu	Huazhong University of Science and Technology
Hailong Yang	Beihang University
Haipeng Dai	Nanjing University
Haiping Huang	College of Computer
Haonan Lu	University at Buffalo
Haozhe Wang	University of Exeter
Heng Qi	Dalian University of Technology
Hongwei Zhang	Tianjin University of Technology
Hua Huang	University of California
Hui Sun	City University of Macau
Humayun Kabir	Microsoft
Ioanna Kantzavelou	University of West Attica
Jaya Prakash	Champati University of Victoria
Jiahui Li	Jilin University
Jin Li	Guangzhou University, China
Jinbin Hu	Hong Kong University of Science & Technology
Jing Gong	KTH Royal Institute of Technology
Jinguang Han	Southeast University
Jingwen Leng	Shanghai Jiao Tong University
Jinwen Xi	Beijing Zhongguancun Laboratory
Jordan Samhi	CISPA – Helmholtz Center for Information Security
Jun Shao	School of Computer and Information Engineering
Kaiping Xue	University of Science and Technology of China
Kejiang Ye	SIAT
Ladjel Bellatreche	LIAS/ENSMA
Lanju Kong	Shandong University
Laurent Lefevre	Inria
Lefeng Zhang	City University of Macau
Lei Wang	Soochow University
Letian Zhang	Middle Tennessee State University
Li Duan	Beijing Jiaotong University
Li Ma	ShangHai Jiao Tong University
Lijie Xu	Nanjing University of Posts and Telecommunications

Lin He	THU
Lingjun Pu	Nankai University
Liu Yuling	Institute of Information Engineering
Lizhao You	Xiamen University
Longxiang Gao	Qilu University of Technology
Lu Zhao	Nanjing University of Posts and Telecommunications
Mahbubur Rahman	City University of New York
Marc Frincu	West University of Timisoara
Massimo Cafaro	University of Salento
Massimo Torquati	University of Pisa
Max Kuok	City University of Macau
Meixuan Ren	Sichuan Normal University
Meng Li	Hefei University of Technology
Meng Li	Nanjing University
Meng Shen	Beijing Institute of Technology
Mengying Zhao	Shandong University
Mi Zhang	ICT
Minfeng Qi	City University of Macau
Minghao Zhao	East China Normal University
Minghui Xu	Shandong University
Mingwu Zhang	Hubei University of Technology
Minyu Feng	Southwest University
Mirazul Haque	Research Scientist
Peter Kropf	University of Neuchâtel
Philip Brown	University of Colorado Colorado Springs
Qianhong Wu	Beihang University
Qing Fan	BIT
Qiong Huang	South China Agricultural University
Radu Prodan	University of Klagenfurt
Ravishka Rathnasuriya	University of Texas at Dallas
Roman Wyrzykowski	Czestochowa University of Technology
Rongxing Lu	University of New Brunswick
Sa Wang	Institute of Computing Chinese Academy of Sciences
Shaojing Fu	National University of Defense Technology
Shen Dian	Southeast University
Sheng Ma	NUDT
Shenglin Zhang	Nankai University
Shuai Gao	Beijing Jiaotong University
Shuai Xu	Nanjing University of Aeronautics and Astronautics

Shuai Zhou	City University of Macau
Shuang Chen	Huawei Cloud
Shujie Han	Peking University
Shuxin Zhong	Rutgers University
Simin Chen	UTD
Songwen Pei	Dept. of Computer Science and Engineering
Su Yao	Tsinghua University
Susumu Matsumae	Saga University
Tao Wu	National University of Defense Technology
Tianqing Zhu	University of Technology
Tianyi Liu	HUAWEI
Tie Qiu	Tianjin University
Tingwen Liu	Institute of Information Engineering
Vladimir Voevodin	RCC MSU
Wei Bao	The University of Sydney
Wei Wang	Central South University
Weibei Fan	Nanjing University of Posts and Telecommunications
Weihua Zhang	Fudan University
Weitian Tong	Georgia Southern University
Weixing Ji	Beijing Normal University
Weizhi Meng	Lancaster University
Wenjuan Li	The Education University of Hong Kong
Wenxin Li	Tianjin University
Wenzheng Xu	Sichuan University
Xiang Zhang	Nanjing University of Information Science and Technology
Xiangyu Kong	Dalian University of Technology
Xiangyun Tang	BIT
Xiangyun Tang	Minzu University of China
Xiaojie Zhang	Hunan First Normal University
Xiaoli Gong	Nankai University
Xiaolu Li	Huazhong University of Science and Technology
Xiaoyang Xie	Rutgers University
Xiaoyi Tao	Dalian University of Technology
Xiaoyong Tang	School of Computer and Communication Engineering
Xiaoyu Wang	Soochow University
Xin He	Nanjing University of Posts and Telecommunications
Xin Xie	The Hong Kong Polytechnic University
Xuan Liu	Hunan University

Xueqin Liang	Xidian University
Yajie Wang	BIT
Yajie Wang	Beijing Institute of Technology
Yang Du	University of Science and Technology of China
Yanyan Wang	Nanjing University
Yi Ding	University of Texas at Dallas
Yi Zhao	Beijing Institute of Technology
Yifei Zhu	Shanghai Jiao Tong University
Yitao Hu	Tianjin University
Yizhi Zhou	Dalian University of Technology
Yongkun Li	University of Science and Technology of China
Yongqian Sun	Nankai University
Youyang Qu	Qilu University of Technology
Youyou Lu	Tsinghua University
Yu Zhang	Huazhong University of Science and Technology
Yuan Cao	Ocean University of China
Yuben Qu	Shanghai Jiao Tong University
Yuchao Zhang	Beijing University of Posts and Telecommunications
Yueming Wu	Nanyang Technological University
Yukun Yuan	University of Tennessee at Chattanooga
Yunxia Lin	Yangzhou University
Yutong Gao	Minzu University of China
Ze Zhang	University of Michigan/Cruise
Zhaoyan Shen	Shandong University
Zhen Ling	Southeast University
Zhengkai Wu	Citadel Securities
Zhengxiong Li	The University of Colorado Denver
Zhenlin An	The Hong Kong Polytechnic University
Zhiqiang Li	University of Nebraska
Zhiquan Liu	Jinan University
Zhou Qin	Amazon
Zhuoxuan Du	Ant Group
Zichen Xu	Nanchang University
Zongheng Wei	School of Computer Science
Zonghua Gu	UMU
Zuobin Ying	City University of Macau

Contents – Part VI

SteDM: Efficient Image Steganography with Diffusion Models	1
<i>Changguang Wang, Haoyi Shi, Qingru Li, Dongmei Zhao, and Fangwei Wang</i>	
BDFC:A New Flow Control Mechanism for Torus Networks	12
<i>Haofei Zhang, Youmeng Li, Yu Deng, and Qi Fang</i>	
A Hybrid Vectorized Merge Sort on ARM NEON	26
<i>Jincheng Zhou, Jin Zhang, Xiang Zhang, Tiaojie Xiao, Di Ma, and Chunye Gong</i>	
An Encoder-Based Framework for Privacy-Preserving Machine Learning	37
<i>Jiayun Wu, Wei Ren, Xianchao Zhang, and Xianghan Zheng</i>	
Load Balancing Optimizations for Distributed GMRES Algorithm	47
<i>Yuxiang Zhang, Shuaizhe Guo, Jianhua Gao, Weixing Ji, and Yizhuo Wang</i>	
ZKCross: An Efficient and Reliable Cross-Chain Authentication Scheme Based on Lightweight Attribute-Based Zero-Knowledge Proof	57
<i>Yuwei Xu, Hailang Cai, Jialuo Chen, Qiao Xiang, Jingdong Xu, and Guang Cheng</i>	
LSSM-SpMM: A Long-Row Splitting and Short-Row Merging Approach for Parallel SpMM on PEZY-SC3s	78
<i>Ligang Cao, Qinglin Wang, Shun Yang, Rui Xia, Weihao Guo, and Jie Liu</i>	
A Model Inference Attack Based on Random Sampling in DLaaS	98
<i>Feng Wu, Shouyue Sun, Jiaxun Yang, Liwen Wu, Lei Cui, Youyang Qu, and Shaowen Yao</i>	
Defense Against Textual Backdoors via Elastic Weighted Consolidation-Based Machine Unlearning	108
<i>Haojun Xuan, Yajie Wang, Huishu Wu, Tao Liu, Chuan Zhang, and Liehuang Zhu</i>	
Cross-Chain Transaction Auditing with Truth Discovery	122
<i>Huishu Wu, Xuhao Ren, Mengxuan Liu, Tao Liu, Yajie Wang, Chuan Zhang, and Liehuang Zhu</i>	

Exploring the Vulnerability of ECG-Based Authentication Systems Through A Dictionary Attack Approach	141
<i>Bonan Zhang, Chao Chen, Ickjai Lee, Kyungmi Lee, and Kok-Leong Ong</i>	
LBVP: Lightweight Blockchain-Based Vehicle Platooning Scheme for Secure and Efficient Platoon Management	159
<i>Wenjie Fan, Zhiqian Liu, Libo Wang, Ying He, Jingjing Guo, Xia Feng, and Jianfeng Ma</i>	
Low-Carbon Geographically Distributed Cloud-Edge Task Scheduling	179
<i>Yingjie Zhu, Ji Qi, Zehao Wang, Shengjie Wei, Yan Chen, Tuo Cao, Gangyi Luo, and Zhuzhong Qian</i>	
Integrating Blockchain, Smart Contracts, NFTs, and IPFS for Enhanced Transparency and Ethical Sourcing in Coffee and Cocoa Supply Chains	200
<i>V. H. Khanh, N. M. Triet, L. K. Bang, P. D. Trinh, N. N. Hung, N. H. Bang, P. T. Nghiem, and T. D. Khoa</i>	
The Role of Artificial Intelligence Technologies in Sustainable Urban Development: A Systematic Survey	217
<i>Maria Rosaria Sessa, Ornella Malandrino, and Antonio Cesarano</i>	
Enhancing Privacy in Machine Unlearning: Posterior Perturbation Against Membership Inference Attack	231
<i>Chen Chen, Hengzhu Liu, Huanhuan Chi, and Ping Xiong</i>	
PEbfs: Implement High-Performance Breadth-First Search on PEZY-SC3s	249
<i>Weihao Guo, Qinglin Wang, Xiaodong Liu, Muchun Peng, Shun Yang, Yaling Liang, Yongzhen Shi, Ligang Cao, and Jie Liu</i>	
Textual Data De-Privatization Scheme Based on Generative Adversarial Networks	268
<i>Yanning Du, Jinnan Xu, Yaling Zhang, Yichuan Wang, and Zhoukai Wang</i>	
Modeling and Simulation Verification of Operating Mode Switching of Train Control System Based on Train-to-Train Communication	282
<i>Qiang Li, Ian Liao, Sheng Wen, and Yang Xiang</i>	
Performance Evaluation of NLP Models for European Portuguese: Multi-GPU/Multi-node Configurations and Optimization Techniques	298
<i>Daniel Santos, Nuno Miquelina, Daniela Schmidt, Paulo Quaresma, and Vítor Beires Nogueira</i>	

FusionFrame: A Fusion Dataflow Scheduling Framework for DNN Accelerators via Analytical Modeling	315
<i>Liutao Zheng, Huiying Lan, Xiang Liu, Linshan Jiang, and Xuehai Zhou</i>	
Author Index	335



SteDM: Efficient Image Steganography with Diffusion Models

Changguang Wang^{1,2} , Haoyi Shi², Qingru Li^{1,2} , Dongmei Zhao^{1,2}, and Fangwei Wang^{1,2}

¹ Key Laboratory of Network and Information Security of Hebei Province,
Hebei Normal University, Shijiazhuang 050024, China
{fw_wang,qingruli}@hebtu.edu.cn

² College of Computer and Cyber Security, Hebei Normal University,
Shijiazhuang 050024, China

Abstract. Existing deep learning-based image steganography schemes mostly neglect the latent space of images. These schemes merely adopt simple concatenation of image feature vectors, resulting in a low utilization rate of features, low steganographic image quality, and poor image robustness. This paper introduces the latent diffusion model into an image steganography scheme, namely SteDM. The SteDM firstly uses an encoder to transform the cover image and the secret image into the latent space, then employing a cross-attention mechanism to fuse them during the inverse diffusion process. Then we use a decoder to obtain a steganographic image containing secret image features. During the extraction process, the latent space-based diffusion model is similarly employed. Training loss is defined as a joint optimization of the autoencoder and diffusion model during the training process. Experimental results demonstrate that the SteDM outperforms existing steganography schemes in some aspects such as visual effects, security, and robustness.

Keywords: Image Steganography · Diffusion Model · Latent Space · Image Extraction

1 Introduction

Traditional image steganography usually hides secret information in the spatial domain and transformation domain of the image [1]. However, with an increase in the amount of secret information, the visual effect of traditional image steganography schemes decrease, which greatly affect security [2]. Additionally, the high complexity and the design cost obstruct their further development. The popularization of deep neural networks has brought new opportunities for the advancement of image steganography. Baluja et al. [3] proposed the first Convolutional Neural Network (CNN) for image steganography tasks. Then, some studies introduced networks such as U-Net, ResNet, and others into the field of image steganography. However, most of these schemes neglect the utilization of the latent space of images. It results in low-quality generated stego images, the

resistance against analysis and robustness against attacks for the images is not guaranteed.

Recently, an emerging generative model, the Denoising Diffusion Probabilistic Models (DDPM) [4], and its variants have captured extensive attention. Among them, the latent diffusion model (LDM) [5] has demonstrated significant potential in image processing tasks, being applied in text-to-image generation and image anomaly detection. By mapping the image to a perceptual equivalent low-dimensional representation, it simplifies the denoising process of the diffusion model. Compared to the DDPM model, the LDM model is more flexible, generates higher-quality samples, and introduce more conditional mechanisms.

This paper proposes the LDM into image steganography. We apply the diffusion process in the latent space of the cover image. An encoder, similar to a pre-trained one, transforms the secret image into latent vectors. These vectors are incorporated as a conditional variable into the inverse diffusion process via a cross-attention mechanism. A pre-trained decoder then generates the stego image, and secret image extraction also occurs in the latent space. A suppressive decoder recovers the secret image while suppressing the cover image semantic information. The joint optimization of the encoder and diffusion model improves the realism of Stego images. The main contributions of this paper are as follows:

- 1) We propose a latent diffusion paradigm that integrates pre-trained autoencoders with the LDM to execute steganography tasks within the latent space of images. This approach facilitates image feature fusion via a cross-attention mechanism, thereby mitigating the training complexity of diffusion models while maintaining image security.
- 2) During the extraction phase, a suppressive decoder is employed to filter image features and recover concealed images. This approach enhances the utilization of image features and ensures the quality of the restored secret images.
- 3) The loss function integrates the image loss with the noise predictive loss within the diffusion model, thereby enabling a thorough optimization of the model.

The rest of this study is organized as follows. Section 2 reviews the relevant technologies about image steganography schemes. Section 3 provides a detailed introduction to the proposed scheme. Section 4 validates the proposed scheme through various types of experiments. Section 5 summarizes this paper.

2 Related Work

In recent years, deep neural networks have developed rapidly, and researchers have widely introduced deep learning techniques into image steganography tasks, proposing many effective image steganography schemes. Baluja et al. [3] designed three fully convolutional networks corresponding to image preprocessing, hiding, and extraction, marking the first application of CNNs in the image steganography task. Rahim et al. [6] introduced a conventional loss into the training of the steganography model, achieving end-to-end training. Bui et al. [7] mapped binary

secret information into the latent embedding of images and directly injected the secret information into the latent encoding of the locked autoencoder. Kumar et al. [8] incorporated skip connections into the U-Net to meet the requirement of preserving details in the cover image. Duan et al. [9] used a double-layer U-Net network, combined with a spatial-channel attention mechanism, to achieve a comprehensive fusion of image features. Jing et al. [10] designed the HiNet model, applying the concept of invertible neural networks to image steganography tasks. They use multiple dense blocks within a set of network parameters for both steganography and extracting image information. Liu et al. [11] mapped secret information into a binary sequence, achieving lossless steganography of information in an invertible neural network. Zhang et al. [12] proposed a joint adjustment model that batched different resolutions of the same image into the encoder model. Feng et al. [13] replaced the invertible submodules in the HiNet with a swin Transformer block, achieving a satisfactory steganography.

The DDPM is a probabilistic generative model, initially proposed by Sohl-Dickstein et al. [4]. The diffusion process injects noise into the data step by step, thereby achieving gradual degradation of the data. The inverse diffusion process uses a predictive model to infer the noise injected at each step and removes the predicted noise from the data. After multiple rounds of prediction and removal, new data samples can be generated. The LDM transforms the diffusion and inverse diffusion processes from image pixel space to image hidden space based on the DDPM, reducing the consumption of computational resources while enhancing the controllability of generated samples. Researchers have applied the diffusion model to various challenging computer vision tasks.

3 Proposed Method

3.1 Steganography Network

Image steganography involves fusion and interpolation operations between different images, and the network model performing these operations in the pixel domain will face various limitations. Compared to pixel space, deeper dimensions allow us to learn higher-level abstract features and provide more flexible and advanced image processing capability. Figure 1 shows the steganography network of our method, through the specified hidden space paradigm, the proposed scheme flexibly uses diffusion and inverse diffusion processes to convey secret images. We employ a pre-trained perceptual image compression autoencoder to achieve conversion between the pixel space and the latent space,

$$z_{cover_0} = E(x_{cover}). \quad (1)$$

Diffuse the latent space of the cover image by injecting noise multiple times in steps,

$$q(z_{cover_1}, \dots, z_{cover_T} | z_{cover_0}) = \prod_{t=1}^T q(z_{cover_t} | z_{cover_{t-1}}). \quad (2)$$

Combined with the transition kernel in the Markov chain, we have:

$$q(z_{cover_t} | z_{cover_{t-1}}) = N(z_{cover_t}; \sqrt{1 - \beta_t} z_{cover_{t-1}}, \beta_t I), \quad (3)$$

where $\beta_t \in (0, 1)$ determines the degree of noise in the image, and a sufficiently large t turns x into an isotropic Gaussian distribution. Let $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{i=1}^t \alpha_i$, the noise injection process can be further simplified. The relationship between z_{cover_t} and the cover image original latent space z_{cover_0} can be directly represented as:

$$z_{cover_t} = \sqrt{\bar{\alpha}_t} z_{cover_0} + \sqrt{1 - \bar{\alpha}_t} \epsilon, \epsilon \sim N(0, 1). \quad (4)$$

It allows sampling of at any time step without the need for calculations of intermediate steps.

To effectively conceal the secret image, we proposed to incorporate a conditional mechanism into the steganography process, mapping the features of the secret image into the cover image. We designed a secret image encoder with a structure similar to the pre-trained encoder R_θ , encoding the secret image x_{secret} into a latent vector with the same scaling factor σ as the cover image:

$$z_{secret} = R_\theta(x_{secret}). \quad (5)$$

The latent vector of the secret image is treated as a conditional mechanism. During the denoising phase of the inverse diffusion model, the latent spaces of both the cover and secret images are integrated using a cross-attention mechanism:

$$p_\theta(z_{cover_0}, \dots, z_{cover_{T-1}} | z_{cover_T}, z_{secret}) = \prod_{t=1}^T p_\theta(z_{cover_{t-1}} | z_{cover_t}, z_{secret}), \quad (6)$$

$$\begin{aligned} & p_\theta(z_{cover_{t-1}} | z_{cover_t}, z_{secret}) \\ &= N(z_{cover_{t-1}}; \mu_\theta(z_{cover_t}, t, z_{secret}), \Sigma_\theta(z_{cover_t}, t, z_{secret})), \end{aligned} \quad (7)$$

where θ is a parameter, $\mu_\theta(z_{cover_t}, t, z_{secret})$ and $\Sigma_\theta(z_{cover_t}, t, z_{secret})$ represent the mean and covariance parameterized by the neural network, respectively. Consider inference process by using isotropic Gaussian distributions, $p_\theta(z_{cover_{t-1}} | z_{cover_t}, z_{secret})$, which are learned from LDM.

The visual depiction of the cross-attention mechanism is illustrated in Fig. 2, where φ_i represents each network layer in the sampling process, z_{cover_f} is an intermediate representation of the noise prediction network. The cross-attention layer maps z_{secret} to the down-sampling and up-sampling layers of the neural network, thereby synthesizing the final representation vector \hat{z}_{cover_0} . Through the pre-trained decoder D , \hat{z}_{cover_0} will be transformed into the stego image: $x_{stego} = D(\hat{z}_{cover_0})$.

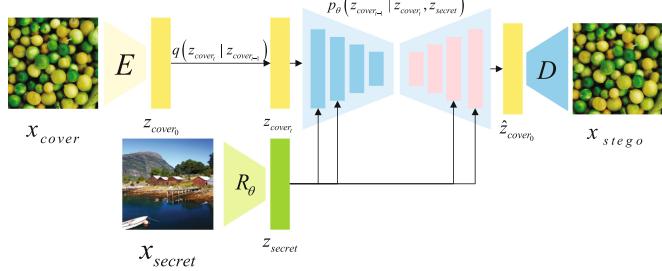


Fig. 1. The structure of the image steganography network.

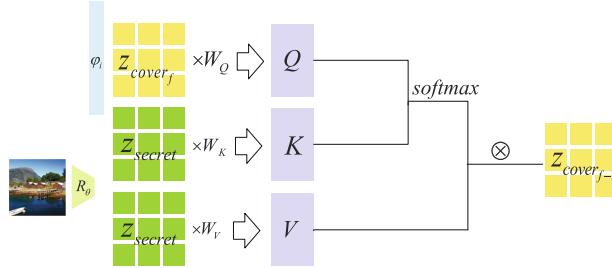


Fig. 2. Visual representation diagram of cross-attention mechanism.

3.2 Extraction Network

The purpose of the extraction network is to recover the secret image, ensuring the effective transmission of information. The extraction network structure is illustrated in Fig. 3, using the same pre-trained encoder E to map the secret image into high-dimensional representation space $z_0 = E(x_{stego})$. Subsequently, analogous to the hidden process, forward and backward diffusion processes are executed. During the extraction phase, multiple fully connected layers are integrated into the denoising neural network. These layers utilize learnable parameters to suppress the features of the cover image while recovering the secret image. Upon obtaining the final feature vector \hat{z}_0 , the newly designed suppressor further accentuates the features of the secret image.

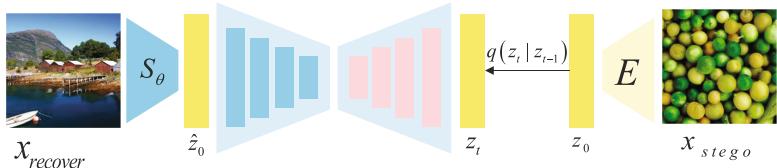


Fig. 3. The structure of the extraction network (S_θ is the suppressive decoder).

The suppressor S_θ mainly performs two tasks: suppressing features belonging to the cover image, emphasizing features in the secret image, and restoring optimized features to the image data space. As shown in Fig. 4, the structure of the suppressor is identical to that of the latent decoder, but it includes an additional residual link, emphasizing the detailed features of the secret image.

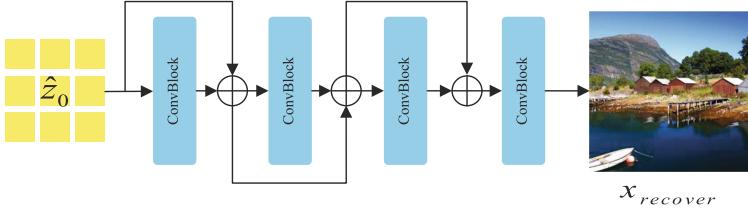


Fig. 4. S_θ : suppression-type decoder structure.

3.3 Training Objective

The denoising neural network in the diffusion model plays a pivotal role in ensuring the image transition to different variants. The combination of forward and backward diffusion can be seen as a variational autoencoder, and its training objective is to minimize the variational lower bound:

$$\begin{aligned} \mathbb{E} [\log p_\theta(z_0)] &\geq \mathbb{E}_q \left| \log \frac{p_\theta(z_{0:T})}{q(z_{1:T} | z_0)} \right| \\ &= \mathbb{E}_q \left[\log p(z_T) + \sum_{t \geq 1} \log \frac{p_\theta(z_{t-1} | z_t)}{q(z_t | z_{t-1})} \right] \\ &= \mathbb{E}_q \left[\log \frac{p(z_T)}{q(z_T | z_0)} + \sum_{t > 1} \log \frac{p_\theta(z_{t-1} | z_t)}{q(z_{t-1} | z_t, z_0)} + \log p_\theta(z_0 | z_1) \right]. \end{aligned} \quad (8)$$

We jointly optimize the parameters of this encoder with the diffusion model. The loss functions for the hidden and extraction stages can be defined as:

$$L_{hidden} = \mathbb{E}_{t, z_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} z_{cover_0} + \sqrt{1 - \alpha_t} \epsilon, t, \tau_\theta(x_{secret})) \right\|^2 \right]. \quad (9)$$

$$L_{extract} = \mathbb{E}_{t, z_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} z_{stego_0} + \sqrt{1 - \alpha_t} \epsilon, t) \right\|^2 \right]. \quad (10)$$

To enable the suppressor to better enhance the features of the secret image adaptively while suppressing the features of the cover image through joint

optimization with τ_θ , the loss function of the extraction phase can be further integrated as:

$$L_{extract} = \mathbb{E}_{t, z_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\alpha_t} z_{stego_0} + \sqrt{1 - \alpha_t} \epsilon, t) \right\|^2 \right] + \|x_{secret} - S_\theta(\hat{z}_{stego_0})\|^2 \quad (11)$$

4 Experiment Results and Analysis

Our experiments utilize the DIV2K to train the proposed image steganography model. During the training process, we employ the Adam optimizer with an initial learning rate of 0.001, which is progressively decayed throughout the training. To facilitate the effective transformation of images from the data space to the latent space, our pre-training model incorporates the pre-trained VQGAN encoder [14].

4.1 Visual Effects

The visual quality of images is the primary metric for evaluating steganography schemes. Pixel distribution histograms can record the frequency of pixel distribution on different channels. The greater the differences between images, the larger the discrepancies between their pixel distribution histograms. We crop the central 256×256 area of the images to compare pixel histograms. According to the results in Fig. 5, it's challenging to visually perceive the changes in pixel distribution among images through histogram testing. This suggests that during the hiding phase, the distribution of hidden locations for secret image characteristic information is uniform, causing minimal impact on the pixel distribution of the cover image.

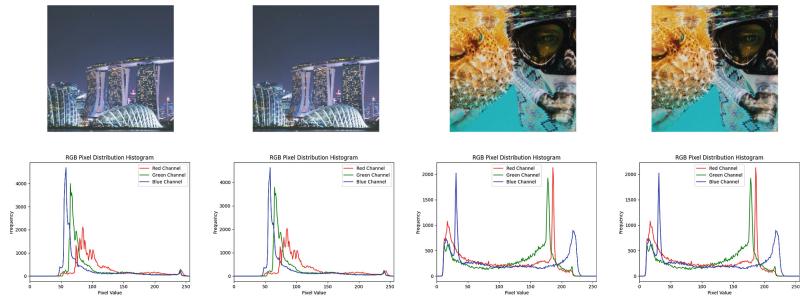


Fig. 5. Experimental results and pixel histograms.

4.2 Capacity Analysis

The hidden capacity of data information is a crucial evaluation metric for image steganography schemes. The proposed approach in this paper can conceal an RGB image within another RGB image of the same resolution. In the capacity analysis, the effective capacity (EC) was chosen to assess the hidden capability of the model. The calculation formula for EC is $EC = NS/NC$, where NS represents the quantity of secret information to be hidden (unit: bits), and NC is the number of pixels in the cover image. Table 1 lists several common traditional image steganography schemes and recent deep-learning steganography schemes. The effective capacity of the image steganography scheme proposed in this paper is 8 bits per pixel (8bpp). It's on par with similar steganography schemes in terms of capacity and significantly surpasses traditional image steganography schemes. Our steganography schemes employ higher image resolutions, typically 2 to 4 times larger than common resolutions. This indicates that our design of moving image steganography to the latent space of images is more efficient.

Table 1. Comparison of information capacity.

Schemes	Cover Image Size	Secret Data Size	EC (bpp)
Ma [15]	512×512 (gray)	1741 (bit)	< 0.25
Liu [16]	512×512 (RGB)	32×32 (RGB)	< 0.25
Feng [13]	224×224 (RGB)	224×224 (RGB)	8
Jing [10]	256×256 (RGB)	256×256 (RGB)	8
Ours	1024×1024 (RGB)	1024×1024 (RGB)	8

4.3 PSNR and SSIM

Peak Signal-to-Noise Ratio (PSNR) calculates the pixel differences between two images, the higher the similarity between the images, the larger the measured PSNR value. The PSNR calculation formula is as follows:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{(2^n - 1)^2}{MSE(I, I_a)} \right). \quad (12)$$

Structural Similarity Index (SSIM) measured the differences in contrast, brightness and structure between two images. The range of values is $(-1, 1)$, when two images are exactly the same, the measurement value is 1. The SSIM calculation formula is as follows:

$$\text{SSIM}(I, I_a) = \frac{(2\mu_I\mu_{I_a} + C_1)(2\text{cov} + C_2)}{(\mu_I^2 + \mu_{I_a}^2 + C_1)(\sigma_I^2 + \sigma_{I_a}^2 + C_2)}, \quad (13)$$

where μ_I and μ_{I_a} are the average values of I and I_a , σ_I^2 and $\sigma_{I_a}^2$ are the variances of I and I_a , cov is the covariance of I and I_a , C_1 and C_2 are used to avoid the denominator being zero. As shown in Table 2, compared to the schemes of [3] and [6], our scheme has more advantages in PSNR and SSIM measurements. Moreover, when applied to the same dataset, our approach also outperforms the schemes of [13] and [10].

Table 2. Comparison of PSNR and SSIM.

Schemes	Index	Stego Image	Recover Image
Baluja [3]	PSRN	36.7	35.75
	SSIM	0.96	0.955
Rahim [6]	PSRN	32.5	34.75
	SSIM	0.94	0.93
Feng [13]	PSRN	34.32	44.05
	SSIM	0.935	0.991
Jing [10]	PSRN	48.99	52.86
	SSIM	0.997	0.9992
Ours	PSRN	49.88	51.40
	SSIM	0.9926	0.9994

4.4 Residual Analysis

We conducted residual analysis on the experimental images and enhanced the residual images. In Fig. 6, the residual images between the cover and stego images appear predominantly black. Even when enlarged by 20 times, the texture information of other images cannot be discerned. This demonstrates that our scheme can achieve high-quality secure image steganography.

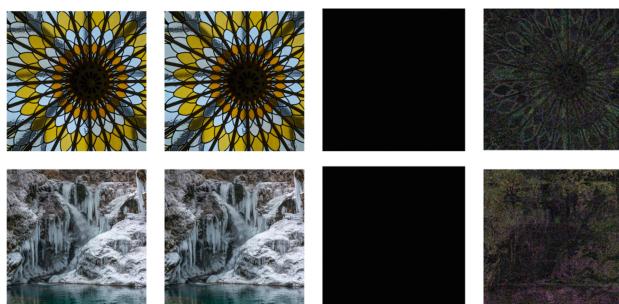


Fig. 6. Experimental results and residual images.

5 Conclusion

This paper presents the integration of a latent diffusion model within an image steganography framework, leveraging a pre-trained autoencoder in conjunction with the diffusion model to facilitate steganography and extraction within the image latent space. Distinct from conventional feature integration methods, our approach utilizes a cross-attention mechanism during the inverse diffusion process, wherein the feature vectors of the secret image are conditionally combined with those of the cover image. The training phase employs a loss function to jointly optimize both the autoencoder and the diffusion model. In subsequent research endeavors, we intend to conduct an in-depth investigation into the potential applications of diffusion models and pre-trained autoencoders within the domain of image steganography.

Acknowledgments. This research was funded by the NSFC under Grant 62462012, and Science and Technology Program of Hebei under Grant 22567606H.

References

1. Aghababaiyan, K.: Novel distortion free and histogram based data hiding scheme. *IET Image Proc.* **14**(9), 1716–1725 (2020)
2. Guo, H., Xue, J.: The analysis of watermarking capacity of packing model and bits replacement model. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 2603–2607 (2016)
3. Baluja, S.: Hiding images in plain sight: deep steganography. In: Proceedings of Advances in Neural Information Processing Systems (NIPS), pp. 2069–2079 (2017)
4. Sohl-Dickstein, J., Weiss, E.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning, pp. 2256–2265 (2015)
5. Rombach, R., Blattmann, A.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
6. Rahim, R., Nadeem, S.: End-to-end trained CNN encoder-decoder networks for image steganography. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–6 (2018)
7. Bui, T., Agarwal, S., Yu, N., Collomosse, J.: RoSteALS: robust steganography using autoencoder latent space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 933–942 (2023)
8. Kumar, A.: Encoder-Decoder architecture for image steganography using skip connections. *Proc. Comput. Sci.* **218**(4), 1122–1131 (2023)
9. Duan, X.: DUIANet: a double layer U-Net image hiding method based on improved inception module and attention mechanism. *J. Vis. Comm. Image Repres.* **98**, 104035 (2023)
10. Jing, J., Deng, X.: HiNet: deep image hiding by invertible network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4733–4742 (2021)
11. Liu, L.: Lossless Image steganography based on invertible neural networks. *Entropy* **24**(12), 1762 (2022)

12. Zhang, L.: Joint adjustment image steganography networks. *Sig. Proc. Image Comm.* **118**, 117022 (2023)
13. Feng, Y., Liu, Y., Wang, H., Dong, J.: Image Hide with invertible network and Swin Transformer. In: International Conference on Data Mining and Big Data, pp. 385–394 (2022)
14. Esser, P., Rombach, R.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873–12883 (2021)
15. Ma, K.: Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inform. Foren. Secur.* **8**(3), 553–562 (2013)
16. Liu, D.: A fusion-domain color image watermarking based on HAAR transform and image correction. *Exp. Syst. Appl.* **170**, 114540 (2021)



BDFC:A New Flow Control Mechanism for Torus Networks

Haofei Zhang¹, Youmeng Li¹(✉), Yu Deng², and Qi Fang²

¹ College of Intelligence and Computing, Tianjin University, Tianjin 300072, China

{liyoumeng,zhanghaofei}@tju.edu.cn

² Phytiun Technology Co., Ltd., Changsha 410073, China

{equal.dy,fangqi}@phytiun.com.cn

Abstract. FBFC (Flit Bubble Flow Control) and Dateline flow control are two commonly used flow control mechanisms in Torus networks. However, each of these two flow control mechanisms has its own performance limitations. The bubble flow control mechanism can lead to starvation, and dateline flow control can decrease channel cache utilization. We propose BDFC (Bubble Dateline Flow Control) technology, which combines the above two flow control mechanisms. It greatly alleviates the hunger problem of bubble flow control technology without increasing costs, while also increasing the cache utilization of Dateline technology, thereby improving the performance of the entire network. In 8×8 Torus, its saturated throughput has increased by a maximum of 44% compared to FBFC. Improved by a maximum of 68% compared to Dateline.

Keywords: Torus on-chip network · bubble flow control · cache utilization · Dateline

1 Introduction

In on-chip networks, torus networks are a structurally symmetric topology that can effectively reduce router hops and network latency [15]. These characteristics can effectively prevent data packets from being congested in the middle area like a mesh network during transmission. Due to its structural advantages, many products in industry adopt torus topology, such as SGI's BlueMountain and IBM's BlueGene/Q [4]. On the other hand, due to the addition of loopback links, the Torus network also has natural CDG loops in every dimension, making it more prone to deadlocks. This forces designers to invest more effort to avoid deadlocks at the minimum cost.

There are several common solutions:

The first one is the Dateline scheme proposed by Dally et al. [7]. It requires manually setting a date change line and at least two virtual channels, and setting priority for the virtual channels. Within each dimension of the loop, only high priority can cross the date change line, and when crossing the date change line, it is necessary to switch to a low priority virtual channel. This prevents the formation of CDG rings and avoids the occurrence of deadlocks [5]. However, setting

priorities and allocation rules on virtual channels can reduce cache utilization, thereby affecting network throughput.

The second is the escape virtual channel proposed by Duato [8], which proves that the CDG ring is a sufficient and unnecessary condition for the network to have no deadlocks. Even if there is a loop in the CDG, as long as there is a non cyclic subset in the CDG, this subset can be used to relieve cyclic dependencies. However, in the Torus network, due to the natural presence of loops in each dimension, its non cyclic sub parts still require special methods to avoid deadlocks.

The third method is to use bubble flow control scheme [3]. It reserves idle cache as bubbles within the ring and controls that newly injected and out of ring converted packets cannot occupy the reserved space, thereby ensuring the avoidance of deadlock within the ring. The latest bubble flow control scheme has reduced the size of bubbles to flit size [12]. However, due to its sole virtual channel and reserved bubbles, it leads to leader blocking and starvation.

2 Related Work

Flow control is responsible for managing the allocation of network buffers and links. A good flow control mechanism can reduce information transmission delay under small load conditions without increasing resource allocation overhead, and improve network throughput by achieving effective sharing of message buffers and links. Traditional flow control mechanisms mainly include storage and forwarding [7], virtual passthrough [10], virtual channels [7], and wormhole flow control [6]. Later, due to the existence of natural CDG rings in the Torus network, some researchers began to propose new methods to avoid deadlocks through flow control [2]. The main purpose is to reserve a bubble of message size in the CDG ring to ensure that messages in the ring can always enter in an orderly manner to avoid deadlocks. Subsequently, many researchers have conducted research on this basis [1, 12, 13, 16], reducing the size of bubbles from a single open packet to a slice size to avoid deadlocks and extending it to adaptive routing.

In recent years, many researchers have conducted extensive research on flow control schemes. Xiao Canwen from the National University of Defense Technology proposed a new type of wormhole bubble flow control, which allows flit to store different packets in an unordered manner and be able to route them separately. He also proved that this strategy can ensure that deadlocks do not occur in one-way loops [1]. Some researchers have also begun to study the feasibility of bubble schemes in irregular topologies [14]. Aniruddh Ramrakhyani et al. from the Georgia Institute of Technology proposed a static bubble scheme that can be applied to the underlying network during design and expands the router subset by adding buffers to ensure that any dependency chain has at least one static atmosphere to ensure that the network is deadlock free. Hossein Farrakhbakht et al. from the University of Toronto proposed a novel flow control called fast-flow, which enhances the level of packets through time division multiplexing and

defines non overlapping channels for bufferless transmission of these high priority packets. This flow control method not only provides high throughput but also solves protocol and network level deadlocks [9].

3 Motivation

As shown in Fig. 1, in traditional Dateline, messages need to first select the higher priority virtual channel VC0 to send at the beginning of transmission. When the message crosses the date change line, it will be assigned to virtual channel VC1. During message transmission, messages transmitted by virtual channel VC0/VC1 can only be sent to virtual channel VC0/VC1. Only when the message crosses the date change line, high priority virtual channels are allowed to apply for lower priority virtual channels. We can see that the utilization rate of the blank cache in the graph is very low, unless the destination node of the packet is these routers, these caches are almost always idle.

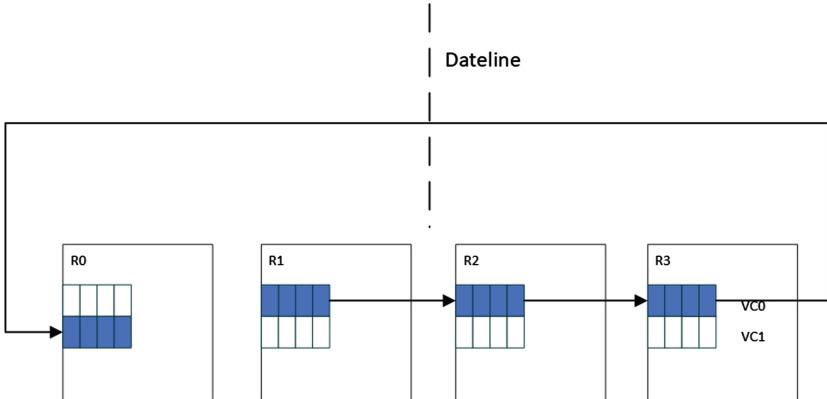


Fig. 1. Dateline flow control.

Although flit bubble flow control technology addresses intra-dimensional deadlocks and optimizes cache utilization, in cross-dimensional message transmission, converted data packets are prohibited from occupying critical bubbles. Downstream routers' ordinary idle cache must be large enough to accommodate the entire message to facilitate its transmission. These limitations have led to a starvation phenomenon in the flow control mechanism, thereby impacting network performance. Furthermore, the longer the packet length, the more severe the impact.

As shown in Fig. 2 There are four routers R0, R1, R2, and R3 within a certain dimension. The gray squares represent key bubbles, which can only be used by messages within the ring, and cannot be used by messages outside the ring. If the key bubble is occupied by the upstream packet slice, the idle slice bubbles

in the upstream will be marked as the key bubble. The key bubbles ensure that messages within the ring can be transmitted in an orderly manner and there are no deadlocks within the ring. The white squares represent idle cache, which can be used for both intra - and extra dimensional packets. The blue squares represent that these caches in the virtual channel are being occupied by packet slicing. The green square represents the message outside the ring. If this message wants to enter the current dimension, it cannot use gray key bubbles.

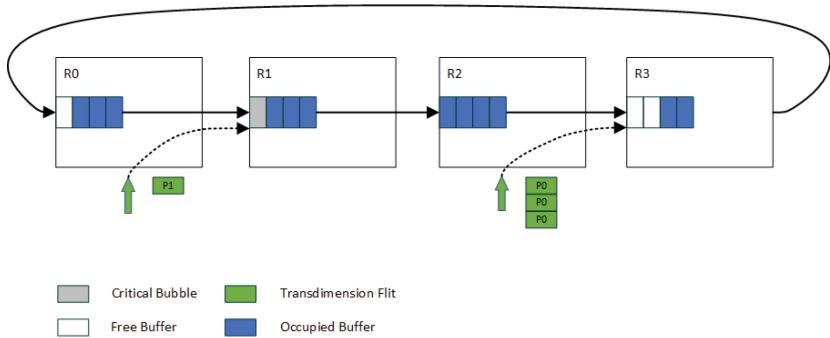


Fig. 2. High injection rate starvation phenomenon.

The reason for hunger phenomenon is that the cache resources required for injected or converted packets are higher than those required for packet transmission within the ring [2]. In Fig. 2, both outer ring messages P0 and P1 do not meet the dimension conversion requirements. Although the packet length of P1 is only 1, there is only one idle cache in downstream router R1, and this idle cache is a critical bubble. Because the critical bubble cannot be occupied by out of loop packets, P1 does not meet the dimension conversion condition. The packet length of P0 is 3, and its downstream router R3 only has two idle caches, which do not meet the dimension conversion condition. The typical approach to addressing this starvation issue is to disable packet injection and dimensionality transfer for all routers in the ring except R1. This enables packets within the ring to continue their journey to the destination router or to other dimensions, ensuring that there is sufficient free cache in the current dimension to accommodate the dimensionality transfer of P1 packets. The injection and transmission of maintenance messages from other routers will only resume after the P1 maintenance conversion is successful. P0 is the same, but P0 packets also face the phenomenon of R2 router's short packets seizing R3 cache, which will make it more difficult for P0 long packets to meet the conversion conditions.

In addition, there is another hunger phenomenon in FBFC. As shown in Fig. 3. When the injection rate of the network is relatively low, crucial bubbles in the network may stay in the R3 router for a long time, resulting in P0 packets not meeting the dimension conversion conditions. The traditional solution is to periodically move crucial bubbles to routers with idle cache in the neighborhood.

For the aforementioned two types of starvation, existing solutions necessitate additional hardware costs. Most critically, they compromise the packet transmission capabilities of other routers within the ring, representing a reactive and passive strategy.

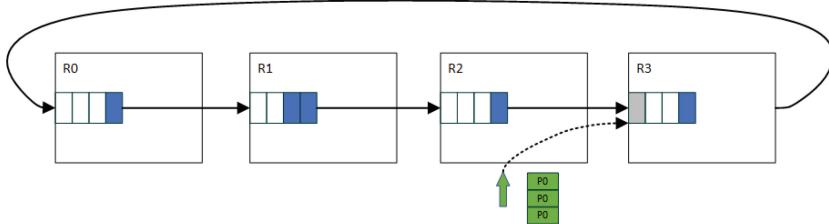


Fig. 3. Low injection rate starvation phenomenon.

Faced with the aforementioned hunger phenomenon, we present a solution to mitigate the issue of queue leader blocking. This is achieved by establishing two virtual channels at each port. One of these channels eliminates the need for a critical bubble. Messages can be transmitted as if within a ring, without the requirement of atomicity, thereby addressing the hunger phenomenon associated with dimensionality transformation and injection.

4 Bubble Dateline Flow Control

Bubble Dateline flow control, a novel flow control mechanism, addresses the issues of leader blockage, transition, and injection hunger in flit bubble flow control technology by employing the Dateline concept. At the same time, the characteristics of bubble flow control are utilized to improve the cache utilization of Dateline.

As shown in Fig. 4. In the traditional Dateline strategy, message P1 needs to switch from VC0 to VC1 when crossing the date change line (Dateline) to avoid interdependence between messages. In the previous FBFC scheme, if only one virtual channel is available and there may be key bubbles in message P2 during dimensionality conversion, it must satisfy atomicity when converting, that is, downstream ordinary cache needs to accommodate the size of the entire message before it can be converted. However, when a crucial bubble is added to VC0, message P1 can choose either virtual channel VC1 or virtual channel VC0, reducing the restrictions on virtual channel selection and increasing cache utilization. In addition, message P2 can choose virtual channel VC1 without any critical bubbles during dimensionality conversion, which does not require atomicity and greatly improves its transmission performance during dimensionality conversion. And it also solves the queue head blocking problem of FBFC, that is, when P2 is blocked, the packets in the virtual channel VC0 of R1 router are still transmitted normally.

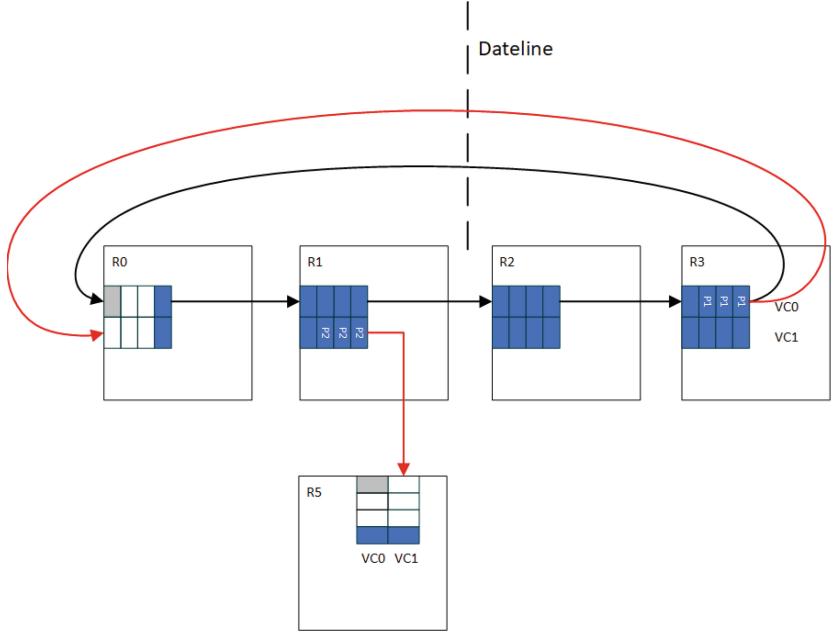


Fig. 4. BDFC theoretical description.

Based on the above theories, we provide a specific flow control design scheme for BDFC (Bubble Dateline Flow Control). (1) We add a crucial bubble in each dimension of the VC0 virtual channel to avoid deadlock in the virtual channel VC0. The packets injected into this virtual channel still need to comply with the atomic requirements of bubble flow control. (2) In order to maintain bubbles and avoid deadlocks during message transmission, we have imposed conditional restrictions on the transmission of messages. Once a virtual channel is selected in the current dimension, it cannot be changed arbitrarily during the transmission process in the current dimension, unless it will cross the date change line or transfer to another dimension. (3) For injection and dimension conversion messages, it will be determined whether the message will cross the date change line in the current dimension. If so, virtual channel VC0 must be selected. (4) For packets that do not pass through the date change line in the current dimension and those that are about to pass through the date change line, congestion awareness will be applied. Based on the congestion level of downstream routers, network transmission will be selected on virtual channel VC0 or virtual channel VC1.

We have made the following deadlock free discussion on the flow control scheme. The condition for deadlock in the dimensional ring is that the packets transmitted between routers within the ring have a cyclic dependency, which prevents all packet slices from moving forward, resulting in a deadlock. There

are two options for messages that need to cross the date change line within the dimension.

1. Choose to switch from virtual channel VC0 to virtual channel VC1. At this time, the transmission of messages will not form a CDG ring, so it will not generate cyclic dependencies between messages and will not generate deadlocks. At this point, its deadlock should avoid being consistent with the Dateline.
2. The next hop continues to use virtual channel VC0. At this time, the transmission of packets will form a CDG ring, but there is a critical bubble inside the ring that does not allow injection or conversion of packets. Therefore, when a packet slice within the same dimension uses this slice bubble, it will generate an idle cache on another router in the same dimension ring (upstream router using this bubble) and relabel it as a critical bubble. The packets in the reciprocating loop can always be transmitted in an orderly manner until they reach the destination router or enter another dimension. At this point, its deadlock avoidance method is the same as FBFC.

5 Performance Analysis

5.1 Experimental Parameter Settings

Booksim2 is an open-source, clock-cycle accurate NoC software simulator that modularizes various components of the NoC, supporting multiple topologies, routing algorithms, router microstructures, and arbiter algorithms. It also features high scalability and detailed behavioral analysis. We modified Booksim2 to implement BDFC, FBFC-C (Flit Bubble Flow Control-Critical, hereinafter referred to as FBFC), and Dateline for performance comparison. Both BDFC and Dateline flow control schemes have two virtual channels, and the Flit bubble Flow Control (FBFC) has one virtual channel. Given the network contains a significant number of single-slice packets [11], the experiment set the proportion of single-slice packets to 80%, with the remaining 20% of packets containing three-slice sizes. The specific parameter configuration of the experiment is shown in Table 1

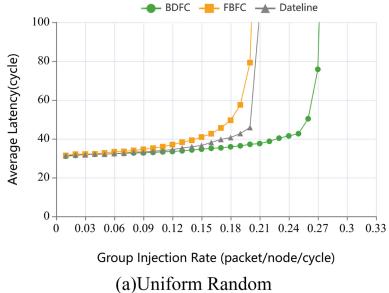
Table 1. Experimental parameter configuration table.

Parameters	Values
Network Topology	4×4torus; 8×8torus; 16×16torus
Traffic Patterns	uniform random, transpose, bitcomp, shuffle, neighbor, bitrev, tornado, randperm
Routing Algorithm	XY
VC	BDFC (2VCs), Dateline (2VCs), FBFC-C (1VC)
Buffers/VC	6flits
Warm-up time	3000cycles
Run time	7000cycles

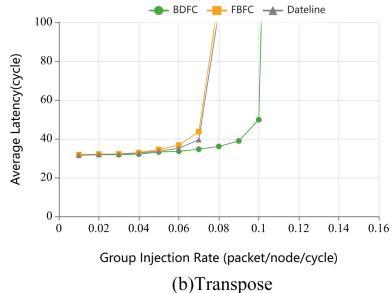
The topology structures used in the experiment are 4×4 , 8×8 , and 16×16 Torus networks. The routing algorithm used is XY routing, which is often used in conjunction with bubble flow control, because the routing algorithm has fewer dimensionality conversions, which can slightly alleviate the hunger problem of bubble flow control dimensionality conversion. Compare the performance of the synthetic traffic mode in uniform random, transpose, bitrev, bitcomp, shuffle, and other modes. In uniform random, each router will send the same number of packets to other routers in the network. The source node and destination node are symmetrical under the transpose, which may result in a large number of packets in the network that need to be transposed. Under bitrev, the router will send packets to other routers in order and eventually return to the first destination router, forming a loop. If a router wants to send a message to a destination router under bitcomp, the destination router will also send a message to the source router. In shuffle mode, the destination router of the message is not fixed and varies according to some random variation rule. In BDFC and Dateline, each router port contains two virtual channels, and each virtual channel contains a cache of six slice sizes. In FBFC, each router port has a cache of 12 slice sizes because it only has one virtual channel. In addition, the warm-up time of the on-chip network in this experiment is 3000 clock cycles, and the remaining 7000 clock cycles are collected as performance sampling cycles.

Figure 5 and Fig. 6 shows the relationship between the delay and injection rate of BDFC, FBFC, and Dateline in eight traffic modes, including uniform random, transfer, shuffle, and bitcomp, in an 8×8 network. The performance of the three traffic modes is not significantly different when the injection rate is low, but as the injection rate increases, the hunger phenomenon of FBFC and the problem of insufficient cache utilization of Dateline gradually limit their performance. Due to the fact that BDFC has a virtual channel during dimensionality conversion and injection that does not require atomicity, its performance has been improved compared to FBFC, with an average performance improvement of 52%. Compared to Dateline flow control BDFC, it has optimized load balancing when crossing date change lines and injection. It can select virtual channels based on whether the message crosses the timeline and the congestion level of the next hop router. Moreover, the addition of bubbles greatly improves the utilization of its virtual channel VC0 during time crossing, resulting in a performance improvement of 60% compared to the average performance of Dateline. FBFC and Dateline have similar performance. In the uniform random traffic mode, the starvation of FBFC has a great impact on the performance of the network when the injection rate is high because of the large number of packets and uniform. However, in shuffle and bitcomp traffic modes, there are more packets crossing the time line, so the problem of insufficient utilization of border routers and virtual channels not crossing the time line is more prominent, which limits the performance of the network, and its performance in these two traffic modes is lower than that of FBFC.

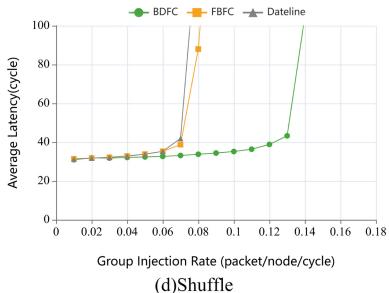
In terms of hardware overhead, BDFC and FBFC have also added two registers on the basis of the original router. It is used to determine whether the con-



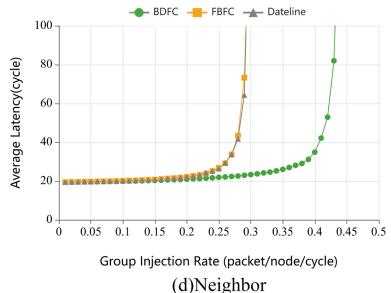
(a)Uniform Random



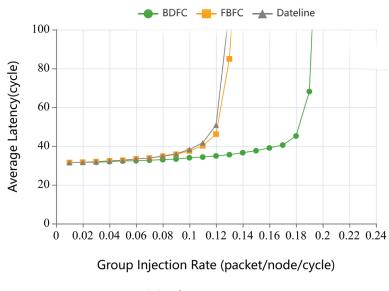
(b)Transpose



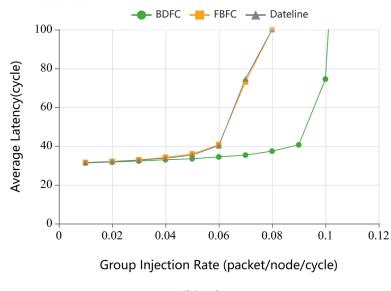
(d)Shuffle



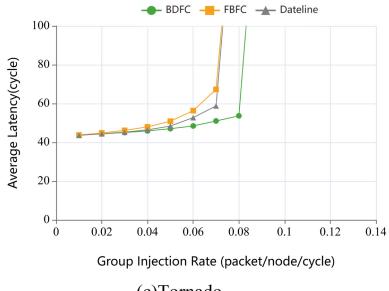
(c)Neighbor

Fig. 5. Performance in uniform random, transpose, shuffle and neighbor traffic mode.

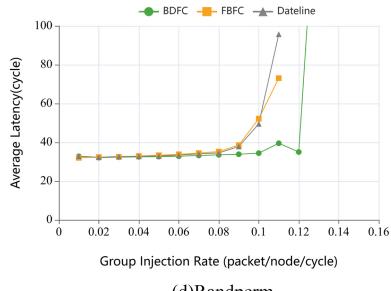
(a)Bitcomp



(b)Bitrev



(c)Tornado



(d)Randperm

Fig. 6. Performance in bitcomp, bitrev, tornado and randperm traffic mode.

verted or injected message meets the application conditions when applying for a virtual channel containing critical bubbles. It can be pre-calculated and therefore does not occupy the critical path. Another register is used to record whether the downstream router type contains critical bubbles. In addition, compared to FBFC flow control, BDFC requires a slightly larger router allocation scale as it requires two virtual channels. But BDFC does not require additional connection resources like FBFC to transmit hunger signals. For Dateline, although BDFC has not increased the allocator size, it will add two registers. Overall, the BDFC scheme alleviates the hunger phenomenon of FBFC and the low cache utilization of Dateline strategy with almost no additional overhead.

5.2 Analysis of Saturation Throughput and Scalability of BDFC

Figure 7 shows the saturated throughput of three flow control mechanisms in uniform random and transpose traffic modes in an 8×8 torus network. When the injection rate is low, there is almost no difference in throughput among the three traffic modes, and they are all proportional to their injection rate. However, as the injection rate increases, congestion in inter dimensional message transmission in FBFC becomes more frequent, and there are more and more cross timeline messages in Dateline. This may lead to cross timeline messages being blocked due to competing for a low priority virtual channel to avoid deadlock, but the utilization of another virtual channel is very low. The average throughput of the BDFC flow control scheme in these two traffic modes has increased by 32% and 35% respectively compared to FBFC and Dateline. Especially in the uniform random traffic mode, its improvement is more significant, with a 64% increase compared to Dateline.

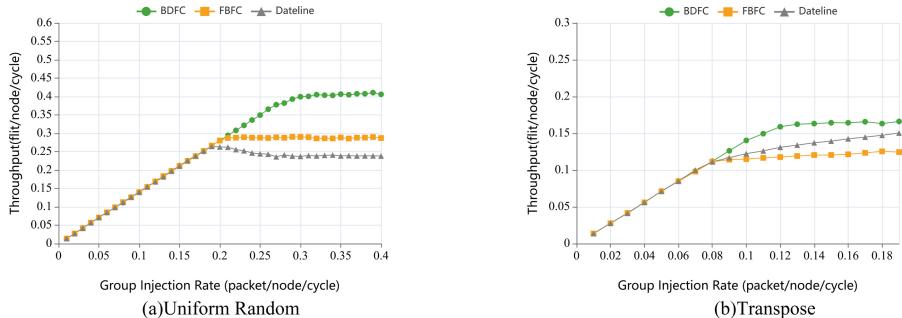


Fig. 7. Saturated throughput in uniform random and transpose modes.

Figure 8(a) shows the performance of three flow control schemes in 4×4 torus and 16×16 torus networks when the packet length is 3 slice sizes, the cache size of each port is 12 slice spaces, and the traffic mode is Uniform Random. Evaluated the scalability of BDFC at different network scales. From the graph, it can be

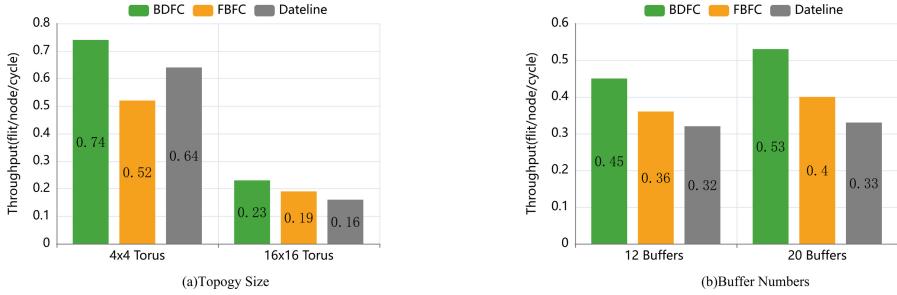


Fig. 8. Performance under different topology scales and buffer number.

seen that in the 4×4 torus network, the BDFC flow control design has improved by 42% and 16% compared to the FBFC and Dateline designs, respectively. In the 16×16 torus network, the BDFC flow control scheme has improved by 21% and 43% compared to the FBFC and Dateline schemes, respectively. As the network size increases, the average number of hops for packet transmission also increases. In the Dateline scheme, the injection of packets and the limitations of transmission have an impact on the performance of the network. In the FBFC scheme, its hunger handling mechanism is to fight for cache space for currently hungry packets by prohibiting the conversion and injection of other routers in the ring. However, as the network size increases, the more routers it needs to

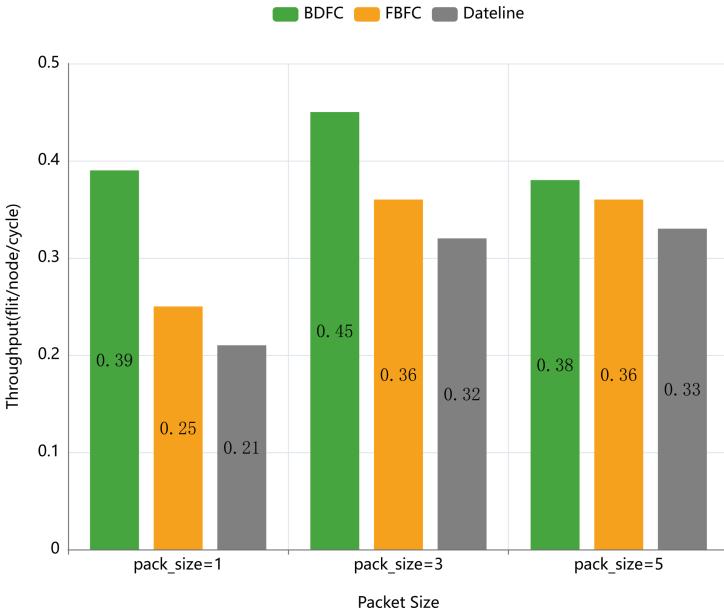


Fig. 9. Packet length sensitivity experiment.

ban, the more packets it needs to ban. This results in FBFC solutions requiring more time and space costs to handle congestion. The 16×16 topology structure constitutes a significant network scale within a single chip. To further expand the network scale across chips, chiplet technology is typically employed; therefore, the scalability analysis is limited to a maximum network size of 16×16 .

Figure 8(b) shows three flow control designs with a network size of 8×8 torus, a traffic mode of uniform random, and a message length of three slices. Performance comparison between two cache configurations. The first configuration involves configuring 12 cache units per router port, while the second configuration involves configuring 20 cache units per router port. From the graph, we can see that when the cache configuration is set to 12 slice sizes, the BDFC flow control scheme has improved by 25% and 41% compared to FBFC and Dateline designs, respectively. When the cache is configured to 20 slice sizes, the BDFC flow control design improves by 32% and 60% compared to FBFC and Dateline designs, respectively. This is because as the cache units increase, BDFC's high utilization of cache units and the advantage of being able to use hunger free virtual channels during dimensionality conversion can be more prominent.

5.3 Sensitivity Analysis of Packet Length

Figure 9 shows the saturation throughput of BDFC, FBFC, and Dateline at different packet lengths, with the measured traffic pattern being uniform random. Because each virtual channel of BDFC and Dateline only contains a cache of 6 slice sizes, experiments were conducted to test the saturation throughput of three flow control mechanisms at packet lengths of single slice, 3 slice, and 5 slice. From the graph, it can be seen that when the packet length is 1 and 3, the saturated throughput of BDFC is higher than that of FBFC and Dateline. However, when the packet length is greater than half of its virtual channel depth, its throughput will decrease slightly, only slightly higher than FBFC and Dateline. This is mainly because there are packets in BDFC that will definitely cross the timeline to apply for virtual channel VC0, which requires atomicity. When the packet length is greater than half of the virtual channel length, it will make it easier for packet starvation to occur, which will affect the entire network. However, because the virtual channel depth of FBFC is 12 slice sizes, the slice length at this time does not affect the throughput of FBFC. Only when the packet length is greater than 6, which is also half of the virtual channel depth, will it have a significant impact on the throughput of FBFC.

6 Conclusion and Future Work

Both bubble flow control and Dateline scheme can effectively avoid deadlocks in torus networks. However, the hunger phenomenon of bubble flow control and the insufficient utilization of cache schemes by Dateline scheme each limit the performance of these two flow control mechanisms. The bubble Dateline flow control scheme proposed in this article combines the advantages of the two flow

control schemes mentioned above, and alleviates the hunger phenomenon of the bubble flow control mechanism through a virtual channel that does not require atomicity. By adding a crucial bubble to one of the virtual channels, the limitation of Dateline's selection of virtual channels when crossing the date change line is lifted, thereby improving its cache utilization.

In the future, we will consider whether it is possible to provide characteristics of the lifecycle of crucial bubbles, or to label more key bubbles within the same latitude when the network injection rate is low, and release these bubbles regularly when waiting for starvation, in order to completely solve the starvation phenomenon of bubble flow control species.

References

1. Canwen, X., Minxuan, Z., Yong, D., Zhitong, Z.: Dimensional bubble flow control and fully adaptive routing in the 2-d mesh network on chip. In: 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 1, pp. 353–358. IEEE (2008)
2. Carrion, C., Beivide, R., Gregorio, J.Á., Vallejo, F.: A flow control mechanism to avoid message deadlock in k-ary n-cube networks. In: Proceedings Fourth International Conference on High-Performance Computing, pp. 322–329. IEEE (1997)
3. Chen, L., Wang, R., Pinkston, T.M.: Critical bubble scheme: an efficient implementation of globally aware network flow control. In: 2011 IEEE International Parallel & Distributed Processing Symposium, pp. 592–603. IEEE (2011)
4. Chen, L., Wang, R., Pinkston, T.M.: Efficient implementation of globally-aware network flow control. *J. Parallel Distrib. Comput.* **72**(11), 1412–1422 (2012)
5. Dally, S.: Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. Comput.* **100**(5), 547–553 (1987)
6. Dally, W.J., Seitz, C.L.: The torus routing chip. *Distrib. Comput.* **1**, 187–196 (1986)
7. Dally, W.J., Towles, B.P.: Principles and Practices of Interconnection Networks. Elsevier, Amsterdam (2004)
8. Duato, J.: A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Trans. Parallel Distrib. Syst.* **4**(12), 1320–1331 (1993)
9. Farrokhabkht, H., Gratz, P.V., Krishna, T., San Miguel, J., Jerger, N.E.: Stay in your lane: a NOC with low-overhead multi-packet bypassing. In: 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp. 957–970. IEEE (2022)
10. Kleinrock, L.: A new computer communication switching technique. *Comput. Netw.* **3** (1979)
11. Ma, S., Jerger, N.E., Wang, Z.: Whole packet forwarding: efficient design of fully adaptive routing algorithms for networks-on-chip. In: IEEE International Symposium on High-Performance Comp Architecture, pp. 1–12. IEEE (2012)
12. Ma, S., Wang, Z., Liu, Z., Jerger, N.E.: Leaving one slot empty: flit bubble flow control for torus cache-coherent NoCs. *IEEE Trans. Comput.* **64**(3), 763–777 (2013)
13. Ouyang, Y., Sun, C., Li, R., Wang, Q., Li, J.: Transit ring: bubble flow control for eliminating inter-ring communication congestion. *J. Supercomput.* **79**(2), 1161–1181 (2023)
14. Ramrakhyani, A., Krishna, T.: Static bubble: a framework for deadlock-free irregular on-chip topologies. In: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 253–264. IEEE (2017)

15. Shin, M., Kim, J.: Leveraging torus topology with deadlock recovery for cost-efficient on-chip network. In: 2011 IEEE 29th International Conference on Computer Design (ICCD), pp. 25–30. IEEE (2011)
16. Wang, R., Chen, L., Pinkston, T.M.: Bubble coloring: avoiding routing-and protocol-induced deadlocks with minimal virtual channel requirement. In: Proceedings of the 27th International ACM Conference on International Conference on Supercomputing, pp. 193–202 (2013)



A Hybrid Vectorized Merge Sort on ARM NEON

Jincheng Zhou¹ , Jin Zhang¹ , Xiang Zhang^{2,3} , Tiaojie Xiao^{2,3} , Di Ma² , and Chunye Gong^{2,3,4}

¹ School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

² College of Computer, National University of Defense Technology, Changsha 410073, Hunan, China

³ Laboratory of Digitizing Software for Frontier Equipment, National University of Defense Technology, Changsha 410073, China

⁴ National Supercomputer Center in Tianjin, Tianjin 300457, China
gongchunye@126.com

Abstract. Sorting algorithms are the most extensively researched topics in computer science and serve for numerous practical applications. Although various sorts have been proposed for efficiency, different architectures offer distinct flavors to the implementation of parallel sorting. In this paper, we propose a hybrid vectorized merge sort on ARM NEON, named NEON Merge Sort for short (NEON-MS). In detail, according to the granted register functions, we first identify the optimal register number to avoid the register-to-memory access, due to the write-back of intermediate outcomes. More importantly, following the generic merge sort framework that primarily uses sorting network for column sort and merging networks for three types of vectorized merge, we further improve their structures for high efficiency in an unified *asymmetry* way: 1) it makes the optimal sorting networks with few comparators become possible; 2) hybrid implementation of both serial and vectorized merges incurs the pipeline with merge instructions highly interleaved. Experiments on a single FT2000+ core show that NEON-MS is 3.8 and 2.1 times faster than std::sort and boost::block_sort, respectively, on average. Additionally, as compared to the parallel version of the latter, NEON-MS gains an average speedup of 1.25.

Keywords: Merge sort · Sorting network · SIMD · Parallel sort

1 Introduction

Sorting is one of the most extensively studied algorithms in computer science and plays a critical role in various computational applications [9], such as database retrieval [11], image processing [7], and visual computing [4]. Specially, with the advent of the big data era, there is a significantly increasing demand in improving sorting algorithms for data processing.

SIMD (Single Instruction, Multiple Data) is one of the most commonly used parallel technologies in modern processors, such as ARM NEON, Intel AVX, and RISC-V Vector Unit. In sorting, SIMD instruction sets are utilized to implement vectorized sorting networks, which handle small-scale data sorting. This is often referred to as in-register sort because all operations are performed on vector registers. Bramas *et al.* [3] explored an efficient quicksort variant on ARM CPUs with Scalable Vector Extensions (SVE-QS). It utilizes the bitonic sorting network for small partitions. Since its small-scale data sorting involves comparing and swapping data (Comparator) within the registers, the interaction across registers directly affects the sorting efficiency. Actually, SVE-QS only uses a few vector registers each time, so the in-register sort runs with the limited SIMD capabilities. Arman *et al.* [2] designed an in-memory merge sort framework, named Origami, which is optimized for scalar operations. Origami stacks all vector registers available to avoid register resource waste, but using too many registers will entail a complex sorting network and could induce unwelcome register-to-memory accesses. Yin *et al.* [12] proposed a highly efficient sorter based on the AVX-512 multi-core architecture. This sorter sorts small-scale data in registers. However, the symmetry merging networks are inefficient.

In terms of the fore-said discussions, on one hand, it is essential to emphasize register-to-memory access rather than register resource utilization. This is because the latter purely uses all the registers, while the former needs to carefully filter out a portion of registers, according to register functionalities. In this way, it is most likely to yield the optimal number of registers for our goal. On the other hand, the simpler the sorting or merging networks, the higher the overall sorting efficiency. Firstly, fewer registers are favorable to simplify the network structure. Secondly, when using the fixed size of registers, few comparators are attractive. Lastly, efficient realization of merging networks could benefit to thread-level parallelism.

- We recognize the optimal register number to reduce the register-to-memory access times.
- We introduce few-comparator column sort by using the best sorting network [5] with the asymmetric structure, which ensures the optimal sorting network with few comparators to be possible, in contrast to symmetric bitonic or even-odd sorting networks.
- We propose a new hybrid bitonic merging network that concurrently implements the serial merge and vectorized merge in an asymmetric fashion, allowing merge instructions to be highly interleaved in the pipeline.

This paper is organized as follows: Section 2 elucidates our optimization strategy and its implementation. Section 3 provides an analysis of the results, and Sect. 4 summarizes the paper’s key findings and contributions.

2 NEON-MS Algorithm

2.1 Overview

The overall flowchart of NEON-MS algorithm includes three core components as shown in Fig. 1. Before detailing them, we are ready to assign the input sequence of the length N to all available threads of the size T , with each thread being responsible for sorting its allocated subsequence of the length $\frac{N}{T}$. A threshold is set to the multiple of the SIMD width. When this threshold is achieved, we will employ the improved in-register sort to arrange small subsequences in some order. Then, we propose a hybrid bitonic merging network, using it as the core of vectorized merge [6] to accelerate the merging process. At last, such locally sorted subsequences will be globally merged. We entails a data partitioning strategy [10]. The primary optimization involves balancing the load so that each thread can allocate a comparable amount of workload. In every partitioning block, the thread executes the aforesaid vectorized merge.

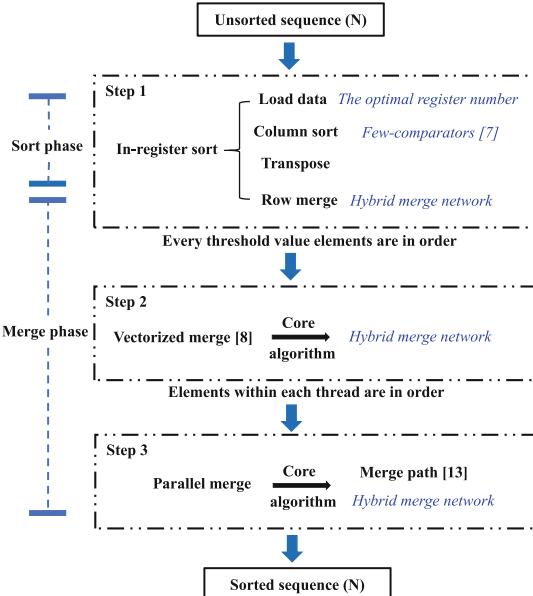


Fig. 1. The flowchart of NEON-MS algorithm, with three improvements highlighted in *blue italic*, i.e., the optimal number of the used registers, sorting networks with few comparators for column sort, and three types of merges with our hybrid merging network. (Color figure online)

As shown in Fig. 2, the in-register sort consists of four steps: load data, column sort, transpose and row merge. The data launch involves the number of the used registers, and the transpose tunes the register location to make

each row sorted. The most important step is column sort, because it not only induces the finest grained sorted subsequences but also offers the transposed yet in-order subsequences to row merge. If the row merge receives the disorder subsequences, the merge outcome is also disorder and meaningless. In light of this insight, it is not surprising that two mergers follows the in-register sort. Both mergers correspond to the vectorized merge and multi-thread parallel merge, respectively. Thus, the row merge of the in-register sort, the vectorized merge and the multi-thread parallel merge share the same merge spirit. In a nutshell, three factors, i.e., the number of the used vector registers, the column sorting network structure, and the merge implementation way, will decide the overall sorting efficiency on ARM NEON.

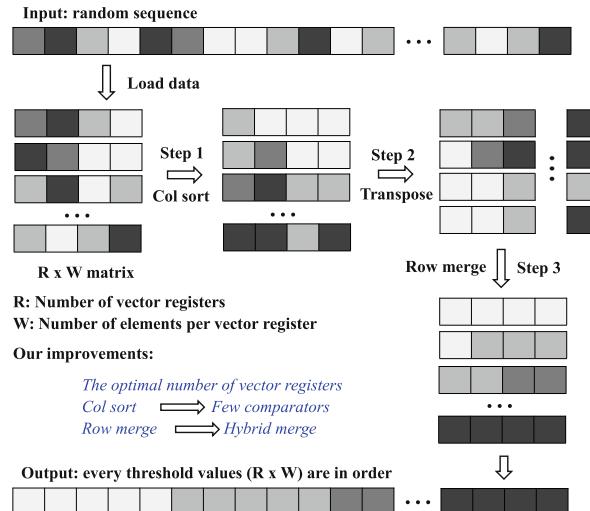


Fig. 2. The workflow of the in-register sort ($W = 4$), where each square represents a data item, with darker cells indicating larger values.

2.2 The Optimal Register Number

Since ARM NEON, different from other instruction sets like Intel AVX, gives the shorter vectorized registers, more flexible usage of such registers is a concern. Naturally, the first question is about how many registers used for the in-register sort is suited. Before that, we need to break a long-standing rule that constrain the number of the loaded elements to equal to the square of the number of elements per vector register, i.e., $W \times W$. If we follow this rule in the ARM NEON that has 32 128-bit vector registers, over 85% of the register hardware resources will be idle and wasted. Recent work [2] addresses this issue by stacking all available R vector registers with the size of W . The 32 vector registers in ARM NEON are all used there.

Once R is set and coupled with the fixed width W , the size of the sorting networks is decided. In the subsequent description, we take 32-bit integers as an example, meaning each vector register can store 4 elements, i.e., $W=4$. As one knows, the bigger the R , the more complex the sorting networks. For example, the comparators in a 32-element sorting network are three times those in a 16-element one. Besides, the register-to-memory access is always a big computation bottleneck. It is more efficient to store and access data on the registers than to do on the main memory. Thus, the proper value of R not only fits to the simple sorting network structure but also bypasses the register-to-memory access. In ARM NEON, the 16 vector registers meet this need, that is, $R = 16$. Our empirical studies in Table 2 show that this case has optimal efficiency on ARM NEON.

2.3 Few-Comparators Column Sort

Column sort involves sorting the same channel across multiple vector registers, which is a key utilization of vectorized properties. Most implementations use the sorting networks due to their simple welding structure. *Which sorting network structure for column sort are most efficient?* The efficiency can be evaluated by the number of the comparators used in a sorting network. Table 1 lists the relation between the number of the comparators and sorting networks of varying input size n . When $n = 2^i$, wherein i is a positive integer, the bitonic and odd-even sorting networks have the symmetric structures and the fixed comparators. Most current implementations belong to this sort but do have no room to further optimize the number of the comparators. In contrast, the asymmetric structure has fewer comparators than the symmetric structure, especially when the input size exceeds 8, because the upper bound of the comparators' number is less than its symmetric siblings. Besides, the odd input sizes are incompatible with the ready-to-use transpose operation, so the respective sorting networks are rarely used by the in-register sort. For larger networks, there exist the proven optimal lowest bounds for the number of comparators [8]. In ARM NEON, larger sorting networks ($n > 32$) is infeasible, because the total register number available is 32 at most.

Table 1. Number of comparators in different sorting networks of input size n .

n	Bitonic	Odd-even	Asymmetric Network
4	6	5	5
8	24	19	19
16	80	63	55–60
32	240	191	135–185

Since the optimal value of R is 16, namely, $n = 16$, according to Table 1, we have the chance to reduce the number of the comparators. Thus, we intro-

duce the 16-element best sorting network [5] for efficient multi-column sorting. This network has fewer comparators than the commonly-used symmetric sorting networks. This saves time.

Through our few-comparators column sort, the locally sorted sequences are distributed by column across different registers. This data layout being order in column must be recovered into the sorted layout in a register (in row) via a matrix transpose before the data are written back to memory. If $W < R$, we regard the $W \times W$ matrix transpose as the base matrix transpose like the atomic operation. Then transposing an asymmetric matrix $R \times W$ can be reduced into multiple small base matrix transpose. This involves adjusting the positions of the vector registers, with few overheads.

2.4 Hybrid Bitonic Merger

Motivation. After column sort and transpose, each ordered subsequences need to be further processed by multiple mergers to make the full sequence in order. As previously mentioned, three mergers including in-register row merge, vectorized merge and parallel merge share the same merge spirit. In ARM NEON, a proper candidate for the merge is the bitonic merging network, due to its simplicity and efficiency. Nevertheless, there are two existing ways to implement bitonic merging network. The first way straightly follows the predefined merging network (See Fig. 4). Due to the presence of the compare and swap operations in the comparator, the underlying assembly codes contain some branch jump instructions (See Fig. 3a). As one knows, once the branch jump prediction meets the errors, this leads to extra execution cycles. Seemingly, these branch jump instructions

(a) Branch jump instruction (**b.le**)

```

1 Function Comparator_v0(a, l, r):
2   | if (a[l] > a[r])           ...
3   |   std :: swap(a[l], a[r]); 28a8:6b04007f cmp w3, w4
4 end                                28ac:5400006d b.le 28b8
                                         ...

```

(b) Conditional swap instruction (**csel**)

```

1 Function Comparator_v1(a, l, r):
2   | bool flag = (a[l] > a[r]); ...
3   | int temp = a[l];           2878:6b03009f cmp w4, w3
4   | a[l] = r? a[r] : a[l]; 287c:1a83d084 csel w4, w4, w3, le
5   | a[r] = r? temp : a[r]; ...
6 end

```

Fig. 3. Two implementations of a comparator: the source code on the left, and the respective core assembly code on the right.

can be transformed into conditional swap instructions by using ternary operations (See Fig. 3b), but each time only one conditional instruction is allowed. In contrast, the second vectorization implementation can concurrently execute multiple comparators for efficient merging. Before each comparison operation, the data needs to be shuffled so that correct comparisons can be obtained. However, the shuffle operation between vector registers in ARM NEON is not sufficiently flexible because it requires additional type conversion operations, which incurs more instruction overheads. Thus, both of implementations have individual pros and cons. This hints that we could enjoy their strengths.

Hybrid Implementation. Recall that the bitonic merging network in itself has symmetric structures, as Fig. 4 shows. Then, a 32-element bitonic merging network has two 16-element symmetric parts, while each 16-element bitonic merging network has two 8-element symmetric parts, etc. Obviously, if necessary, such symmetrical parts can be implemented in any one of two previous ways and then run in parallel. By this insight, we propose a hybrid bitonic merging network that concurrently utilizes the above two implementation ways in the tail of the merging network, as the **black** and **blue** rectangles of Fig. 4 show. Such a hybrid implementation, with the help of the compiler, enable the assembly

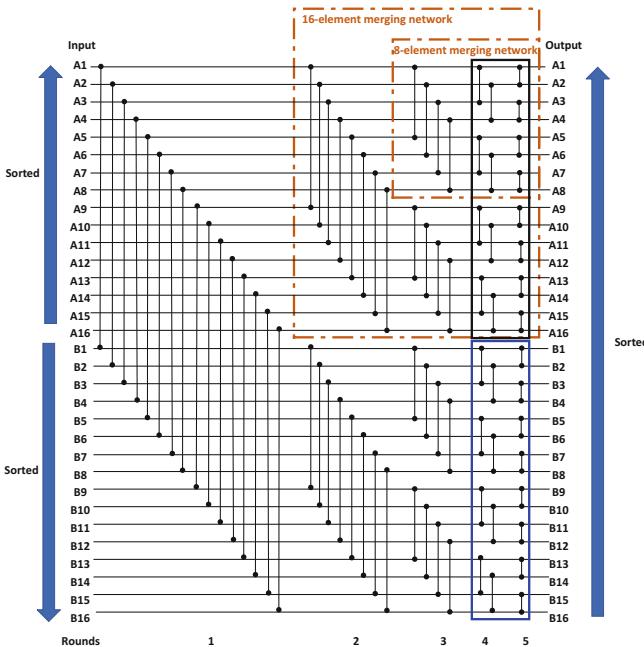


Fig. 4. A 32-element bitonic merging network. The **black** and **blue** rectangles represent data swapping in vector registers, characterized by their symmetry and independent operation. (Color figure online)

instructions of both the serial and the vectorized implementations to interleave with each other in the pipeline. This benefits to reducing waiting time on the conditional instructions and the times of data swapping among vector registers.

This hybrid spirit will serve for our three mergers which are used in the in-register sort, vectorized merge and multi-thread parallel merge, respectively.

3 Experimental Results

In this section, we evaluate NEON-MS on the FT2000+ processor, which operates at 2.3 GHz and features 64 cores. The processor architecture includes a shared 2M L2 cache for every group of 4 cores. Additionally, every 8 cores form a NUMA node, which is managed by DDR4 memory control. NEON-MS is implemented using the C++ language, taking advantage of ARM NEON features for optimization. In the parallel version, we employ the OpenMP standard. All implementations are compiled using GCC 9.3.0 with *-O3* level optimization.

First, we analyze the optimization effects of NEON-MS at different stages. As sorting small-scale data is very fast, we calculate average performance through multiple iterations. In our tests of in-register sort, we traversed 64K random integers and iterated 100 times. Second, we compare the performance of NEON-MS with that of boost::block_sort—one of the most efficient functions in C++ sorting implements [1], std::sort—a widely used sorting function in the C++ standard library in a single-thread context. Finally, we test the parallel performance of NEON-MS and boost::block_sort.

3.1 Localized Performance

Table 2. The running time (μ s) for sorting X elements in an $R \times 4$ matrix.

R	Every X elements are in order				
	$X=4$	$X=8$	$X=16$	$X=32$	$X=64$
4	38	105	186	-	-
8	-	49	112	179	-
16	-	-	76	134	203
16*	-	-	65	121	183
32	-	-	-	128	194

16* indicates the best 16-element sorting network.

Table 2 displays the running times required for sorting every X elements in in-register sort across various numbers of vector registers. It is evident that using 16* vector registers yields the best running time, primarily because this setup utilizes a best network designed for 16 elements, which requires fewer comparisons.

Moreover, it is observable that as the number of vector registers increases, the time required to sort the same quantity of X sorted elements decreases (excluding the 16*). This improvement occurs because more data can be processed simultaneously within each in-register sort. Although $R = 32$ demonstrates better performance than $R = 16$, due to its complexity, we do not consider it. Therefore, we conclude that the optimal number of vector registers is 16*.

Table 3. The relationship between merge lengths and merging speeds (elements/ μ s) across two merging methods.

Merge Length →	$2 \times 8 \rightarrow 16$	$2 \times 16 \rightarrow 32$	$2 \times 32 \rightarrow 64$
Vectorized Bitonic	873.81	1024	897.75
Hybrid Bitonic	1057.03	1092.27	840.21

Table 3 shows the merging speeds for different merge lengths across two merging methods. It can be observed that the merge speed of the hybrid bitonic method is faster than that of the vectorized bitonic method for merge lengths of 8 and 16. This advantage arises because our hybrid bitonic merging network concurrently utilizes two implementation ways in the tail of the merging network. This benefits to reducing waiting time on the conditional instructions and the times of data swapping among vector registers. For larger merge lengths, however, the merge speed of hybrid bitonic is slower than that of vectorized bitonic. This performance decrease is due to the fact that serial implementation generates temporary data that cannot be stored in the limited registers available. These data must be stored in memory, resulting in increased overhead.

3.2 Overall Performance

Figure 5 shows the performance of various sorting algorithms from 512K to 128M data sizes. This figure indicates that the overall performance of NEON-MS is better than other two methods. It achieves performance ranging from 23.5 to 70 ME/s, averaging 2.1 times faster than boost::block_sort, and 3.8 times faster than std::sort. In addition, NEON-MS is 1.25 times faster than boost::block_sort with 64 threads for large-scale data. This is attributed to the advantages of our parallel merge strategy, where each available thread remains active. For small data scales, the performance of NEON-MS with 64 threads is poorer than that of the boost::block_sort. This is because the creation of parallel workloads, thread synchronization, and extra computing tasks, such as the calculation of segmentation points in our implementation, occupy a major portion of the execution time. Meanwhile, boost::block_sort, while also a merging algorithm with multi-thread environments, features a small auxiliary memory (block_size multiplied by the number of threads). This configuration substantially minimizes the consumption of additional memory, leading to enhanced sorting efficiency.

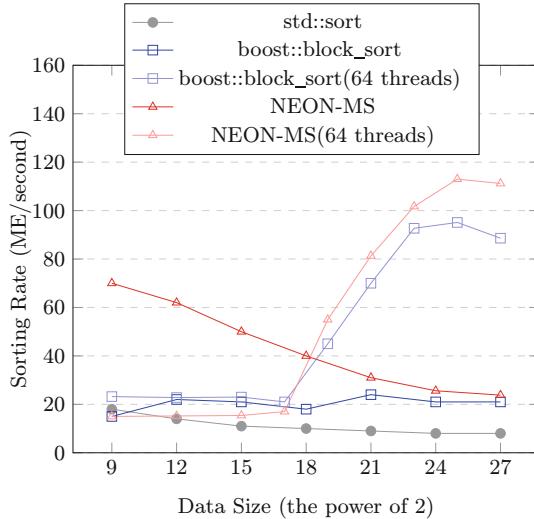


Fig. 5. Sorting Rate (ME/s: million elements per second) of different sorting methods for different data sizes.

4 Conclusion

This paper proposes a hybrid vectorized merge sort on ARM NEON architecture. This implementation utilizes the structural features of FT2000+ processors. In the sort phase, we use the optimal number of registers to achieve a simplified, efficient sort and introduce the best sorting network to further enhance sorting efficiency. During the overall merge phase, we propose a new hybrid bitonic merging network that allows merge instructions to be highly interleaved in the pipeline. The results show that single-thread NEON-MS performs, on average, 3.8 and 2.1 times faster than std::sort and boost::block_sort, respectively. Additionally, the multi-thread NEON-MS demonstrates a good performance enhancement.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (Grant Nos. 62032023, 42104078 and 6190241).

References

1. Boost.sort 3.-Parallel Algorithms (2021). <https://www.boost.org/doc/libs/develop/libs/sort/doc/html/sort/parallel.html>
2. Arman, A., Loguinov, D.: Origami: a high-performance mergesort framework. Proc. VLDB Endow. **15**, 259–271 (2021)
3. Bramas, B.: A fast vectorized sorting implementation based on the arm scalable vector extension (sve). PeerJ Comput. Sci. **7** (2021)
4. Christensen, P., Fong, J., Renderman, S., et al.: An advanced path-tracing architecture for movie rendering. ACM Trans. Graph. **37**(3) (2018)

5. Gamble, J.M.: Sorting network generator (2019). <http://pages.ripco.net/~jgamble/nw.html>
6. Inoue, H., Moriyama, T., Komatsu, H., et al.: Aa-sort: A new parallel sorting algorithm for multi-core SIMD processors. In: 16th International Conference on Parallel Architecture and Compilation Techniques, pp. 189–198 (2007)
7. Kobayashi, R., Kise, K.: A high performance FPGA-based sorting accelerator with a data compression mechanism. IEICE Trans. Inf. Syst. **100-D**, 1003–1015 (2017)
8. Marianczuk, J.: Engineering faster sorters for small sets of items. Softw. Pract. Exp. **51**, 1004–965 (2019)
9. Martin, W.A.: Sorting. ACM Comput. Surv. **3**(4), 147–174 (1971)
10. Odeh, S., Green, O., Mwassi, Z., et al.: Merge path - parallel merging made simple. In: IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 1611–1618 (2012)
11. Satish, N., Kim, C., Chhugani, J., et al.: Fast sort on CPUs and GPUs: a case for bandwidth oblivious SIMD sort. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (2010)
12. Yin, Z., Zhang, T., Müller, A., et al.: Efficient parallel sort on AVX-512-based multi-core and many-core architectures. In: IEEE International Conference on High Performance Computing and Communications, pp. 168–176 (2019)



An Encoder-Based Framework for Privacy-Preserving Machine Learning

Jiayun Wu^{1,2} , Wei Ren^{1,2,3} , Xianchao Zhang^{4,5} , and Xianghan Zheng^{6,7}

¹ School of Computer Science, China University of Geosciences,
Wuhan 430074, China

{jiayun.wu,weirencs}@cug.edu.cn

² State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR,
CASM, Beijing, China

³ Key Laboratory of Data Protection and Intelligent Management (Sichuan
University), Ministry of Education, Chengdu, China

⁴ Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province,
Jiaxing University, Jiaxing, China
 zhangxianchao@zjxu.edu.cn

⁵ Engineering Research Center of Intelligent Human Health Situation Awareness of
Zhejiang Province, Jiaxing University, Jiaxing, China

⁶ College of Computer and Big Data, Fuzhou University, Fuzhou, Fujian, China
 xianghan.zheng@fzu.edu.cn

⁷ School of Information and Intelligent Engineering, Sanya College, Hainan, China

Abstract. As a data-driven science, machine learning requires vast amounts of training data and computational resources. However, for highly privacy-sensitive data, it is crucial to protect the privacy of the data during both the training and utilization of machine learning models. In this paper, we propose a privacy-preserving machine learning approach using autoencoders and differential privacy mechanisms to safeguard data privacy while minimizing the impact on data availability. Specifically, we augment logistic regression and ResNet18 models with different architectures of autoencoders to perform data encryption? without compromising the machine learning tasks. Additionally, we employ differential privacy mechanisms to introduce gradient perturbations in the encoding part of the autoencoder, enhancing the algorithm's security and further protecting data privacy. We also design the cosine similarity between the encoded and original data as a metric for evaluating data privacy, considering model performance, privacy budget, and data privacy collectively to balance data availability and privacy. Extensive experiments conducted on MNIST, CIFAR-10, PathMNIST, and BloodMNIST datasets demonstrate that for simple logistic regression models handling easily classifiable datasets, employing simple autoencoder structures can enhance classification accuracy, with significant performance impact after adding differential privacy. For ResNet18, utilizing convolutional autoencoders for data encryption generally has minimal impact on model classification performance and can even improve accuracy in most cases. Adding differential

privacy has minor effects on model classification performance. Selecting appropriate model structures and privacy budgets for different usage scenarios can ensure both data availability and privacy.

Keywords: Privacy preservation · Differential Privacy · Autoencoder

1 Introduction

Machine learning (ML) is increasingly being adopted in various application domains. Typically, high-performing ML models rely on large amounts of training data and high-performance computing resources. However, this demand for large-scale data usage poses serious privacy issues, with the potential risk of leakage of highly sensitive information. For instance, in the medical field, a vast array of medical data can aid in training high-performing ML models, facilitating better medical diagnosis and analysis. Yet, medical data is highly sensitive, and regulatory, ethical, and moral requirements restrict its sharing, impacting applications such as medical image analysis and hindering the widespread deployment of ML models.

ML is a data-driven field, and the development of ML models heavily relies on extensive data training. To address privacy concerns while balancing data utilization, numerous privacy-enhancing techniques have been proposed. Privacy protection technologies based on differential privacy introduce artificial synthetic data or add noise perturbations to gradient parameters, weight parameters, objective functions, or model outputs during the model training process to ensure model or training data privacy [1]. Homomorphic encryption [2] and secure multiparty computation techniques [3] protect data privacy during computation processes through encryption methods. However, homomorphic encryption may lead to slower training processes, and for more complex models, the computational complexity increases exponentially with the increase in data volume. Secure multiparty computation incurs significant communication overhead as the number of participants increases. When using differential privacy techniques, adding excessive noise can impact model performance, necessitating a balance between data availability and security.

To achieve “usable but invisible” data, meaning data utilization without revealing the original data, this paper proposes a machine learning implementation solution utilizing autoencoders and differential privacy. By encoding data through the encoding part of an autoencoder before inputting it into the model, machine learning on encrypted data is achieved, while the autoencoder can also extract data features, in some cases enhancing the performance of the machine learning model. To further safeguard data privacy and prevent adversaries from inferring additional information from obtained encrypted data, the addition of a differential privacy mechanism within the autoencoder section is proposed. Controlling the impact of the differential privacy mechanism on the performance of the machine learning model achieves the goal of “usable but invisible” data, while mitigating membership inference attacks, feature inference attacks, and label inference attacks. The main contributions of this paper are summarized as follows:

- We propose to use autoencoder networks to conceal data to protect data privacy, yet data can still be used for further processes in deep learning.
- We add extra differential privacy to the autoencoder model by injecting noise during the model's gradient update process, thus defend against inferring the privacy content of individual data through gradient information.
- We aim to maintain the balance between data utility and data privacy by parameter selections, in extensive experiments on four typical datasets.

2 Preliminaries

2.1 Autoencoder

Autoencoder [4] is an artificial neural network consisting of an encoder and a decoder, used for learning effective encodings of unlabeled data. The encoder learns hidden feature encodings of input data, while the decoder reconstructs the original input data using the learned features. Besides being used for feature dimensionality reduction, autoencoders can act as feature extractors to extract more effective new features, which can be fed into supervised learning models.

The learning process of autoencoder can be unsupervised, requiring no labels or category information to generate data models. It extracts useful information from each instance, projects it into a feature vector through some transformation, then maps the representation back to the original feature space through a similar set of transformations, and evaluates the autoencoder based on the fidelity of the reconstruction. This feedback allows iterative parameter modification until convergence is achieved.

Autoencoders can be seen as a combination of encoding mapping f and decoding mapping g , where the encoding mapping f projects the input into a different feature space, and the decoding mapping performs the reverse operation. The primary goal of autoencoders is to recover as much information from the original input as possible, thus attempting to minimize the distance between the input and output.

$$\min_{\theta} \sum_{x \in X} d(x, g_{\theta}(f_{\theta}(x))) \quad (1)$$

The distance function d typically used in the loss function is either mean squared error or cross-entropy. For the former, data may not be normalized, and output units use unbounded activation functions; for the latter, each input and output variable is modeled to follow a Bernoulli distribution, so the data should be scaled to the $[0,1]$ interval, and output units can use sigmoid activation [5].

2.2 Differential Privacy

Differential privacy [6] is a privacy protection technique that introduces controlled noise during the data analysis process to balance the relationship between data utility and individual privacy. By introducing randomness into the dataset, differential privacy blurs and makes the differences between query results for different individuals ambiguous and unreliable. These random operations protect

privacy while still allowing useful statistical information to be extracted from the dataset.

The mechanisms for achieving differential privacy mainly include classic methods such as adding Laplace noise, exponential mechanisms, and function perturbation methods.

DEFINITION 1(local differential privacy). For a random algorithm M whose domain is $\mathbb{N}^{|X|}$, if for all $S \subseteq Range(M)$, and all databases that satisfies $\|x - y\|_1 \leq 1$ satisfies:

$$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta \quad (2)$$

M satisfies (ϵ, δ) -differential privacy. ϵ is the privacy budget of M . The smaller the ϵ value, the stronger the privacy protection and the lower the data availability [7].

3 Problem Formalization

3.1 Problem Definition

When it comes to sensitive data, disregarding inference attacks [8] and similar threats, data owners can only locally train the data to ensure its privacy as much as possible in plaintext scenarios. However, each data owner possesses extremely limited data, resulting in models with limited performance. To increase the sample size and obtain better-performing models, considering a special encryption method to protect the privacy of sensitive data is crucial. This method would maintain the data in encrypted form while ensuring that the features remain recoverable under encryption. Moreover, to ensure that the encrypted data can still be recognized by the classifier model. That is, the original plaintext data D undergoes a certain mapping p , resulting in the encoded data $D' = p(D)$, which can participate in machine learning tasks. The performance of the machine learning model trained on the original data D is denoted as R . The goal is to ensure that the performance R' of the model trained on the transformed data D' is as close to R as possible, or even greater than R . At the same time, it is essential to preserve the privacy of the data D' . Suitable evaluation metrics need to be selected to assess the privacy S of the data. The key is to select a mapping p that allows the data to meet the requirements while maximizing privacy preservation.

In summary, it is essential to consider a method that can encrypt data while not affecting subsequent data processing, while also balancing the privacy of sensitive data and model performance.

3.2 Design Goals

The aim of this paper is to design a model capable of implementing machine learning tasks with encrypted data while possessing certain privacy protection capabilities and relatively high model accuracy. Based on these objectives, the proposed solution should achieve the following goals:

- Privacy protection capability: It is essential to ensure the privacy of data during both training and application phases, thereby avoiding the possibility of data leakage.
- Machine learning under encrypted data: The model must ensure that machine learning classification tasks are performed under encrypted data conditions, meaning that the original data only appears in the initial process of encrypting the data.
- High model accuracy: It is imperative to ensure that the data's encrypted processing does not excessively impact the accuracy of machine learning model classification and to control the input of noise during the implementation of differential privacy.

4 Proposed Scheme

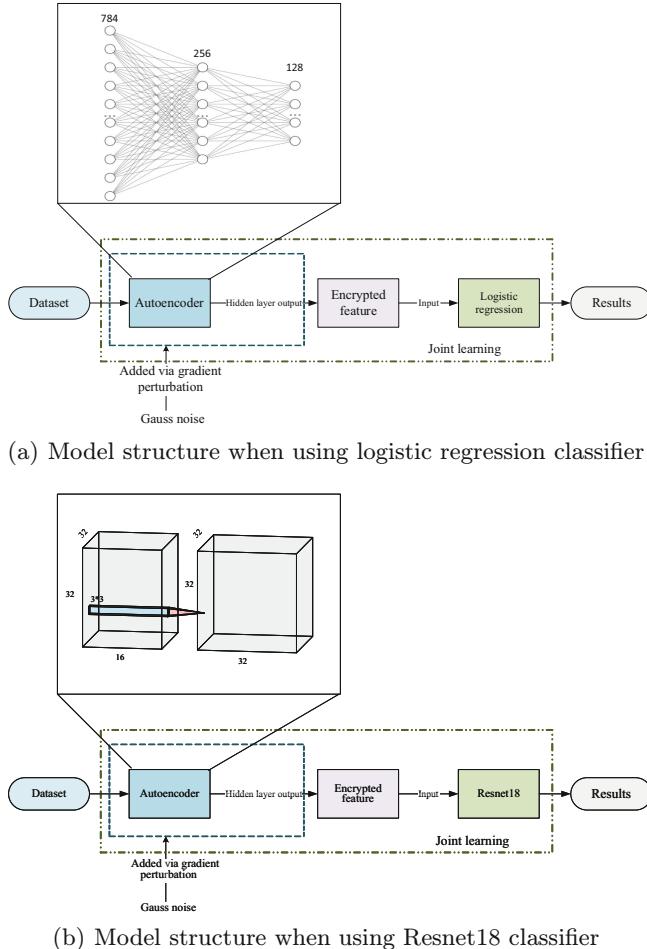
4.1 Technical Overview

In order to enable data to be used for machine learning under encryption, we choose the encoding part of the autoencoder to process the data into encrypted form. The encoding part of the autoencoder maps the input data to a low-dimensional encoding space, where the hidden layers contain valuable features of the data. By using the autoencoder to reduce the dimensions of the data, we simultaneously eliminate redundant information and extract the main features of the data. This not only encrypts the data but also allows us to directly use the extracted data features as input for the machine learning model. To further protect the privacy of the data during the encryption process, we add differential privacy to the encryption part, specifically the encoding part of the autoencoder. Since adding differential privacy to both the dataset and the encoding results significantly affects the performance of subsequent machine learning tasks, we choose to implement differential privacy by perturbing gradients with Gaussian noise. This approach allows us to perform machine learning tasks with encrypted data while minimizing the risk of data leakage due to member inference attacks during the encryption process.

Although this method ensures the privacy of the data, we also need to consider the data's availability. We need to find a balance between data privacy and model performance. Therefore, we quantify the privacy protection capability of the solution and balance the quantified indicators with machine learning model performance evaluation indicators to obtain a set of parameters that best balances data privacy and availability considerations.

4.2 Framework

This paper designs two autoencoder model structures for use with logistic regression and Resnet models as classifiers, as illustrated in Fig. 1. Both models follow the same overall process but differ in the autoencoder and classifier structures used. Data is first processed into encrypted form by the autoencoder, during

**Fig. 1.** Model flow chart

which Gaussian noise is added using gradient perturbation to achieve differential privacy. The output encrypted features from the autoencoder are then used as input for the classifier model to obtain classification results. The autoencoder shown in Fig. 1(a) consists of several fully connected layers, with the dimension changes of each layer illustrated using the example of the MNIST dataset. The autoencoder depicted in Fig. 1(b) comprises two convolutional layers, with the width, height, depth of the convolutional layers, and the size of the convolutional kernels also shown. The autoencoder and classifier are jointly trained to improve feature extraction and classification performance. The detailed description of the approach is as follows:

4.3 Differential Privacy Enhancement

Incorporating differential privacy into the encoding part of the autoencoder, rather than throughout the entire training process of the autoencoder and classifier, involves updating the parameters' gradients of the autoencoder model weights using the differentially private stochastic gradient descent algorithm. Gaussian noise is added to the gradients at each iteration. Utilizing the Opacus library to create a PrivacyEngine and attach it to the model to impart different levels of privacy with varying privacy budgets. To control the scale of the added noise, the gradient norm is clipped to be less than a certain value, ensuring that the overall sensitivity of the data is bounded.

By introducing Gaussian noise into the gradient during each iteration, the model parameters after each update satisfy differential privacy. For a dataset D , let the model parameters be θ , and the loss function be $L(\theta; D)$. In the gradient descent algorithm, after adding Gaussian noise, the gradient $g(\theta; D)$ is updated as follows:

$$\theta_{t+1} = \theta_t - \eta(g(\theta_t; D) + N(0, \delta^2 I)) \quad (3)$$

where η is the learning rate that controls the step size of each update, and $N(0, \delta^2 I)$ is the added Gaussian noise. For any two neighboring datasets D and D' , the difference in gradient updates is given by

$$g(\theta; D) + N(0, \delta^2 I) - (g(\theta; D') + N(0, \delta^2 I)) = \Delta g + N(0, \delta^2 I) - N(0, \delta^2 I) \quad (4)$$

Due to the properties of Gaussian noise and the definition of sensitivity, the output distribution satisfies (ϵ, δ) -DP. This mathematical proof ensures that the gradient update in each training iteration provides privacy guarantees, thereby protecting the privacy of the training data.

5 Results and Analysis

5.1 Experiment Settings

Datasets. In this experiment, three open-source datasets were utilized: MNIST, CIFAR-10, and the two subsets of MedMNIST [9]: BloodMNIST and PathMNIST. MedMNIST is a large-scale standardized biomedical image dataset similar to MNIST, comprising 12 2D datasets and 6 3D datasets. PathMNIST, one of the 2D datasets, consists of nine types of histopathological slices from colorectal cancer tissues, suitable for multi-class classification tasks; BloodMNIST consists of 8 classes of normal cells from individuals, also suitable for multi-class classification tasks. For each dataset, we evaluated the model's accuracy under different parameter settings. Specifically, we examined the impact of the privacy budget ϵ on the model and the privacy protection performance p of the data before and after adding differential privacy. For the logistic regression classifier model, the learning rate is set to 0.001 with a batch size of 100; for the ResNet18 classifier model, the learning rate is set to 0.001 with a batch size of 200. The privacy budget is set to 1.0/0.5/0.1.

Metrics. The evaluation metrics used in this experiment include commonly used metrics for classification tasks such as accuracy, precision, recall, and F1-score. In addition, the evaluation also incorporates the differential privacy parameters δ and ϵ , which are used to assess the privacy protection performance during the data encoding process. The privacy budget ϵ reflects the level of privacy protection provided by the algorithm, with lower values indicating higher levels of privacy protection. To facilitate the assessment of data utility, we compute the cosine similarity between the original data and the encoded data.

Model Structure. When using the logistic regression model as the classifier, the architecture of the autoencoder is adjusted based on different datasets. For the MNIST and CIFAR-10 datasets, the autoencoder consists of two fully connected layers, while for the PathMNIST and BloodMNIST datasets, the autoencoder consists of three fully connected layers. For all datasets, the autoencoder employs `relu()` as the activation function. When using resnet18 as the classifier, the autoencoder structure is fixed and consists of two convolutional layers. Despite the relatively simple model architectures chosen and the variation in autoencoder structures for different classifiers, the main purpose of the experiment is to validate the feasibility of the dense data machine learning approach. Therefore, theoretically, the experimental results can be extended to other more complex machine learning models.

Comparison Schemes. The experiment compares the classification performance of the model when only passing through the classifier, the classification performance of the model after adding the autoencoder encoding module, and the classification performance of the model after adding differential privacy during the autoencoder encoding process. Additionally, it compares the privacy protection performance of the model before and after adding differential privacy. Through these comparisons, a set of parameters that balance classification performance and privacy protection performance is selected as the optimal parameters.

5.2 Analysis

Analysis of Model Performance. The performance of the model is a crucial evaluation metric for assessing data usability and the rationality of the model structure. In this paper, we compare the performance of the model before and after adding the autoencoder module, as well as the performance of the differential privacy model before and after adding different privacy budget values. By contrasting the model performance under different settings, we aim to select a model that balances data privacy and usability. Figure 2 illustrates the accuracy of the classifier with and without the autoencoder module for different datasets. For logistics regression model, it can be observed that for relatively simple datasets like MNIST, the model's performance improves after adding a structurally simple autoencoder module. However, for the other three more

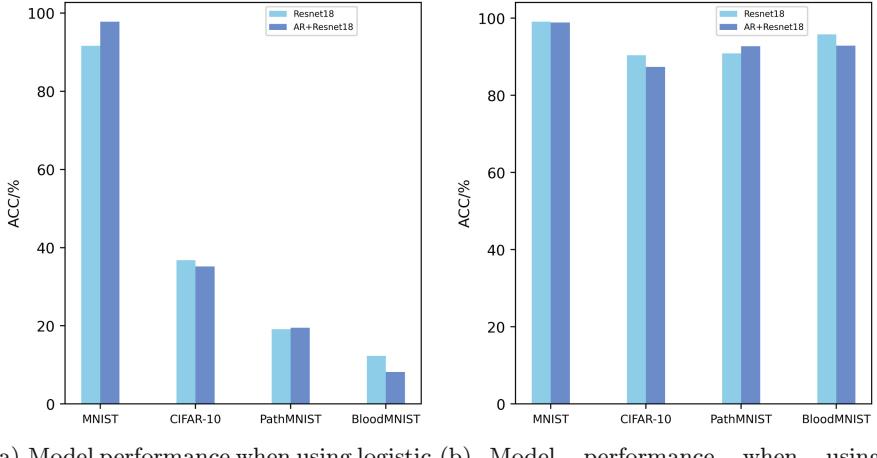


Fig. 2. Model performance with different model structure

complex datasets, especially BloodMNIST and PathMNIST, where the logistic regression classifier alone exhibits poor performance, the addition of a structurally simple autoencoder module does not lead to an improvement in model accuracy. For Resnet18, the model performs well on both simple and complex datasets. When an autoencoder module is added to the Resnet18, the accuracy does not differ significantly, with slight improvements observed in some cases.

In summary, by appropriately selecting the autoencoder’s structure and classifier based on different datasets and usage scenarios, we can ensure minimal impact on classifier performance or even enhance classifier performance while maintaining data privacy and algorithm security.

6 Conclusion

In this paper, we propose a privacy-preserving machine learning approach to ensure the privacy and usability of data during both the training and utilization processes of machine learning models. We utilize autoencoders as tools for processing data in a privacy-preserving manner, aiming to protect the privacy of the data while potentially enhancing the performance of machine learning models in certain scenarios. By incorporating the mechanism of differential privacy, we enhance the security of the algorithm and further bolster the privacy of the data. We employ autoencoder models of different structures in conjunction with classifier models and design evaluation metrics to measure data privacy. Experimental validation on four datasets demonstrates that employing this approach of learning from encoded data and selecting appropriate parameters can facilitate achieving the goal of “usable and invisible” for the data, thereby aiding in safeguarding data privacy and fostering increased possibilities for data sharing.

Acknowledgments. The research was financially supported by the State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR, CASM (No. 2023-04-04), the Key Laboratory of Data Protection and Intelligent Management, Ministry of Education, Sichuan University and also the Fundamental Research Funds for the Central Universities (No. SCU2023D008), Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province (No. MEDC202305), and The Innovation Platform for Academician of Hainan Province (No. YSPTZX202145).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Zhao, J., Chen, Y., Zhang, W.: Differential privacy preservation in deep learning: challenges, opportunities and solutions. *IEEE Access* **7**, 48901–48911 (2019)
2. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. *ACM Comput. Surv. (CSUR)* **51**(4), 1–35 (2018)
3. Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., van der Maaten, L.: Crypten: secure multi-party computation meets machine learning. *Adv. Neural. Inf. Process. Syst.* **34**, 4961–4973 (2021)
4. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
5. Chartre, D., Chartre, F., del Jesus, M.J., Herrera, F.: An analysis on the use of autoencoders for representation learning: fundamentals, learning task case studies, explainability and challenges. *Neurocomputing* **404**, 93–107 (2020)
6. Dwork, C.: Differential privacy. In: International Colloquium on Automata, Languages, and Programming, pp. 1–12. Springer (2006)
7. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **9**(3–4), 211–407 (2014)
8. Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: a survey. *ACM Comput. Surv. (CSUR)* **54**(11s), 1–37 (2022)
9. Yang, J., et al.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Sci. Data* **10**(1), 41 (2023)



Load Balancing Optimizations for Distributed GMRES Algorithm

Yuxiang Zhang¹ , Shuaizhe Guo² , Jianhua Gao¹ , Weixing Ji¹ ,
and Yizhuo Wang²

¹ Beijing Normal University, Beijing, China
`zyx@mail.bnu.edu.cn, {gaojh,jwx}@bnu.edu.cn`
² Beijing Institute of Technology, Beijing, China
`{chakra_guo,frankwyz}@bit.edu.cn`

Abstract. The Generalized Minimal Residual Method (GMRES) is one of the most important iterative algorithms for solving large-scale sparse linear systems, which are widely used in fields such as computational fluid dynamics and computational electromagnetics. As the scale of problems increases, multi-node distributed systems become one of the most popular running environments. Communication efficiency is usually the primary performance bottleneck for distributed GMRES. Traditional work reduces communication load mainly by balancing the computation load, while the balance of communication load is also important. This paper proposes a rule-based algorithm and a reinforcement learning (RL)-based algorithm to balance the communication load. By optimizing the partitioning of sparse matrices using the rule-based algorithm, the balance of both computation and communication loads among devices is improved. Experimental results show that the speedup can reach up to 1.34x. Moreover, RL-based algorithm improves the efficiency of the iterative algorithm by optimizing the task allocation of the partitioned sub-matrices. Experimental results present that the speedup can reach up to 1.30x.

Keywords: GMRES · Sparse matrix · Load balancing · Reinforcement learning

1 Introduction

One of the most important steps in many scientific and engineering computational problems is solving linear systems of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$. The Generalized Minimum Residual Method (GMRES) [1] is one of the most popular iterative methods for solving sparse linear systems. It is widely applied in fields such as electromagnetic simulation [2], computational fluid dynamics [3], and circuit simulation [4].

Y. Zhang and S. Guo—Contributed equally to this work.

This work is supported jointly by China Postdoctoral Science Foundation under grants No. 2024M750223 and GZC20230261.

Given that large sparse matrices can occupy tens to hundreds of gigabytes of storage space, distributing the GMRES algorithm across multiple nodes becomes essential. Partitioning sparse matrices allows for distributing computation across multiple devices, but two main bottlenecks arise: communication load and computation load. Computation load refers to the assigned SpMV calculation tasks per device, and communication load refers to the required data communication tasks per device before running the SpMV calculation. Frequent inter-device communication often creates a significant performance bottleneck. Existing work [5] reduces communication load significantly by only transmitting useful vector elements, but imbalanced loads persist because the irregular distribution of non-zero elements and inappropriate partitioning positions. Therefore, this paper proposes two algorithms to achieve more balanced loads.

Our main contributions include three parts. First, we propose a rule-based algorithm that adjusts partitioning positions based on the even division of rows, thereby achieving more balanced communication loads among devices. Second, this paper introduces a reinforcement learning (RL)-based method to balance workloads during iterations. Third, to evaluate the effectiveness of proposed methods, we conducted extensive experiments using various sparse matrices with different non-zero element distributions. We compared our rule-based and RL-based load-balancing optimization methods with traditional techniques. The results show that the rule-based method achieved an average speedup of 1.17x, with the highest speedup of 1.34x, and the RL-based method achieved an average speedup of 1.12x, with the highest speedup of 1.30x.

2 Related Work

Since the introduction of the GMRES algorithm, numerous optimized variants have emerged to address evolving hardware and application demands. Walker [12] proposed the simplified GMRES (SGMRES), Vorst and Vuik [6] introduced GMRESR, and Essai [7] developed the weighted GMRES algorithm (WGMRES), as well as the mixed version WSGMRES of SGMRES and WGMRES algorithm [8]. To manage increasing computational and storage demands during iterations, the restarting technique [9] was introduced. Besides, Researchers have increasingly focused on implementing GMRES on heterogeneous systems. Couturier et al. [10] demonstrated GMRES advantages on a CPU+GPU system, and others optimized GMRES for specific applications, enhancing performance through preconditioners [11], SpMV optimizations [12], and improved data transmission [13]. De Vries et al. [14] parallelized GMRES across multi-core and multi-GPU systems, analyzing their performance in practical problems.

The use of distributed implementations, which require inter-process communication across nodes, is becoming increasingly prevalent. Khodja et al. [5] optimized communication by compressing data and employing hypergraph partitioning. Yamazaki et al. [15] reduced communication frequency through communication avoiding GMRES, while He et al. [16] improved SpMV computation with segSpMV. Other optimizations include orthogonalisation enhancements [17]

and asynchronous global reductions [18], which overlap computation with communication.

Despite these advances, most research focuses on reducing communication overhead globally or locally minimizing data transmission, with insufficient emphasis on balancing communication load across devices.

3 Methodology

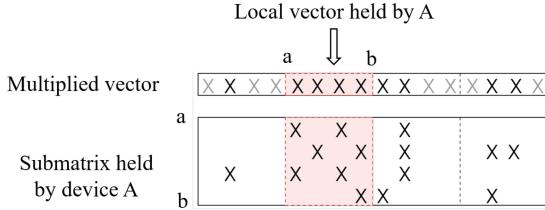
This paper focuses on balancing computation and communication load for the distributed GMRES. We first investigate the causes of communication load imbalance, which can be worse when compressed optimization reduces communicated data. To tackle this issue, this paper proposes two optimization strategies. First, we designed a rule-based low-overhead load balancing algorithm that adjusts the partitioning of the sparse matrix to balance computation and communication load. Moreover, we introduced a RL-based load balancing method. It includes training a neural network to find relatively balanced partitions.

3.1 Load Balancing Analysis

In distributed GMRES, sparse matrices need to be partitioned. Sparse matrix partitioning can be categorized into one-dimensional (1D) and two-dimensional (2D) partitioning. Because 2D partitioning increases communication complexity and load, 1D partitioning is used in this paper.

Dense vectors in matrix operations must also be partitioned. As shown in Fig. 1, if device A holds the submatrix spanning rows a to b , it only has the corresponding vector elements from a to b before SpMV computation. The column indices of the submatrix's non-zero elements determine which vector elements are needed from other devices. Thus, inter-device communication is required to obtain these elements. In a distributed system with p devices, each device communicates with $p - 1$ other devices, which often asynchronously use methods like *MPI_Isend*. However, SpMV operations can only begin after all communications are finished. If communication load is unbalanced, SpMV must wait for all data communication, leading to significant performance degradation.

Equal row partitioning typically results in balanced communication load, assuming no compression. However, compressed transmission or irregular row distribution may lead to significant communication imbalances. Since partitioning determines the communication load, it can be optimized during preprocessing. Furthermore, as the computation load also plays a critical role in overall performance, the load balancing algorithm must also consider the computation load.

**Fig. 1.** SpMV on device *A*.

3.2 Rule-Based Load Balancing Method

In Sect. 3.1, the load balancing algorithm assumes a fully asynchronous environment to balance communication and computation load by adjusting sparse matrix partitioning across devices. Exhaustively enumerating all possible partition configurations is infeasible due to the impracticality of predicting execution time without running the actual algorithm and the complexity of approximately $nRow^p$ if the sparse matrix has $nRow$ rows and there are p devices.

To balance loads, the algorithm reallocates submatrix from the device with the highest communication load to others. The communication load, represented in two dimensions, must be reduced to a single dimension by summing the total data received by each device. As shown in Fig. 1, reducing communication load is not directly proportional to the number of rows removed. The algorithm ensures that removing rows does not increase the device's communication load.

Algorithm 1. Rule-based load balancing algorithm.

Input: relative standard deviation evp , the array of total communication load for each device CL , the total number of processes $nRanks$, maximum number of partition adjustments $maxCnt$, preset threshold $evpLimit$

Output: none

```

1: cnt = 0
2: while (cnt < maxCnt) && (evp > evpLimit) do
3:   maxcr = 0
4:   mcr = CL[0]
5:   for i = 1 to nRanks do
6:     if CL[i] > mcr then
7:       maxcr = i
8:       mcr = CL[i]
9:     end if
10:   end for
11:   execute partition adjustment algorithm, return ret and evp
12:   if ret == True then
13:     cnt ++
14:   end if
15: end while

```

The load balance is measured using the relative standard deviation evp of communication load, and if it exceeds a preset threshold $evpLimit$, the load balancing process begins. We use Eq. (1) to calculate the relative standard deviation evp of the data received before SpMV computation on each device. Algorithm 1 identifies the device with the highest communication load and reallocates its submatrices to neighboring devices, continuing until the load is balanced or a maximum number of iterations is reached.

$$evp = \frac{\sqrt{\sum_0^{nRanks} (CL[i] - avg)^2}}{avg} \quad (1)$$

When performing partition adjustment, there are two cases: boundary and non-boundary devices. Here we discuss the case of device 0, while the others are similar. It calculates the load ratio δ between $CL[1]$ and $CL[0]$. The partition is adjusted if $\delta \leq 1.0 - indRatio$ is satisfied. The number of rows moved from device 0 to device 1 is calculated based on the proportion of rows held by device 0. For instance, if device 0 holds 100 rows and $\delta = 0.8$, then $dt = 100 \times \frac{0.2}{adjustFact}$, where $adjustFact$ is a scaling factor. This operation reallocates a portion from the end of submatrix 0 to submatrix 1, ensuring that communication load of device 0 (initially the highest) does not increase, while the load of device 1 increases. After each adjustment, the relative standard deviation evp is recalculated to check if further rebalancing is needed. Single adjustment may not significantly improve balance, thus requiring multiple iterations.

3.3 RL-Based Load Balancing Method

This section focuses on optimizing load balancing for specific sparse matrices and distributed systems, where the distribution of non-zero elements, GPUs, and nodes is fixed.

Although theoretically optimal load balancing could be achieved by enumerating all possible partitions. Fine-grained adjustments, such as one row at a time, have small effect on execution time and are easily affected by runtime fluctuations. Therefore, a relatively coarse-grained adjustment is used, such as 10% rows at once, which reduces the search space. The sparse matrix can be divided into more than p partitions, and the algorithm can select execution devices for each partition. Predicting partition performance is difficult, so reinforcement learning (RL) is used to balance loads. RL agents interact with their environment, refining strategies to maximize performance. Recent work, like Mirhoseini et al. [19], used deep RL to optimize operator placement across devices. Similarly, in this context, RL adjusts submatrix device configurations in distributed GMRES. Execution time is fed back to the RL model, which iteratively optimizes partitioning until a relatively optimal partition is found.

The RL Network Design. The deep RL network is inspired by Mirhoseini et al. [19], and uses a seq2seq model with an encoder-decoder structure (Fig. 2). The

encoder processes submatrix features (e.g., rows, non-zero elements) and passes the hidden state to the decoder. The decoder generates device allocations for submatrices sequentially by selecting devices with the *argmax* function. The RL network is trained by pre-partitioning sparse matrix, feeding submatrix features into network, and generating allocation strategies. The RL network runs on CPU, and allocation strategies are executed by the distributed system. Execution time is used as rewards to refine the network through backpropagation.

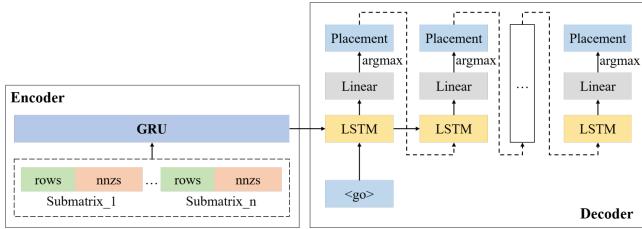


Fig. 2. Reinforcement learning network structure.

4 Experimental Evaluation

4.1 Experiment Setup

Distributed GMRES is executed on a cluster including 4 nodes, each running on 64-bit CentOS 7.5 operating system and equipped with two Intel Xeon E5-2640 v4 processors with 2.40 GHz clock frequency and 10 physical cores. Each node is configured with four Tesla V100 GPUs, and each GPU has 5,120 CUDA cores and 16 GB device memory. The CUDA toolkit 11.8 and NCCL 2.13.4 are used. The cluster adopts a common CPU+GPU+NIC configuration, where all 4 GPUs share an Intel Omni-Path NIC for inter-node communication.

Regarding the sparse matrix test set, this paper selects 12 large sparse matrices from the SuiteSparse Matrix Collection [20], originating from different domains. The basic information of these sparse matrices is summarized in Table 1. Their non-zero elements are in tens of millions, with varying numbers of rows and columns ranging from hundreds of thousands to tens of millions. When the number of non-zero elements is similar, larger numbers of rows and columns result in a higher proportion of communication overhead in distributed GMRES.

4.2 Evaluation for Rule-Based Load Balancing Optimization

To investigate the effect of using 4 GPUs on sparse matrix partitioning, experiments are conducted in two configurations: 2-node and 4-node setups. And our

Table 1. Tested sparse matrices

Matrix	Rows/Cols	NNZ
adaptive	6,815,744	27,248,640
AS365	3,799,275	22,736,152
cage14	1,505,785	27,130,349
F1	343,791	26,837,113
nlpkkt80	1,062,400	28,704,672
GAP-road	23,947,347	57,708,624
hugebubbles-00020	21,198,119	63,580,358
delaunay_n23	8,388,608	50,331,568
ljournal-2008	5,363,260	79,023,142
asia_osm	11,950,757	25,423,206
hugebubbles-00010	19,458,087	58,359,528
road_central	14,081,816	33,866,826

analysis of communication load imbalance across the GPUs revealed the relative standard deviation ranged from 6% to 105%, with most matrices exceeding 20%.

In load balancing Algorithm 1, key parameters are set empirically: $evpLimit = 0.20$, meaning partitions with a relative communication load imbalance above 20% are required to be adjusted; $maxCnt = 8$, setting a limit of 8 repartitions; when performing partition adjustment, $indRatio = 0.10$, meaning repartition is used when communication load difference between devices exceeds 10%; and $adjustFact = 4.0$, meaning 25% of the communication load difference is multiplied by row count of adjacent submatrix to determine the partition position.

After applying the load balancing algorithm, five out of seven matrices show reduced communication load, as shown in Fig. 3. Notable speedups are observed for *adaptive* and *delaunay_n23* matrices, with the highest speedup of 1.25x and an average of 1.06x. Despite increased communication load, execution time decreases due to effective overlap of communication and computation. Moreover,

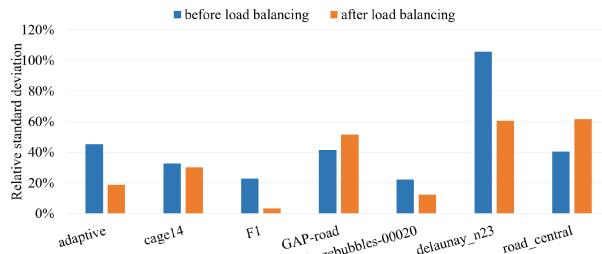


Fig. 3. The comparison of the relative standard deviation of communication load before and after load balancing with $maxCnt = 8$ and 4 GPUs.

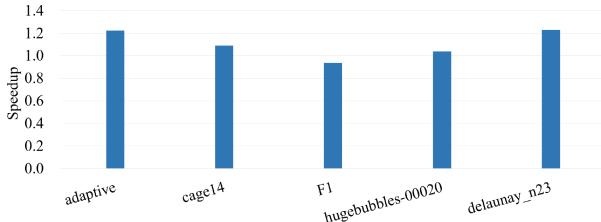


Fig. 4. The speedup of the load balancing with $maxCnt = 8$ under 4 nodes compared to the baseline without load balancing.

the experiment is executed on four nodes and found to be more significant, with the highest speedup of 1.23x and an average of 1.10x, as shown in Fig. 4. This indicates the significant impact of network bandwidth on performance.

Given the small computation overhead of the proposed load balancing algorithm, further experiments with adjusted parameters are conducted under the 4-node configuration. Increasing $maxCnt$ to 30 and setting $adjustFact$ to 6.0 lead to finer-grained adjustments and improved performance across all five tested matrices, achieving the highest speedup of 1.34x and an average of 1.17x.

4.3 Evaluation for RL-Based Load Balancing Optimization

The experiments are conducted on the same cluster platform as before, with additional software environments including Python 3.11 and PyTorch 2.1 for training RL networks. The RL network's hidden layers for both encoder and decoder are set to $(1, 2 \times p)$, where p is the number of pre-partitions of the sparse matrix across n GPUs. The network is optimized using the Adam optimizer with a high learning rate 0.3 due to limited training data. The reward function is defined as the execution time of the distributed algorithm. Pre-partitioning is recommended to be $p > 2n$, ensuring more submatrices than GPU devices.

In this section of the experiment, two main benchmarks are compared: the default allocation, where submatrices are sequentially assigned to GPUs, and a random allocation method, where allocations are randomly generated, tested, and selected based on the shortest execution time. For both methods, if any GPU received no submatrices, the allocation is discarded and regenerated. The RL-based optimization is evaluated against these benchmarks.

The experiment still involves 8 sparse matrices, using four nodes and four GPUs. For RL-based optimization, the maximum number of partition adjustments is set to 100, with the same limit for random generation. The results, shown in Fig. 5, indicate that the RL-based method outperforms the default allocation with an average speedup of 1.12x and a maximum of 1.30x. Compared to the random allocation, RL optimization achieves an average speedup of 1.06x and a maximum of 1.13x.

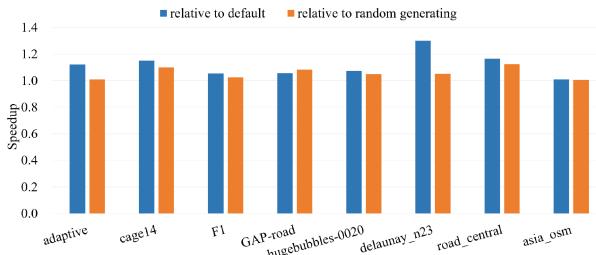


Fig. 5. The speedup of optimal strategies chosen by the reinforcement learning network compared to the default allocation method.

5 Conclusion

This paper first thoroughly analyzes the causes of load imbalance in distributed GMRES algorithms optimized with compressed transmission and proposes several feasible solutions. Building upon this analysis, the paper proposes a rule-based load balancing algorithm from the perspective of practical application. This algorithm fine-tunes the partitioning of sparse matrices multiple times in the preprocessing stage to optimize load balance during the iteration process. Experimental results demonstrate that this method achieves an average speedup ratio of 1.10x and up to a 34% performance improvement in test data. Furthermore, for scenarios where the distribution of non-zero elements and distributed platforms are fixed, the paper proposes a RL-based load balancing optimization algorithm. Experimental results prove that this method achieves an average speedup of 1.12x, with a performance improvement of approximately 10% compared to the default allocation strategy.

References

1. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *Soc. Ind. Appl. Math.* 856–869 (1986). <https://doi.org/10.1137/0907058>
2. Yang, G., Du, Y.: A robust preconditioned GMRES method for electromagnetic scattering from dielectric rough surfaces. *IEEE Trans. Geosci. Remote Sens.* 3396–3408 (2012). <https://doi.org/10.1109/TGRS.2012.2184291>
3. Soulaimani, A., Salah, N.B., Saad, Y.: Enhanced GMRES acceleration techniques for some CFD problems. *Int. J. Comput. Fluid Dyn.* 1–20 (2002). <https://doi.org/10.1080/10618560290003991>
4. Li, Z., Shi, C.: An efficiently preconditioned GMRES method for fast parasitic-sensitive deep-submicron VLSI circuit simulation. *Des. Autom. Test* 752–757 (2005). <https://doi.org/10.1109/DATC.2005.57>
5. Ziane Khodja, L., Couturier, R., Giersch, A., Bahi, J.M.: Parallel sparse linear solver with GMRES method using minimization techniques of communications for GPU clusters. *J. Supercomput.* 200–224 (2014). <https://doi.org/10.1007/s11227-014-1143-8>

6. Van Der Vorst, H.A., Vuik, C.: GMRESR: a family of nested GMRES methods. *Numerical Linear Algebra Appl.* 369–386 (1994). <https://doi.org/10.1002/nla.1680010404>
7. Essai, A.: Weighted FOM and GMRES for solving nonsymmetric linear systems. *Numer. Algorithms* 277–292 (1998). <https://doi.org/10.1023/A:1019177600806>
8. Yang, S., Lu, L.: A weighted simpler GMRES algorithm. *J. Xiamen Univ.* 484–488 (2008). (in Chinese)
9. Morgan, R.B.: A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.* 1154–1171 (1995). <https://doi.org/10.1137/S0895479893253975>
10. Couturier, R., Domas, S.: Sparse systems solving on GPUs with GMRES. *J. Supercomput.* 1504–1516 (2012). <https://doi.org/10.1007/s11227-011-0562-z>
11. Liu, X., Wang, H., Tan, S.: Parallel power grid analysis using preconditioned GMRES solver on CPU-GPU platforms. In: IEEE/ACM International Conference on Computer-Aided Design, pp. 561–568 (2013). <https://doi.org/10.1109/ICCAD.2013.6691171>
12. Ma, W., Hu, Y., Yuan, W., Liu, X.: Developing a multi-GPU-enabled preconditioned GMRES with inexact triangular solves for block sparse matrices. *Math. Probl. Eng.* 1–17 (2021). <https://doi.org/10.1155/2021/6804723>
13. Gao, J., Wu, K., Wang, Y., Qi, P., He, G.: GPU-accelerated preconditioned GMRES method for two-dimensional Maxwell's equations. *Int. J. Comput. Math.* 2122–2144 (2017). <https://doi.org/10.1080/00207160.2017.1280156>
14. DeVries, B., Iannelli, J., Trefftz, C., O'Hearn, K.A., Wolffe, G.: Parallel implementations of FGMRES for solving large, sparse non-symmetric linear systems. *Procedia Comput. Sci.* 491–500 (2013). <https://doi.org/10.1016/j.procs.2013.05.213>
15. Yamazaki, I., Rajamanickam, S., Boman, E.G., Hoemmen, M., Heroux, M.A., Tomov, S.: Domain decomposition preconditioners for communication-avoiding Krylov methods on a hybrid CPU/GPU cluster. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 933–944 (2014). <https://doi.org/10.1109/SC.2014.81>
16. He, K., Tan, S.X.D., Zhao, H., Liu, X.X., Wang, H., Shi, G.: Parallel GMRES solver for fast analysis of large linear dynamic systems on GPU platforms. *Integration* 10–22 (2016). <https://doi.org/10.1016/j.vlsi.2015.07.005>
17. Aquilanti, P.-Y., Petiton, S., Calandra, H.: Parallel GMRES incomplete orthogonalization auto-tuning. *Procedia Comput. Sci.* 2246–2256 (2011). <https://doi.org/10.1016/j.procs.2011.04.245>
18. Ghysels, P., Ashby, T.J., Meerbergen, K., Vanroose, W.: Hiding global communication latency in the GMRES algorithm on massively parallel machines. *SIAM J. Sci. Comput.* 48–71 (2013). <https://doi.org/10.1137/12086563X>
19. Mirhoseini, A., et al.: Device Placement Optimization with Reinforcement Learning. *arXiv* (2017). <https://doi.org/10.48550/arXiv.1706.04972>
20. Davis, T.A., Hu, Y.: The university of Florida sparse matrix collection. *ACM Trans. Math. Softw.* 1–25 (2011). <https://doi.org/10.1145/2049662.2049663>



ZKCross: An Efficient and Reliable Cross-Chain Authentication Scheme Based on Lightweight Attribute-Based Zero-Knowledge Proof

Yuwei Xu^{1,2,3(✉)}, Hailang Cai^{1,3}, Jialuo Chen^{1,3}, Qiao Xiang⁴, Jingdong Xu⁵, and Guang Cheng^{1,2}

¹ School of Cyber Science and Engineering, Southeast University, Nanjing, China
xuyw@seu.edu.cn

² Purple Mountain Laboratories for Network and Communication Security, Nanjing, China

³ Engineering Research Center of Blockchain Application, Supervision and Management, Nanjing, China

⁴ School of Informatics, Xiamen University, Xiamen, China

⁵ College of Computer Science, Nankai University, Tianjin, China

Abstract. The consortium blockchain finds wide application in finance and logistics. However, these independent networks create information silos. Cross-chain authentication aims to bridge these gaps. It verifies qualifications across different blockchains, yet it can risk leaking sensitive data. The main methods include Attribute-Based Signatures (ABS) and Attribute-Based Zero-Knowledge Proof (ABZKP). ABZKP better handles key management, general applicability, and attribute verification than ABS. However, the current ABZKP suffers from large proof files and slow verification. Additionally, many schemes can't handle sets of multiple attributes and may introduce centralization. Overall, the current schemes based on zero-knowledge proof still struggle to strike a balance between security and efficiency. We propose a reliable cross-chain authentication scheme. Our lightweight zero-knowledge proof algorithm enhances verification speed and reduces file size. We also introduce parallel verification to address security concerns. Security analysis and experiments show that ZKCross when properly configured, achieves near-perfect accuracy and outperforms others in size and speed. Thus, ZKCross stands as an efficient and dependable cross-chain authentication scheme.

Keywords: Blockchain · Cross-chain authentication · Zero-knowledge

1 Introduction

In recent years, consortium blockchains have been widely applied across various industries such as finance [1], logistics [2], and healthcare [3] due to their distributed deployment, immutability of data, and traceability of transactions.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
T. Zhu et al. (Eds.): ICA3PP 2024, LNCS 15256, pp. 57–77, 2025.

https://doi.org/10.1007/978-981-96-1551-3_6

However, the lack of effective connectivity mechanisms between these independent blockchain networks has led to issues of information silos and isolated value systems. Cross-chain authentication technology aims to break down the barriers between different blockchain networks. Through cross-chain authentication, qualifications can be verified across disparate blockchain networks. For example, in commercial and governmental contexts, service providers can authenticate the internal data of service requesters to determine the legitimacy of their identity and decide whether to provide the corresponding services. Therefore, the research and application of cross-chain authentication technology hold significant practical importance and offer vast development prospects.

However, during the cross-chain authentication process, the transmission of sensitive information or data may lead to leakage issues. In recent years, researchers have proposed various solutions to address privacy leakage concerns associated with cross-chain authentication. Among these, Attribute-Based Signature (ABS) and Attribute-Based Zero-Knowledge Proof (ABZKP) are the two most popular approaches.

ABS allows a business chain to sign a message based on a set of attributes it possesses. Service providers can verify the validity of the signature to determine whether the business chain holds a specific combination of attributes, thereby deciding whether to provide the service without needing to know the exact attribute information of the business chain. However, ABS has several shortcomings. First, ABS tends to verify the ownership of attributes, for instance, a service provider can ascertain whether the business chain possesses certain attributes but cannot verify the specific values of these attributes. Additionally, ABS requires different signing and verification key pairs for different sets of attributes, which limits its general applicability in cross-chain authentication [4]. Finally, managing these private keys necessitates a trusted authority and a secure and efficient management mechanism. The leakage or improper management of these keys can compromise the system's security [5].

Compared to ABS, ABZKP effectively addresses the aforementioned issues. First, ABZKP can verify the content of attributes without revealing any secret information, such as determining whether an attribute value falls within a certain range [6]. Additionally, ABZKP offers general applicability, the same proof verification algorithm can be applied to different sets of attributes without the need to redesign the algorithm. Finally, many schemes based on ABZKP do not generate a large number of specific signing keys for attribute sets, which reduces the reliance on a trusted third party to some extent.

However, existing research still faces the following problems. Some interactive zero-knowledge proof schemes require multiple rounds of interaction to complete the proof and verification process, necessitating multiple exchanges between communication nodes and posing a risk of replay attacks [7,8]. To address this, many studies have focused on non-interactive zero-knowledge proof schemes. Nevertheless, generating a zero-knowledge proof typically requires extensive computation, involving complex mathematical calculations and cryptographic operations, which can slow down the proof generation process and impact the speed of

cross-chain authentication [9]. Additionally, some zero-knowledge proof schemes generate large proof files, which entail substantial data transmission, particularly problematic in bandwidth-constrained network environments [10]. Consequently, some work has introduced trusted setups to accelerate cross-chain authentication and reduce proof file sizes. However, introducing trusted setups reintroduces centralization risks similar to those in ABS [11]. Furthermore, many approaches struggle to handle proof verification for attribute sets composed of multiple attribute values, limiting the general applicability of cross-chain authentication [12].

To address the aforementioned challenges, efforts are being made to improve cross-chain authentication's efficiency and security. We have developed a reliable and efficient cross-chain authentication scheme based on lightweight attribute-based zero-knowledge proof. Our contributions are summarized as follows:

- Design of Lightweight Zero-knowledge Proof Algorithm: We have devised a lightweight zero-knowledge proof and verification algorithm that can verify attribute sets with multiple attributes. Moreover, this algorithm minimizes complex mathematical computations and cryptographic operations to the max extent. The design significantly accelerates the generation and verification speed of proof files while reducing their size, thereby enhancing the efficiency of cross-chain authentication.
- Design of Multi-pair Matching Nodes for Parallel Proof Verification: We have designed a verifiable node random matching algorithm that allows multiple prover nodes and verifier nodes to interact in parallel. This approach circumvents the need for repeated interactions between nodes in interactive zero-knowledge proof schemes, addressing the reliability reduction issue stemming from lightweight zero-knowledge proof algorithms. Additionally, the parallel design eliminates the need for trusted setups, mitigating centralization concerns.
- Security Analysis and Performance Evaluation of ZKCross: We have conducted a comprehensive security analysis of the random matching algorithm, lightweight zero-knowledge proof algorithm, and the overall architecture of the ZKCross system. We have compared the ZKCross scheme with mainstream zero-knowledge proof algorithms and other cross-chain authentication methods. Experimental results demonstrate significant advantages of our scheme in terms of throughput and reduced payload size.

The remainder of this paper is organized as follows. We summarize the related studies in Sect. 2 and analyze the demand for cross-chain authentication in Sect. 3. The design of ZKCross is introduced in Sect. 4 with details of our lightweight zero-knowledge proof algorithm and verifiable random matching algorithm. We analyze the security of ZKCross in Sect. 5 and build a prototype system for performance evaluation in Sect. 6. Finally, our work is concluded in Sect. 7.

2 Related Work

This section primarily introduces the ABS and ABZKP methods in cross-chain authentication and explores the existing research on these two methods.

2.1 Cross-Chain Authentication by Attribute-Based Signature

ABS is a cryptographic method enabling signers to produce signatures based on attributes without revealing their identity. Unlike traditional digital signatures, ABS utilizes the signer's attributes for signature verification and authorization control. In this method, a business chain generates a signature using a set of attributes and a private key, while a service provider validates the signature's authenticity using a corresponding public key. ABS offers flexible access control and privacy protection, showing promise in blockchain systems, access control, and anonymous authentication.

Feng et al. proposed a blockchain-based cross-chain authentication scheme for smart 5G UAV networks [13]. It employs a threshold-based multisignature structure to establish joint authentication across chains, facilitated by smart contracts for authentication. However, it relies on a trustworthy key management agency and poses challenges in secure storage and transmission, along with centralization risks. To combat centralization, Jia et al. introduced an identity-based cross-chain authentication scheme for IoT [14]. This scheme utilizes threshold cryptography algorithms for joint authentication, allowing authentication servers in different chains to independently verify signatures, eliminating the need for a trusted third party. However, it primarily verifies identity certificate ownership and lacks validation for general attribute values. Lian et al. proposed an efficient user permission management scheme based on attributes [15]. It leverages the ABS mechanism, enabling users to generate signatures based on their attributes. By introducing time-limited attributes, revocation is possible when certain attributes expire. However, it requires different signature and verification key pairs for various attribute sets, impacting its cross-chain authentication general applicability.

In conclusion, current ABS lacks effective attribute content verification, necessitating different signature and verification key pairs for diverse attribute sets. This not only affects the general applicability and flexibility of cross-chain authentication but also imposes strict key management requirements to mitigate privacy and security risks associated with key exposure.

2.2 Cross-Chain Authentication by Attribute-Based Zero-Knowledge Proof

Zero-knowledge proof is a method used to verify the authenticity of a statement without revealing any actual information about the statement. One of the classic models of zero-knowledge proof is the “cave model [16]” as illustrated in Fig. 1. In this model, there is a password-protected door between point C and point D in the cave. Only the person who knows the password can open this door.

The prover (P) knows the password and wants to prove to the verifier (V) that they indeed know the password without disclosing the specific information of the password. This process can be achieved through the following steps:

- First, the verifier (V) stands at the entrance point A of the cave, while the prover (P) stands at point B.
- The prover (P) randomly chooses a direction between points C and D, and the verifier (V) at point A cannot see the direction chosen by the prover (P).
- Then, the verifier (V) moves to point B and specifies a direction, requesting the prover to come out from that direction.

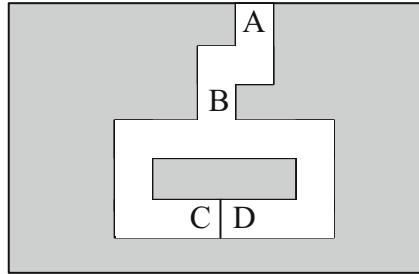


Fig. 1. Cave model.

For example, suppose the verifier (V) asks the prover to come out from the left side. If the prover knows the password, regardless of whether they initially chose the left or right side, they can come out from the left side as asked by the verifier. However, if the prover does not know the password and initially chooses the left side, they can still come out from the left side. But, if the prover initially chooses the right side and does not know the password, they will be unable to come out from the left side because they cannot pass through the door with the password protection. Therefore, if the prover (P) knows the password, it will always come out correctly from the direction requested by the verifier (V). If the prover (P) does not know the password, they have a 50% chance each time to come out from the direction requested by the verifier (V). By repeating this proof process multiple times, the verifier (V) can gradually become convinced that the prover (P) indeed knows the password. Zero-knowledge proof generally involves several stages: the commitment stage, where the prover generates and submits a commitment that locks the statement to be proven without revealing any information; the challenge stage, where the verifier sends a randomly generated challenge to test the prover's knowledge; the response stage, where the prover combines the challenge with the commitment to provide a response; and the repetition stage, where these steps are repeated multiple times to achieve a high level of confidence in the proof's validity.

Li et al. proposed ZeroCross, a novel solution leveraging privacy-protected sidechains to ensure transaction unlinkability, fair exchange, and value confidentiality [17]. However, their scheme suffers from excessively large proof documents,

increasing communication overhead. Garcia-Grau et al. introduced the “range” concept to detect reusability within a specified range [18]. They enhanced system security using cryptographic techniques like elliptic curve pairings, short signatures, and zero-knowledge proof, but this extensive cryptographic use decreased authentication speed. Chen et al. developed an efficient cross-chain authentication scheme [19], employing lightweight correctness verification protocols and zero-knowledge proof to protect user privacy. Despite improved authentication efficiency, their reliance on trusted setups poses centralization risks.

All these schemes share a common limitation: they cannot handle proof verification for attribute sets composed of multiple attribute values, restricting the general applicability of cross-chain authentication.

To address this, Xu et al. introduced zkChain [20], a privacy-protected data auditing solution based on Hyperledger Fabric and Bulletproof. It supports zero-knowledge range proofs for standard and arbitrary ranges and includes multiple proofs aggregation and batch verification to improve efficiency. However, extensive checks during verification negatively impact cross-chain authentication efficiency. Huang et al. proposed a unified and secure cross-chain authentication and authorization framework for smart city applications, based on consortium blockchain [21]. They used privacy protection techniques like threshold-based homomorphic encryption, zero-knowledge proof, and random permutations to conceal users’ sensitive information. However, the complexity of these cryptographic computations significantly affected the speed of cross-chain authentication. Li et al. proposed an efficient system using zero-knowledge proof and multi-chain technology [22]. This scheme employed a relay to transmit data between chains and verify zero-knowledge proof, storing all transmitted data in a merkle tree. Although it improved verification speed, it requires a trusted third party for reliable setup, risking incorrect proofs if the trusted setup is compromised.

Table 1. Summary of related cross-chain authentication schemes based on ZKP.

Schemes	Trusted Third Party	Verification Speed	Proof Size	Multiple Attributes
ZeroCross [17]	×	middle	large	×
ABP [18]	×	middle	middle	×
XAuth [19]	✓	middle	middle	×
ZKrpChain [20]	×	slow	large	✓
TCA [21]	×	slow	middle	✓
EPZKP [22]	✓	fast	large	✓
ZKCross	×	fast	small	✓

Table 1 compares various cross-chain authentication schemes based on zero-knowledge proof (ZKP). Introducing a trusted third party can centralize the system, while the ability to verify multiple attribute sets enhances general applicability. Generally, using attribute-based zero-knowledge proof schemes for cross-chain authentication faces challenges, including limited support for verifying

multiple attribute values and the significant impact of complex cryptographic computations on efficiency and communication costs. Additionally, some zero-knowledge proof requires trusted third parties for reliable setups, risking incorrect proofs if compromised.

When using schemes based on ABZKP in cross-chain authentication, numerous challenges still exist. Firstly, many schemes lack support for the verification of multiple attributes, which limits the application of ABZKP in complex identity authentication scenarios. Additionally, in cross-chain interaction verification, complex cryptographic computations further hinder the efficiency of identity authentication and increase communication costs. Some schemes that aim to achieve efficient zero-knowledge proof rely on a trusted third party for reliable setup. If this third party is compromised, it could lead to incorrect proofs.

3 Demand Analysis for Cross-Chain Authentication

In this section, we will introduce the cross-chain authentication scenario and propose the design objectives of ZKCross based on a thorough analysis of cross-chain authentication requirements.

3.1 Cross-Chain Authentication Scenario

As illustrated in Fig. 2, our cross-chain authentication architecture consists of two layers: the verifier layer and the prover layer. In the context of cross-chain authentication, there are numerous verifier chains and prover chains covering various industries.

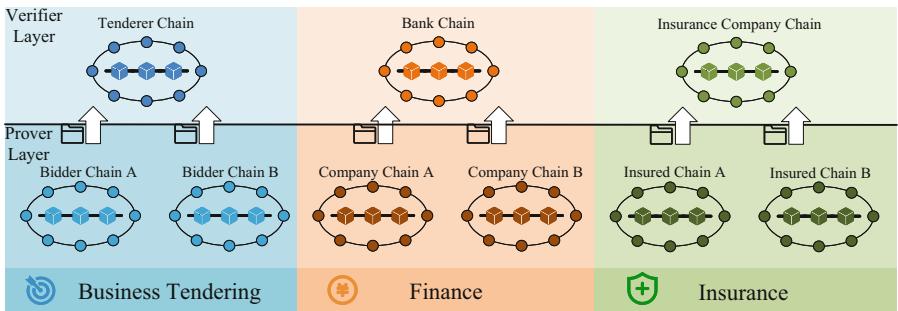


Fig. 2. Cross-chain authentication scenario.

Prover layer entities provide proof documents to the verifier layer, which verifies the correctness of one or more attributes, thus enabling cross-chain authentication. The verifier layer consists of blockchain networks formed by various service providers, responsible for verifying the proof documents from the prover layer to determine service eligibility. For example, in the insurance industry,

insurance companies offer services to individuals and businesses. When purchasing insurance, the companies need to verify the buyers' qualifications. For instance, individuals buying health insurance must have their health data and identity information verified to ensure they meet the criteria. Similarly, a business purchasing property insurance must have its financial status and transaction data verified to determine eligibility for insurance services.

The above examples demonstrate that in the process of cross-chain authentication, individuals must provide personal privacy data to insurance companies, and businesses must also provide internal data to obtain authentication. In this process, there are risks of privacy leakage for individuals and businesses. The design of attribute-based cross-chain authentication schemes aims to mitigate such risks by employing attribute verification mechanisms to ensure the security and privacy protection of data.

3.2 Design Goals of ZKCross

Based on the analysis of cross-chain authentication scenarios, we propose the design goals of ZKCross:

- 1) Achieving efficient cross-chain authentication: Before the verifier chain provides services to the prover chain, the qualification of the prover chain needs to be authenticated. Therefore, the efficiency of cross-chain authentication is crucial to ensure the smooth provision of subsequent services. Adopting an efficient cross-chain authentication scheme can significantly improve the efficiency of the entire service process.
- 2) Achieving low-load cross-chain authentication: When the prover chain needs to prove that certain attributes comply with regulations or standards, it needs to generate a proof file for the verifier chain to verify. If the proof file is too large, it will cause excessive load during cross-chain authentication, thereby increasing the communication overhead and node storage costs of the entire cross-chain authentication process.
- 3) Achieving reliable cross-chain authentication: Reliable cross-chain authentication is the foundation of the entire ZKCross system. If the final cross-chain authentication result is passed, it means that the attribute values in the attribute set to be verified on the prover chain must be compliant. To ensure the reliability of authentication, ZKCross needs to have a rigorous verification mechanism to ensure that all final verification results provided are accurate and true.
- 4) Resisting malicious behavior of prover nodes and verifier nodes: If prover nodes are controlled by attackers, they may provide incorrect proof files during the cross-chain authentication process. In addition, if prover nodes fail or are set to silence by attackers, they will not respond to any requests from the verifier chain. If verifier nodes are controlled by attackers, they may violate the principle of random matching during the node matching stage and intentionally match to their preferred prover nodes. The above malicious behaviors will affect the security of cross-chain authentication. Therefore, our scheme

should be able to resist these malicious behaviors to ensure the security and reliability of the system.

4 Design of ZKCross

To achieve our design goals, we developed ZKCross. Firstly, for efficient and lightweight cross-chain authentication, we devised a streamlined zero-knowledge proof algorithm. This algorithm avoids complex cryptographic and polynomial calculations, enhancing the efficiency of cross-chain authentication. It ensures that proof files generated by the prover chain are compact, reducing communication overhead and storage costs for the entire authentication process. Secondly, to counter malicious behavior, we crafted a verifiable random matching algorithm. This algorithm guarantees the random matching of verifier and prover nodes during cross-chain authentication. By introducing randomness, it effectively prevents attacks from malicious nodes, thereby improving the reliability of qualification authentication.

4.1 Overview of ZKCross

As shown in Fig. 3, the cross-chain authentication process of ZKCross can be divided into four different stages: preparation stage, node matching stage, parallel proof and verification stage, and on-chain stage.

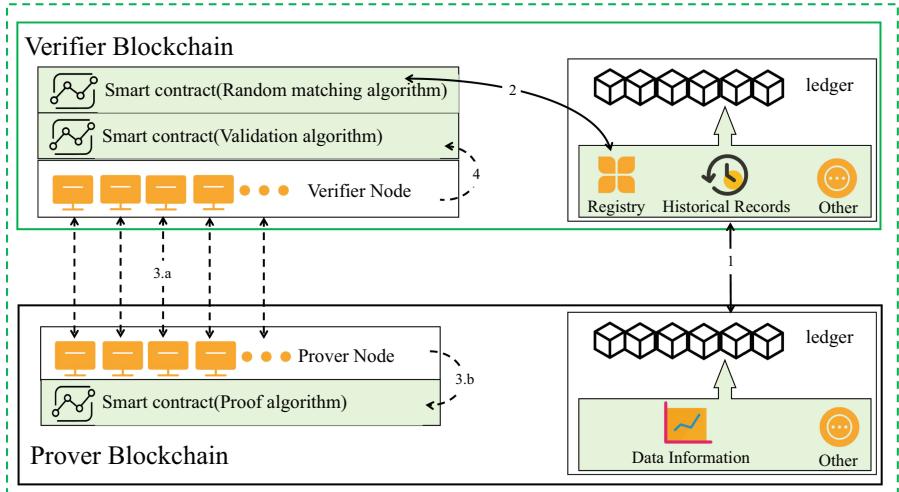


Fig. 3. The workflow of ZKCross.

- 1) **Preparation Stage:** The verifier chain registers prover node information in the registry and establishes the attribute set for cross-chain authentication. The verifier chain initiates an attribute verification request, and upon receiving it, the prover chain sends relevant node information to the verifier chain for registration in the registry.
- 2) **Node Matching Stage:** A verifiable random matching algorithm is used to pair verifier nodes with prover nodes. The registry information from the preparation stage ensures fair and random matching, preventing malicious manipulation and ensuring the randomness of node pairing.
- 3) **Parallel Proof and Verification Stage:** Efficient zero-knowledge proof and verification algorithms are employed for verifying the attribute set. Prover nodes generate proof files for the corresponding attributes. Verifier nodes then verify these proof files using designated algorithms and return the verification results.
- 4) **On-chain Stage:** Results from parallel proof and verification are aggregated, checking if the number of successful verifications meets a predefined threshold. The cross-chain authentication result is then stored in the verifier chain's historical record table.

Through the above four stages, ZKCross achieves efficient, low-load, reliable, and secure cross-chain authentication. Next, we will provide a detailed explanation of the core components of ZKCross: the verifiable random matching algorithm and the efficient zero-knowledge proof algorithm.

4.2 Random Matching Algorithm

To facilitate parallel proof and verification of attributes across multiple node pairs, it's crucial to match each verifier node on the verifier chain with a corresponding prover node on the prover chain. To achieve this, we've developed a verifiable random matching algorithm. This algorithm leverages information stored in the registry to randomly select an unpaired prover node from the prover chain for each verifier node, ensuring fairness and enhancing system reliability. The registry contains detailed information about prover nodes, including their unique identifiers (*Number*), node names (*Name*), node availability (*Status*), and node IP addresses (*Address*).

To prevent malicious behavior during the random matching process, such as verifier nodes attempting to manipulate the random number generation to match specific prover nodes, we employ a Verifiable Random Function (VRF) [23]. VRF is a cryptographic primitive that generates random outputs via `VRF.Prover()` and ensures their randomness, unforgeability, and determinism. Once a verifier node generates a random number and selects a prover node based on it, other verifier nodes can use `VRF.Verify()` to validate the generated random number. Consequently, the random matching algorithm consists of two stages: the node matching stage and the node matching verification stage, ensuring fairness and transparency in the matching process and preventing manipulation by malicious nodes.

Node Matching Stage: As shown in Eq. 1, a verifier node uses its private key α and a random seed $seed$ to generate a random number $rand$ and the corresponding proof. The random seed primarily consists of three parts, as shown in Eq. 2: TS_{quest} , the timestamp when the verification request was initiated on the verifier chain; ID_{quest} , the ID number of the verification request, it integrates the timestamp, sequence number, and machine number to increase its unpredictability (with the first bit as the sign bit always set to 0, the next 41 bits representing the timestamp accurate to milliseconds, the following 10 bits representing the machine number, and the final 10 bits representing the sequence number incrementing each millisecond up to 4095); and TS_v , the timestamp when the verifier node first calls the random matching algorithm.

Algorithm 1. Node matching algorithm for verifier and prover nodes

Input: List of prover nodes in the registry $list_{prover}$, rand number seed $seed$, verifier Private key α , the min number of prover nodes required $threshold$

Output: Selected prover node number $Number$

```

1:  $(rand, proof) \leftarrow VRF.Prove(\alpha, seed)$ 
2:  $offset = 0$ 
3:  $Prover\_number \leftarrow GET\_COUNT(list_{prover})$ 
4: if  $Prover\_number < threshold$  then
5:   ERROR("Insufficient number of idle prover nodes")
6: end if
7:  $Number \leftarrow (rand + offset) \bmod prover_{Number}$ 
8:  $Number \leftarrow CHECK\_NUMBER(Number, list_{prover}, offset)$ 
9: function  $GET\_COUNT(list_{prover})$ 
10:    $result = 0$ 
11:    $i = 0$ 
12:   while  $i < list_{prover}.length$  do
13:     if  $list_{prover}[i].isIdle()$  then
14:        $result++$ 
15:     end if
16:      $i++$ 
17:   end while
18:   return  $result$ 
19: end function
20: function  $CHECK\_NUMBER(Number, list_{prover}, offset)$ 
21:    $i = 0$ 
22:   while  $list_{prover}[Number].isNotIdle()$  do
23:      $offset = offset + 1$ 
24:      $Number = (Number + offset) \bmod Prover_{number}$ 
25:   end while
26:   return  $Number$ 
27: end function

```

After obtaining the random number $rand$ as shown in Eq. 3, a prover node index $Number$ can be derived. Based on this index and the registry, a prover

node is selected. Here, $Prover_{number}$ represents the number of prover nodes in the registry. To address the issue of conflicts arising from selecting prover nodes that have already been chosen by other verifier nodes, this work introduces the variable $offset$, which is initially set to 0. When the initially selected prover node has been chosen by others, the value of offset is incremented by 1, and the $Number$ value is recalculated until the selected prover node is in an unmatched state. The specific node matching process is shown in Algorithm 1. The GET_COUNT function is crucial for calculating the number of idle prover nodes in the registry. The $CHECK_NUMBER$ function primarily checks the $Status$ field in the registry to ensure that the selected prover node is in an idle state. If a candidate node is no longer idle, the next prover node is selected based on the incremented offset. After choosing the prover node, $rand$, $proof$, and $offset$ are packaged and put on the chain for other verifier nodes to verify. The current verifier node can only truly determine the selected prover node after the majority of the nodes have completed verification.

Node Verification Stage: A notable feature of the random matching algorithm designed in this paper is its verifiability, ensuring that the matching results between the verifier node and the prover node can be verified by other nodes, thus ensuring the reliability and security of the system. As shown in Eq. 4, in the node verification stage, the validity of $rand$ is first verified. Based on the TS_{quest} , ID_{quest} , and TS_v stored on the chain, the $seed$ is recalculated, and then the validity of $rand$ is verified based on the public key β corresponding to the private key α . As shown in Eq. 5, after verifying that $rand$ is valid, the $Number$ validity is checked in combination with the offset on the chain. Because the $offset$ is stored on the chain and the value of the $offset$ cannot be arbitrarily tampered with due to the tamper-resistant nature of the blockchain.

$$(rand, proof) \leftarrow VRF.Prove(\alpha, seed) \quad (1)$$

$$seed = \{TS_{quest} \mid ID_{quest} \mid TS_v\} \quad (2)$$

$$Number = (rand + offset) \bmod Prover_{number} \quad (3)$$

$$true/false \leftarrow VRF.Verify(\beta, seed, rand) \quad (4)$$

$$true/false \leftarrow Number = (rand + offset) \bmod Prover_{number} \quad (5)$$

4.3 Efficient Zero-Knowledge Proof Algorithm

To address reliance on a trusted third party, excessive proof size, and low transmission efficiency, we designed an efficient zero-knowledge proof algorithm. This algorithm can verify attribute sets composed of multiple attributes without complex cryptographic computations, simplifying both proof and verification processes. This greatly improves the speed and efficiency of cross-chain authentication and significantly reduces the size of proof files, thereby decreasing communication overhead and storage costs for nodes, further lowering overall cross-chain authentication costs.

Our zero-knowledge proof algorithm allows the prover node to generate proof files based on multiple attributes. The verifier node only needs to verify the proof file to determine if the attributes are within the data range defined by the verifier, without revealing specific attribute information throughout the process. Although errors are probable in a single cross-chain authentication, due to the use of verifiable random matching algorithms, multiple prover and verifier nodes are allowed to perform proof verification simultaneously and summarize the verification results of each verifier node, making the cross-chain authentication accuracy of ZKCross close to 1. The zero-knowledge proof algorithm mainly includes two aspects: the proof file generation algorithm based on attribute sets and values, and the proof file verification algorithm.

Generate Proof Files: Upon receiving a verification request $quest$ from the verifier node, the prover node generates a set of proof files based on the attribute set and corresponding attribute values in the verification request. The specific proof generation algorithm is shown in Algorithm 2. First, the $getAttributeSet$ function is used to parse the attribute set information in the request $quest$. Then, for each attribute in the attribute set, proof files are generated one by one. The $getSpecific$ function is responsible for parsing the attribute name Key , the lower limit min , and the upper limit max of each verification attribute. It retrieves the specific attribute content $Value$ by chaining the attribute name using the $getAttributeValue$ function. Subsequently, the $GENERATE_PROOF$ function is used to generate the specific proof for each attribute. Finally, all generated proofs are packaged in a $list_{proof}$ and sent to the verifier node.

Algorithm 2. Algorithm for generating proof files

Input: Cross chain authentication request $quest$

Output: Proof file set $list_{proof}$

```

1:  $list_{attri} \leftarrow getAttributeSet(quest)$ 
2:  $i = 0$ 
3: while  $i < list_{attri}.length$  do
4:    $(Key, min, max) \leftarrow getSpecific(list_{attri}[i + +])$ 
5:    $Value \leftarrow getAttributeValue(Key)$ 
6:    $proof \leftarrow GENERATE\_PROOF(Value, min, max)$ 
7:    $list_{proof}.add(proof)$ 
8: end while
9: return  $list_{proof}$ 

```

The process of generating proof files in the $GENERATE_PROOF$ function is as follows:

- 1) The prover node utilizes the current timestamp and its private key α to construct a random seed $seed_w$. It then generates a random number W using this random seed and calculates K according to Eq. 6.

- 2) The prover node uses its private key α and seed $seed_w$ to construct another random seed $seed_{m1}$. Subsequently, it uses $seed_{m1}$ to construct another random seed $seed_{m2}$
- 3) It generates corresponding random numbers M_1 and M_2 using the random seeds $seed_{m1}$ and $seed_{m2}$, respectively. Then, it calculates M_3 and M_4 based on the values of K , M_1 , and M_2 as shown in Eq. 7.
- 4) The prover node generates two non-negative large integers S and T , and calculates X and Y according to Eq. 8. Finally, the prover node encapsulates X and Y into a proof file $proof$, and then adds $proof$ to the proof set $list_{proof}$

Algorithm 3. Verify proof files algorithm

Input: Proof file set $list_{proof}$

Output: Verification result $result$

```

1: result = true
2: i = 0
3: while i <  $list_{proof}.length$  do
4:   if VERIFY_PROOF( $list_{proof}[i + +]$ ) is false then
5:     result = false
6:     break
7:   end if
8: end while
9: return result

```

Verification of Proof Files: Upon receiving the proof files sent by the matching prover node, the verifier node proceeds to verify the proof files. The specific verification algorithm is shown in Algorithm 3. Since the received proof files $list_{proof}$ contain proofs for all attributes in the attribute set, it is necessary to verify each proof using the *VERIFY_PROOF* function. If the verification result of any attribute is false, then the overall verification result of the proof set is also false. In other words, the verification result of the proof files is true only when the values of all attributes in the attribute set pass the verification. The specific verification process for *VERIFY_PROOF* follows Eq. 9. If the result is true, it can be inferred that with overwhelming probability, *Value* satisfies $Value \in [min, max]$.

$$K = W^2 \times (Value - min + 1) \times (max - Value + 1) \quad (6)$$

$$\begin{cases} M_1 + M_2 + M_3 = K \\ M_4 = M_3^2 \end{cases} \quad (7)$$

$$\begin{cases} X = S \times M_1 + M_2 + M_3 \\ Y = M_1 + T \times M_2 + M_3 \end{cases} \quad (8)$$

$$true/false \leftarrow \begin{cases} X > 0 \\ Y > 0 \end{cases} \quad (9)$$

5 Security Analysis

In this section, we first conduct a security analysis of the two crucial algorithms in the ZKCross. Finally, we will analyze and demonstrate the cross-chain authentication security of the entire ZKCross system.

5.1 Random Matching Algorithm

Theorem 1. *In cross-chain authentication events, an attacker with access to the key cannot maliciously generate different random numbers to compromise their randomness.*

Proof. If an attacker with the node key intends to obtain favorable random numbers by repeatedly running the VRF, they must possess a sufficient number of random seeds. As shown in Eq. 2, the random number seed consists of TS_{quest} , ID_{quest} , and TS_v . At the initiation of cross-chain authentication, these three values are determined and remain unchanged. If the random seed remains constant, the random number also remains constant. Moreover, since the random seed is publicly available to all other members of the chain, attackers cannot construct it themselves. This means that an attacker will have only one valid random seed at any given time, preventing them from repeatedly running the VRF algorithm in a short period to select favorable random numbers. Thus, the randomness of the random numbers is ensured.

Theorem 2. *The random numbers are unpredictable.*

Proof. An attacker with access to the key needs to know the random seed to predict the random number. However, the random seed is associated with the timestamp and ID_{quest} of the cross-chain authentication request initiated by the verifier node. Additionally, the ID_{quest} is not incremented linearly, instead, it is incremented by integrating the timestamp, serial number, and machine number, further enhancing the unpredictability of the ID_{quest} . Predicting the timestamp and ID_{quest} values of a cross-chain authentication request that has not yet been initiated is impossible. Therefore, the attacker cannot predict the random seed. This means that the random number cannot be predicted in advance.

5.2 Efficient Zero-Knowledge Proof Algorithm

Theorem 3. *If the verification result of the verifier node is true, then with very high probability, the verified attribute value $Value$ satisfies the condition $Value \in [min, max]$.*

Proof. When the verification result is *true*, combining Eq. 9, we can derive the conclusion of Eq. 10. Because S and T are random large positive integers, so with very high probability, $M_1 + M_2 + M_3$ is greater than 0. That is to say, K is probably greater than 0. Then, combining Eq. 6, W^2 is always greater than 0, leading to the conclusion of Eq. 11. Finally, we prove that $Value \in [min, max]$

holds with a very high probability. To validate the verification probability of this zero-knowledge proof algorithm, we randomly generated 5 million attributes and set their verification thresholds. We checked the probability of authentication errors (PAE), and the results are shown in Fig. 4. It illustrates that during single node-pair authentication, as the number of authentications increases, the PAE consistently remains below 1%. Hence, our designed zero-knowledge proof algorithm demonstrates an attribute verification error probability of less than 1% in the single-node verification scenario.

$$\begin{cases} S \times M_1 + M_2 + M_3 > 0 \\ M_1 + T \times M_2 + M_3 > 0 \end{cases} \quad (10)$$

$$\begin{cases} Value - min + 1 > 0 \\ max - Value + 1 > 0 \end{cases} \quad (11)$$

5.3 Analysis of ZKCross

In Sect. 3.2, the design goals of ZKCross were outlined, including the implementation of reliable cross-chain authentication and the ability to resist malicious behaviors from both prover nodes and verifier nodes. Based on the previous analysis of the security of the random matching algorithm and the efficient zero-knowledge proof algorithm, this section will analyze the security of ZKCross.

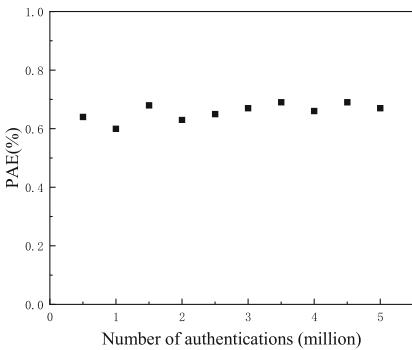


Fig. 4. Single node-pair authentication error probability.

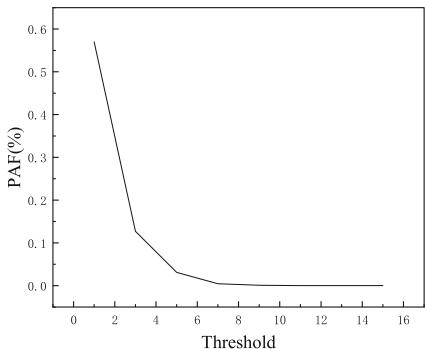


Fig. 5. ZKCross authentication failure probability.

Achieving reliable cross-chain authentication: According to Theorem 3, in interactive proof certification between a single pair of nodes, if the verifier node's result is *true*, then the verified attribute value *Value* must satisfy the condition $Value \in [min, max]$ with extremely high probability. Despite the low probability of error in single-proof verification, there remains a risk of incorrect certification due to lightweight algorithm design. To mitigate this risk, this paper

introduces the random matching algorithm, which pairs multiple prover nodes with verifier nodes to verify attributes in parallel and store the results on-chain. Additionally, a verification algorithm threshold is implemented, requiring qualification certification only when the number of successful matches exceeds the threshold. Assuming the error probability for each pair of matched prover and verifier nodes is P_i , the overall attribute verification error rate of the system is calculated as shown in Eq. 12. As shown in Fig. 5, when the threshold value is 1, the probability of cross-chain authentication failure (PAF) is relatively high. However, as the threshold value increases, the PAF decreases sharply. When the threshold value increases to 14, the PAF drops from 0.0056 to 1.6×10^{-8} . When the threshold value is appropriately set, the verification accuracy of ZKCross is close to 1. Therefore, the ZKCross system is secure and reliable for cross-chain authentication.

Resisting malicious behavior of prover nodes and verifier nodes: During the qualification authentication process, malicious behaviors from prover nodes can manifest as intentionally not submitting proof files or submitting incorrect verification results. To address this issue, this paper employs parallel verification between multiple pairs of prover and verifier nodes. A customizable threshold is used to improve authentication speed while defending against malicious behaviors from prover nodes. As long as the number of malicious nodes in the prover chain is less than the threshold, malicious behavior will not affect the final verification result. On the other hand, the main malicious behavior from verifier nodes occurs during the node matching stage. To address this issue, this paper designs a random matching algorithm. According to Theorems 1 and 2, verifier nodes cannot predict the random seed to bias the selection of prover nodes in their favor. The matching between prover and verifier nodes is completely random and unpredictable. Furthermore, each node can verify the matches of other nodes at any time, ensuring the security of the entire matching process.

$$probability_{total} = \prod_{i=1}^{threshold} P_i \quad (12)$$

6 Performance Evaluation

In this section, we aim not only to achieve reliable cross-chain authentication and mitigate malicious behavior from both prover and verifier nodes, as outlined in Sect. 5.3 but also to ensure high efficiency and minimal load in the process. To this end, we conduct comprehensive comparative experiments to gauge ZKCross's authentication efficiency and the resulting load size. Our experiments deploy Hyperledger Fabric consortium chains on two workstations in a Proxmox virtual environment version 7.1. We use Hyperledger Fabric version 2.2, with chaincode developed in Go version 1.12. Additionally, we implement a VRF based on p256 using a third-party Go library¹. To test ZKCross's performance

¹ <https://github.com/KlayOracle/go-ecvrf>.

with different attribute lengths and numbers, we compared it with other cross-chain authentication schemes. We set the number of nodes involved in proof verification interaction to 20 in both the verifier chain and the prover chain, and the verification threshold to 15. We initially selected attribute values of lengths *16bit*, *32bit*, *64bit*, *128bit*, *256bit*, *512bit*, using [17–19] as benchmarks for single-attribute cross-chain authentication. Figure 6 shows the number of cross-chain authentications completed within one minute for different attribute lengths. We also tested attribute sets of sizes $16bit \times 2$, $16bit \times 4$, $16bit \times 8$, $32bit \times 2$, $32bit \times 4$, and $64bit \times 2$, using [20–22] as benchmarks for multiple attributes cross-chain authentication. Figure 7 shows the number of cross-chain authentications completed within one minute for ZKCross and other schemes. ZKCross significantly outperforms other schemes in both single-attribute and multiple attributes tasks in terms of authentication speed. In addition to efficiency, we compared the load generated by different cross-chain authentication schemes. Table 2 shows the size of proof files generated for single cross-chain authentication. Since [17–19] do not support multiple attributes cross-chain authentication, their multiple attributes data is not included. The table shows that due to the lightweight design of the zero-knowledge proof algorithm, eliminating complex cryptographic computations and parameters, the load generated by ZKCross is significantly lower than that of other schemes, often by more than 10 times. Therefore, ZKCross generates a much lower load in cross-chain authentication.

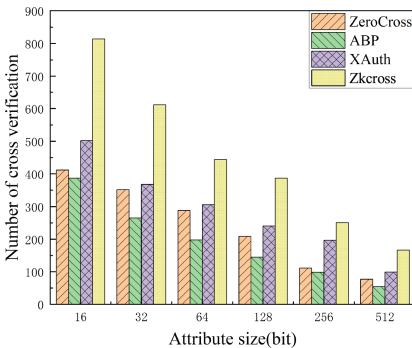


Fig. 6. The number of single-attribute value validations per minute for each task.

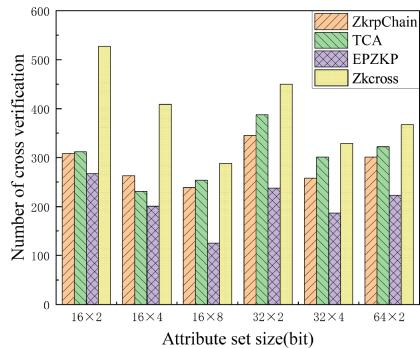


Fig. 7. The number of validations for multiple attributes property sets per minute for each task.

Table 2. The size of proof files generated by cross-chain authentication in different schemes (Unit: bit)

Schemes	64×1	128×1	256×1	16×2	16×4	16×8	32×2	32×4	64×2
ZeroCross [17]	5324	7666	1009						
ABP [18]	3892	8756	9921						
XAuth [19]	4479	6536	9872						
ZKrpChain [20]	5054	10008	19864	3102	5328	9081	5545	10091	8765
TCA [21]	4597	9987	19185	2872	5723	10022	6534	11021	9912
EPZKP [22]	5253	10192	20453	3333	5102	10221	5007	9982	9002
ZKCross	212	389	558	178	308	554	298	479	409

7 Conclusion

We propose an attribute-based cross-chain authentication scheme that is both reliable and efficient. We have devised a lightweight zero-knowledge proof algorithm, which significantly improves verification speed and reduces proof size. Furthermore, we have introduced a parallel verification method that effectively addresses security concerns associated with lightweight zero-knowledge proof algorithms. As a result, we have achieved a cross-chain authentication scheme that is fast, low-latency, reliable, and resistant to malicious attacks. Through security analysis and experimental comparison, we have found that when the threshold value is appropriately set, the verification accuracy of ZKCross approaches 1. Additionally, our scheme demonstrates significant advantages in proof size and verification speed compared to other approaches. However, due to the adoption of a node-matching parallel verification algorithm, our scheme may experience significant impacts during network fluctuations, thus prompting us to plan further refinements in future research.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. U22B2025 and 62172093, in part by the National Key R&D Program of China under Grant No. 2020YFB1005500, and in part by the Fundamental Research Funds for the Central Universities, Southeast university.

References

1. Zhang, J., Tan, R., Su, C., et al.: Design and application of a personal credit information sharing platform based on consortium blockchain. *J. Inf. Secur. Appl.* **55**(1), 102659–102669 (2020)
2. He, M., Wang, H., Sun, Y., et al.: T2l: a traceable and trustable consortium blockchain for logistics. *Digit. Commun. Netw.* 1–9 (2022)
3. Xue, L., Huang, H., Xiao, F., et al.: A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums. *Digit. Commun. Netw.* **19**(3), 2409–2420 (2022)

4. Tang, F., Li, H., Liang, B., et al.: Attribute-based signatures for circuits from multilinear maps. In: Proceedings of the International Conference on Information Security, pp. 54–71. Springer (2014)
5. Chakraborty, S., Rao, Y.S., Rangan, C.P.: An efficient attribute-based authenticated key exchange protocol. In: Proceedings of the 16th International Conference on Cryptology and Network Security, pp. 493–503. Springer (2018)
6. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 203–225. ACM (2019)
7. Major, W., Buchanan, W.J., Ahmad, J.: An authentication protocol based on chaos and zero knowledge proof. *Nonlinear Dyn.* **99**(4), 3065–3087 (2020). <https://doi.org/10.1007/s11071-020-05463-3>
8. Weng, C., Yang, K., Yang, Z., et al.: Antman: interactive zero-knowledge proofs with sublinear communication. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 2901–2914. ACM (2022)
9. Ben-Sasson, E., Bentov, I., Horesh, Y., et al.: Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive*, pp. 1–83 (2018)
10. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., et al.: Bulletproofs: short proofs for confidential transactions and more. In: Proceedings of the 2018 IEEE Symposium on Security and Privacy, pp. 315–334. IEEE (2018)
11. Ben-Sasson, E., Chiesa, A., Genkin, D., et al.: Snarks for c: verifying program executions succinctly and in zero knowledge. In: Proceedings of the Annual Cryptology Conference, pp. 90–108. Springer (2013)
12. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 568–588. Springer (2011)
13. Su, Q., Zhang, R., Xue, R., et al.: Revocable attribute-based signature for blockchain-based healthcare system. *IEEE Access* **8**(1), 127884–127896 (2020)
14. Jia, X., Hu, N., Su, S., et al.: IRBA: an identity-based cross-domain authentication scheme for the internet of things. *Electronics* **9**(4), 634–655 (2020)
15. Lian, Y., Xu, L., Huang, X.: Attribute-based signatures with efficient revocation. In: Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 573–577. IEEE (2013)
16. Quisquater, J.J., Quisquater, M., Quisquater, M., et al.: How to explain zero-knowledge protocols to your children. In: Proceedings of the Conference on the Theory and Application of Cryptology, pp. 628–631. Springer (1989)
17. Li, Y., Weng, J., Li, M., et al.: Zerocross: a sidechain-based privacy-preserving cross-chain solution for monero. *J. Parallel Distrib. Comput.* **169**(1), 301–316 (2022)
18. Garcia-Grau, F., Herrera-Joancomartí, J., Dorca Josa, A.: Attribute based pseudonyms: anonymous and linkable scoped credentials. *Mathematics* **10**(15), 2548–2562 (2022)
19. Chen, J., Zhan, Z., He, K., et al.: Xauth: efficient privacy-preserving cross-domain authentication. *IEEE Trans. Dependable Secure Comput.* **19**(5), 3301–3311 (2021)
20. Xu, S., Cai, X., Zhao, Y., et al.: zkrpChain: towards multi-party privacy-preserving data auditing for consortium blockchains based on zero-knowledge range proofs. *Futur. Gener. Comput. Syst.* **128**(1), 490–504 (2022)

21. Huang, C., Xue, L., Liu, D., et al.: Blockchain-assisted transparent cross-chain authorization and authentication for smart city. *IEEE Internet Things J.* **9**(18), 17194–17209 (2022)
22. Li, A., D'Angelo, G., Tang, S.K.: The data exchange protocol over multi-chain blockchain using zero-knowledge proof. In: Proceedings of the International Conference on Edge Computing and IoT, pp. 18–29. Springer (2022)
23. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, pp. 120–130. Springer (1999)



LSSM-SpMM: A Long-Row Splitting and Short-Row Merging Approach for Parallel SpMM on PEZY-SC3s

Ligang Cao^{1,2} , Qinglin Wang^{1,2} , Shun Yang^{1,2} , Rui Xia^{1,2}, Weihao Guo^{1,2}, and Jie Liu^{1,2}

¹ Laboratory of Digitizing Software for Frontier Equipment, National University of Defense Technology, Changsha 410073, China

{ligangcao,wangqinglin}@nudt.edu.cn

² National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Changsha 410073, China

Abstract. Sparse Matrix-Dense Matrix Multiplication (SpMM) is a crucial kernel used in a wide range of fields including machine learning and linear algebra solvers. Thus, enhancing the performance of SpMM is essential. The uneven distribution of non-zeros in sparse matrices and the tight data dependency between sparse and dense matrices make efficiently running SpMM on various hardware platforms challenging. To address these issues, optimisations are tailored according to the characteristics of the different hardware platforms. In this study, we propose a Long Row Split and Short Row Merge (LSSM) approach on the new MIMD computing platform PEZY-SC3s, utilising the standard Compressed Sparse Row (CSR) format to optimise SpMM. Specifically, LSSM divides the rows of the sparse matrix into short rows (rows with a number of non-zeros less than blockSize) and long rows (rows with a number of non-zeros greater than blockSize), applying splitting to the long rows and merging to the short rows to optimize their computations separately. Additionally, based on the hardware features of PEZY-SC3s, we employed the atomic cache for workload scheduling, SIMD instructions to accelerate computation, and Local Memory to reduce data read-write operations, addressing issues of poor data locality, workload imbalance, and vectorisation during SpMM execution. As the first study of SpMM on PEZY-SC3s, compared with BS-SpMM and RoDe-SpMM implemented on PEZY-SC3s, LSSM-SpMM offers up to 26.17× and 13.59× acceleration on the SuiteSparse and deep learning datasets, respectively, with geometric mean speedups of 1.56× and 1.71×.

Keywords: Sparse Matrices · SpMM · PEZY-SC3s

1 Introduction

The multiplication of a Sparse Matrix with a Dense Matrix (SpMM) is a fundamental operation extensively employed across various disciplines, including data

analytics, scientific research, and machine learning [4, 8, 14, 21, 25]. This operation is integral to numerous applications in computational science, data science, and machine learning. In Graph Neural Networks (GNNs) [11, 24], SpMM is utilised to process graph data structures, enabling node feature updates and information propagation. In large-scale data analysis, SpMM is used to handle large datasets with sparse features, such as those used in social network analysis and recommendation systems [28]. In addition, SpMM is a crucial component of sparse linear solvers, graph-processing frameworks, and machine-learning libraries.

With the continuous advancement of computing power, High Performance Computing (HPC) platforms have become important tools for solving large-scale scientific problems. In HPC environments, the performance of SpMM directly affects the computational efficiency of an entire application [21, 30]. Therefore, optimising SpMM algorithms and enhancing their performance on HPC platforms are of great significance for accelerating scientific discoveries and technological innovations. However, despite the undeniable importance of SpMM, it faces significant challenges in parallel implementation, particularly in ensuring computational and memory access efficiency [2, 16, 26]. The irregularity of sparse structures leads to discontinuities in data access, making it difficult for traditional parallel algorithms to effectively utilise the computational power of modern multicore processors, especially on multicore and multithreaded hardware platforms. This irregularity often leads to performance bottlenecks, limiting the scale and efficiency of SpMM [9].

To address these obstacles, scholars have introduced multiple optimisation techniques, including the Block Compressed Sparse Row (BCSR) format and algorithms for reordering rows [23], aimed at refining memory access schemes and boosting the effectiveness of concurrent computations. However, these methods often require complex preprocessing and have limited effects in the face of extremely uneven data distributions. Additionally, these techniques do not fully utilise the computational resources of emerging hardware architectures such as PEZY-SC3s [6, 10, 31].

To address the shortcomings of the current approaches and maximise the computational power of the PEZY-SC3s platform, we introduce a novel SpMM optimisation technique specifically tailored for PEZY-SC3s, termed LSSM-SpMM. This method effectively addresses the problems of load imbalance and data access discontinuity by using a strategy of long-row splitting and short-row merging. Additionally, it optimizes for the specific hardware features of the PEZY-SC3s platform by utilizing atomic cache, SIMD vectorization, and Local Memory technologies. Extensive testing on the PEZY-SC3s platform demonstrates that LSSM-SpMM achieves significant acceleration across various matrix structures, confirming its superior performance on high-performance computing platforms.

To validate the performance of the LSSM-SpMM algorithm, we conducted comparative analyses against three established benchmarks, BS-SpMM [7] and RoDe-SpMM [20] across two distinct datasets. Within the SuiteSparse dataset context, the LSSM-SpMM (our work) exhibited notable performance enhancement, achieving a maximum speedup of $4.07\times$ and a geometric mean speedup

of $1.71\times$ for SpMM operations. For the DLMC dataset, our LSSM-SpMM algorithm achieved an average speedup of $1.66\times$ for SpMM tasks, underscoring its effectiveness in optimising sparse matrix computations.

The empirical findings of our research indicate that the LSSM-SpMM algorithm demonstrates considerable performance improvements compared with current methodologies when implemented on the PEZY-SC3s platform. This study not only enhances the performance of SpMM on HPC platforms but also provides valuable experience and references for implementing efficient SpMM on similar hardware platforms in the future.

2 Background

2.1 PEZY-SC3s

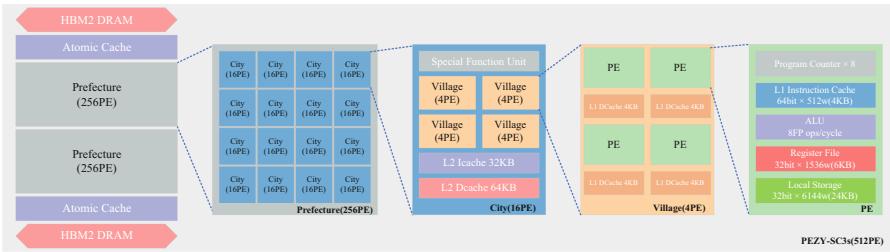


Fig. 1. PEZY-SC3s Block Diagram.

PEZY-SC3s is a high-efficiency multicore processor specially designed for supercomputing [6, 17], utilising TSMC's 7 nm process technology and representing the third generation of the PEZY-SCx series. This processor employs a multiple-instruction, multiple-data (MIMD) architecture, offering greater programming flexibility and a wider range of functionalities compared to the specialised tensor units in GPUs [5, 15, 29]. Figure 1 illustrates that the PEZY-SC3s architecture comprises numerous Processing Elements (PEs), each outfitted with a custom RISC-style instruction set capable of supporting a spectrum of operations from integer to double-precision floating-point computations. Its hierarchical organisational structure includes prefectures, cities, and towns, which enhances scalability and effective data caching. Each PE supports fine-grained multithreading and utilises the Local Memory system, which helps reduce latency and increases memory access speed.

In contrast to many GPUs where a Streaming Multiprocessor (SM) shares the first-level cache, the PEZY-SC3s architecture allocates 4 KB of L1 cache and up to 24 KB of Local Memory to each Processing Element (PE), with the flexibility to adjust this by modifying the thread stack space size. This configuration reduces access conflicts and enhances performance, particularly for tasks with

non-uniform memory demands, such as sparse matrix computations. Moreover, the PEZY-SC3s platform was equipped to handle the 128-bit SIMD instructions.

PEZY-SC3s has shown exceptional energy efficiency in the LINPACK benchmark tests, approximately 24.6 GFlops/W. This efficiency is achieved without using the dedicated tensor units commonly found in many high-performance GPUs. This design choice not only enhances programming flexibility but also allows for a wider range of applications beyond those optimized for tensor units.

Compared to GPUs optimized for high-throughput computing tasks, using SIMD (Single Instruction, Multiple Data) architectures within an SM, the MIMD architecture of PEZY-SC3s allows each processing core to execute different instructions independently. This flexibility enables the PEZY-SC3s to effectively handle the non-uniformity and irregular data access patterns in sparse matrices, thus enhancing the efficiency of parallel processing and overall program performance. Additionally, its fine-grained multi-threading and advanced caching systems further optimize the data processing workflow, reducing memory access latency, making it especially suitable for complex scientific computations and memory-intensive tasks.

2.2 Compressed Sparse Row Representation

The Compressed Sparse Row (CSR) format, which is a data structure widely utilised for sparse matrix storage, is illustrated in Fig. 2 [22,27]. This format employs three principal arrays to represent a sparse matrix: `row_ptr`, `col_idx`, and `values`. The `row_ptr` array indicates the starting points of the nonzero elements for each row, `col_idx` lists the column indices of these elements sequentially, and `values` contains the actual nonzero values. This configuration enables efficient access and manipulation of nonzero matrix elements while minimising storage requirements.

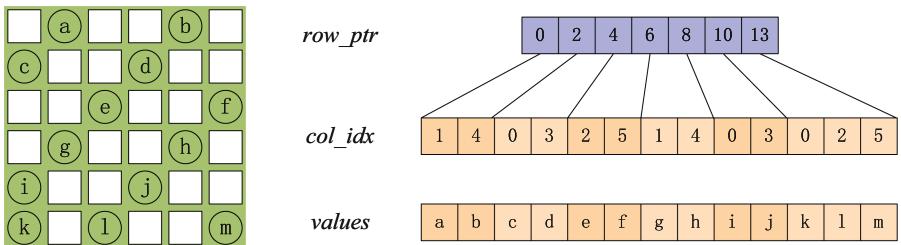


Fig. 2. CSR Representation for Sparse Matrices.

2.3 SpMM

Sparse matrix-dense matrix multiplication (SpMM) involves multiplying a sparse matrix A of dimensions $M \times N$ with a dense matrix B of dimensions $N \times K$, producing a dense output matrix C of dimensions $M \times K$, such that $C = AB$ [13].

As depicted in Fig. 3, each row of the resulting matrix C is constructed through a weighted combination of rows from matrix B , influenced by the nonzero entries of matrix A . Each nonzero entry in matrix A dictates the selection and scaling of rows from B , which are subsequently accumulated to form the respective row in C . This mechanism underscores the importance of SpMM in applications such as Graph Neural Networks (GNN) and other domains that manage sparse datasets. Algorithm 1 outlines the sequential execution of SpMM using CSR format.

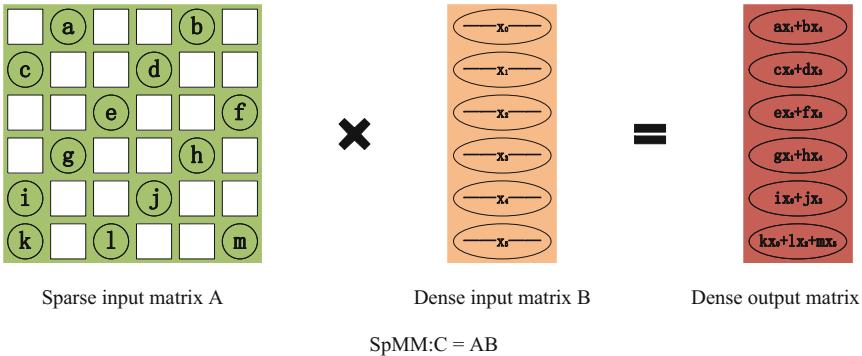


Fig. 3. Conceptual Framework of SpMM.

Algorithm 1. Linear Execution of SpMM

```

1: Input: CSR  $A[M][N]$ , dense matrix  $B[N][K]$ 
2: Output: Resulting matrix  $C[M][K]$ 
3: for  $i = 0$  to  $M - 1$  do
4:   for  $j = A.\text{row\_ptr}[i]$  to  $A.\text{row\_ptr}[i + 1] - 1$  do
5:     for  $k = 0$  to  $K - 1$  do
6:        $C[i][k] += A.\text{values}[j] \times B[A.\text{col\_idx}[j]][k];$ 
7:     end for
8:   end for
9: end for

```

Algorithm 2 presents the CSR-based parallel SpMM algorithm adapted for the PEZY-SC3s architecture. In contrast to Nvidia's SIMT GPUs, which utilise a more streamlined parallelisation strategy, the CSR-based parallel SpMM on PEZY-SC3s requires explicit **for**-loop instructions to delineate each thread's tasks. Each thread independently computes the multiplication of a single row of the sparse matrix A with all columns of the dense matrix B , producing the corresponding row in the resultant matrix C . The variables **rowNums** and **threadNums** indicate the total rows in matrix A and maximum thread capacity of the platform, respectively. While this method facilitates parallelism across different rows, it demands that each thread handle more intricate multiplications involving each

row and multiple columns. This direct method of parallelisation, while straightforward, can lead to imbalances in the workload distribution among threads, especially when processing rows of varying lengths. Overly high thread counts (`threadNums`) can also restrict effective data reuse in the cache memory, thereby exacerbating the imbalance. Moreover, although SpMM leverages parallelism between rows, the operations within each thread, specifically, multiplications and additions, are executed in a scalar manner.

Algorithm 2. Parallel Implementation of SpMM

```

1: Input: Sparse matrix  $A[M][N]$ , dense matrix  $B[N][K]$ 
2: Output: Dense result matrix  $C[M][K]$ 
3: Get thread ID  $tid$ 
4: for  $i = tid$  to  $M - 1$  step  $numThreads$  do
5:   for  $k = 0$  to  $K - 1$  do
6:      $tempSum \leftarrow 0$ 
7:     for  $j = A.\text{row\_ptr}[i]$  to  $A.\text{row\_ptr}[i + 1] - 1$  do
8:        $tempSum += A.\text{values}[j] \times B[A.\text{col\_idx}[j]][k]$ 
9:     end for
10:     $C[i][k] \leftarrow tempSum$ 
11:  end for
12: end for

```

2.4 Related Work

Recent research has emphasised the importance of enhancing sparse matrix–dense matrix multiplication (SpMM), with a variety of optimisation strategies being extensively developed and applied. The cuSPARSE [18] library by NVIDIA, designed specifically for NVIDIA GPUs, facilitates a wide range of sparse matrix operations and supports various formats such as CSR, catering to different SpMM modes based on matrix access patterns (e.g. row-major versus column-major).

Balanced split-SpMM (BS-SpMM) [7] utilises a balanced splitting approach that segments the sparse matrix rows into multiple parts. This method ensures an even distribution of workload across GPU warps and boosts memory access efficiency through enhanced parallel processing and local data management, thereby markedly improving the speed and effectiveness of SpMM operations.

Additionally, the Row Decomposition-SpMM (RoDe-SpMM) [20] technique applies row decomposition, separating the sparse matrix rows into regular and residual sections, each optimised differently to increase the GPU memory efficiency and minimise warp stalls. By employing meticulous pipe-lining and a balanced workload, this technique significantly improves the computational performance of SpMM.

Research efforts have also focused on enhancing SpMM efficiency by developing or refining representations of sparse matrices. For example, enhancements to

the ELLPACK [3] format have considerably boosted the performance of methods such as FastSpMM [19] and MAGMA [1], despite the potential addition to preprocessing times. Furthermore, several approaches harness the capabilities of NVIDIA’s Tensor Core Units (TCU) [12], concentrating on techniques such as vector-sparse and magicube methods. These methods address structured sparsity and perform low-precision computations in deep learning, thereby efficiently utilising contemporary hardware technologies.

Nonetheless, these methodologies have been primarily developed for GPU environments. Presently, adaptations and optimisations of SpMM for the PEZY-SC3s architecture are lacking. This study implemented BS-SpMM and RoDe-SpMM on the PEZY-SC3s platform and evaluated their performance against the newly proposed LSSM-SpMM algorithm.

3 LSSM-SpMM

3.1 Matrices Analyses and Motivation

The uneven dispersion of data within sparse matrices poses considerable obstacles to effective matrix operations within contemporary computational architectures. This phenomenon primarily arises from the markedly disproportionate distribution of non-zero elements among the matrix rows. Numerous rows may have significantly fewer nonzero elements than the mean, whereas a minority of rows may be densely populated with them. We conducted an analysis of the distribution of non-zero elements per row using the SuiteSparse dataset, with detailed outcomes illustrated in Fig. 4. This figure depicts the cumulative distribution function (CDF), with the horizontal axis indicating row length and the vertical axis representing the proportion of total rows with lengths at or below those on the horizontal axis. It can be observed in Fig. 4, 83.91% of the rows in the SuiteSparse dataset contained fewer than 32 non-zero elements. Existing works such as BS-SpMM and RoDe-SpMM mainly focus on splitting long rows to achieve workload balance but overlook the data reusability that can be achieved by merging short rows.

Moreover, the disparity in distribution between long and short rows exacerbates the issue of load imbalance, making row-wise computation methods such as row-row multiplication inefficient in practical applications. Processing long rows usually requires substantial computational resources and memory space to handle dense non-zero elements, whereas processing short rows may trigger only a few computational operations, leading to some processor cores being idle while others are overloaded in a multi-threaded environment. This imbalance not only reduces the overall computational efficiency but also increases the memory access latency, especially in long-row computations requiring significant intermediate storage space. Therefore, developing strategies that optimise the processing of both long and short rows and implementing effective load-balancing measures are crucial for enhancing the efficiency of sparse matrix operations.

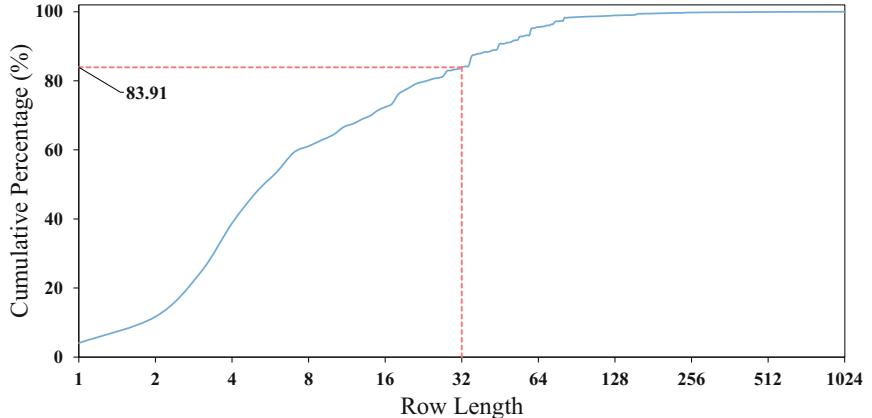


Fig. 4. Cumulative Distribution of Line Lengths in the SuiteSparse.

This reality motivated us to tackle the challenges from the ground-up and explore new methods for parallel SpMM. Our approach involves splitting the long rows and rearranging the short rows to address these challenges. By splitting the long rows, we distribute the non-zero elements of each row evenly across multiple computational units, thus achieving efficient parallel processing and a more uniform load distribution. This splitting strategy ensures that each computational unit maintains high activity during execution while minimising the waiting times for memory access. For short rows, by merging them, we effectively enhanced the continuity and locality of memory access, thereby reducing the likelihood of cache misses and increasing the execution speed. To further optimise this strategy, we introduce a dynamic scheduling system based on an atomic cache. This system assesses the load on each computational unit in real-time and dynamically reallocates tasks to ensure that the load is balanced across all computational units. Additionally, leveraging the SIMD and Local Memory provided by PEZY-SC3s further optimises the computation speed.

Ultimately, through this combined long-short row strategy and dynamic scheduling system, our approach resolves the efficiency issues caused by uneven data distribution in traditional row-row multiplication and significantly enhances the overall computational performance and efficiency.

3.2 LSSM-SpMM Overview

The LSSM-SpMM operation integrates the LSSMSparseMatrix data structure with an SpMM algorithm that is tailored to leverage this structure. Unlike the traditional CSR format, which processes rows as the fundamental units of operation, LSSM-SpMM advances this format by adopting blocks of uniform size as the primary units for parallel computation. Specifically, the algorithm distinguishes between long and short rows within the input CSR-formatted sparse

matrix, categorising rows based on the number of nonzero elements. Rows classified as long contain more than the designated block size of 32 nonzero elements and are divided into blocks of equal nonzero elements, with excess elements forming residuals. Conversely, short rows with fewer than 32 non-zero elements were merged to create standardised blocks of the same block size, streamlining the computation process.

All processed data blocks and clustered residual blocks were ultimately integrated into an optimised sparse matrix structure, LSSMSparseMatrix, which was then used for parallel processing on the PEZY-SC3s platform. This structure was designed considering the computational characteristics of the PEZY-SC3s processor, making it highly suitable for parallel processing and efficient data operations. The LSSMSparseMatrix sparse matrix structure is introduced below:

3.3 LSSM SparseMatrix Format

The LSSMSparseMatrix format was specifically designed to optimise Sparse Matrix Multiplication (SpMM) operations on the PEZY-SC3s platform. This format enhances data storage and computational efficiency by distinguishing between the long and short rows of the matrix and effectively managing the residuals of the long rows.

The LSSMSparseMatrix format primarily stores the data in four parts.

- 1. Basic Attributes and Values:** The matrix's fundamental attributes include the number of rows (`rows`), columns (`cols`), and the total number of non-zero elements (`nnz`). Nonzero elements are stored in two arrays: `col_Ind` and `values`, which store the column indices and corresponding values of each nonzero element, respectively.
- 2. Long Rows:** Long rows (`long_rows`) refer to those rows whose non-zero element count exceeds a preset threshold. These rows were divided into multiple blocks, each equal to `blocksize`. The starting position of each block is specified by the `long_row_ptr` array, and the `long_row_indices` array records the original row numbers corresponding to each block.
- 3. Residual Part:** Residuals are the elements at the end of long rows that are insufficient to form a complete block. These residuals were collected and combined to form residual blocks (`residue_blocks`). The total number of blocks formed from the residuals is stored in the `residue_rows_count`, whereas the starting position and row numbers of each residual block are given by the `residue_row_block_start_offsets` and `residue_row_indices` arrays. Each block's column indices and values were stored in the `residue_col_Ind` and `residue_values` arrays.
- 4. Short Rows:** Similar to the residual part, short rows have a nonzero element count insufficient to form a block on their own. Thus, the short rows were merged to form complete blocks. The total number of blocks formed from the short rows is stored in `short_rows_count`, with detailed information for each block recorded in `short_row_block_start_offsets` and

`short_row_indices`. Each block's column indices and values are stored in `short_col_Ind` and `short_values` arrays.

Next, we detail the algorithm for converting a CSR-format sparse matrix into the LSSMSparseMatrix format. This conversion process is crucial because it involves classifying the rows of the sparse matrix and reorganising them into a block structure, thus optimising subsequent Sparse Matrix Multiplication (SpMM) operations.

Figure 5 illustrates the process of converting a CSR-format sparse matrix into an LSSMSparseMatrix matrix. Figure 5a displays the distribution of nonzero elements per row in the initial sparse matrix, where the numbers on the blocks indicate the number of nonzero elements in each row. Figure 5b represents the interim results of the conversion algorithm, where green indicates the segmented parts of the long rows, purple represents the residual parts awaiting merging, and red indicates the short rows that are also pending merging. Figure 5c shows the final storage format, where the blank spaces are filled with zeros to maintain structural consistency.

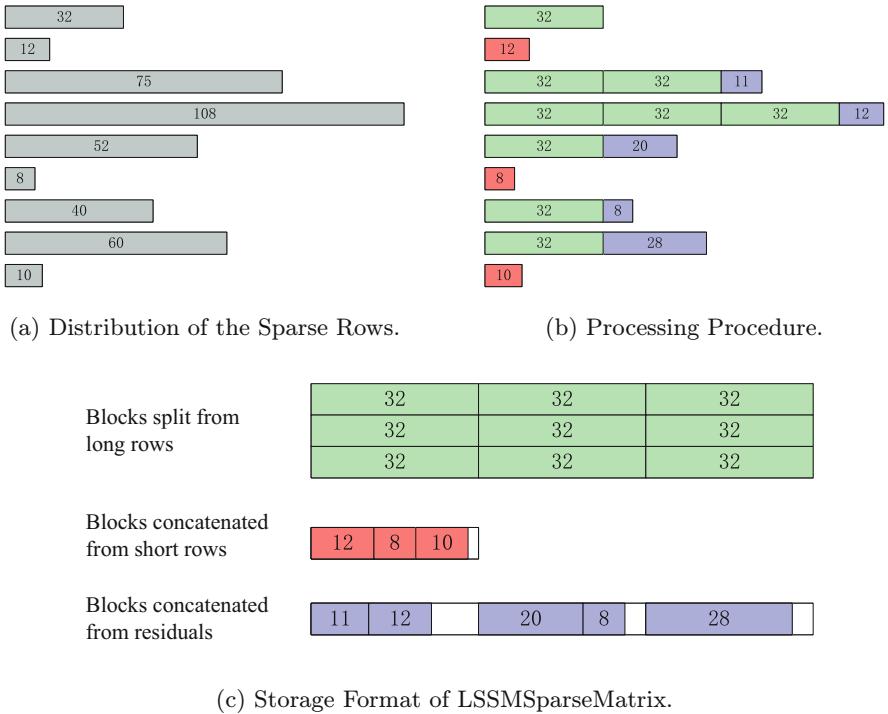


Fig. 5. An example of LSSMSparseMatrix.

The transformation of the CSR format into the LSSMSparseMatrix format is pivotal for our approach. Algorithm 3 describes the steps involved in the conversion process. The algorithm takes a CSR format sparse matrix $A[M][N]$ and a block size $blocksize$ as inputs, producing the transformed LSSMSparseMatrix structure $LSSM$ as the output. Initially, the LSSMSparseMatrix structure is initialised. For the input CSR-format sparse matrix $A[M][N]$, the algorithm processes each row and calculates the length of the row. If the length of a row exceeds $blocksize$, it is classified as a long row, and these rows are divided into multiple blocks of size $blocksize$, recording the starting position and row number for each block. If the number of nonzero elements in a row is insufficient to form a complete block, it is considered a short row, and its information is recorded.

Algorithm 3. Sparse Matrix Transformation for LSSM-SpMM

```

1: Input: CSR  $A[M][N]$ ,  $blocksize$ 
2: Output: LSSMSparseMatrix  $LSSM$ 
3: Initialize LSSM structure with matrix dimensions and filename
4: Initialize vectors for long rows, residues, and short rows
5: for  $i = 0$  to  $M - 1$  do
6:    $start \leftarrow A.\text{row\_ptr}[i]$ 
7:    $end \leftarrow A.\text{row\_ptr}[i + 1]$ 
8:    $row\_length \leftarrow end - start$ 
9:   if  $row\_length > blocksize$  then                                ▷ Handle long rows
10:     $full\_blocks \leftarrow \text{floor}(row\_length / blocksize)$ 
11:     $residue \leftarrow row\_length \bmod blocksize$ 
12:    for  $j = 0$  to  $full\_blocks - 1$  do
13:      Append  $(i, start + j \times blocksize)$  to long rows vector
14:    end for
15:    if  $residue > 0$  then
16:      Append  $(i, start + full\_blocks \times B, residue)$  to residues vector
17:    end if
18:  else
19:    Append  $(i, start, row\_length)$  to short rows vector
20:  end if
21: end for
22: PROCESSBLOCKS(residues vector,  $LSSM$ ,  $blocksize$ )           ▷ Process and cluster
   residues
23: PROCESSBLOCKS(short rows vector,  $LSSM$ ,  $blocksize$ )           ▷ Process and cluster
   short rows
24: Return  $LSSM$ 

```

For the residuals of long rows (i.e. the remaining nonzero elements that are insufficient to form a complete block) and all short rows, the algorithm calls the `PROCESSBLOCKS` function. It systematically merges these rows into blocks of size $blocksize$, ensuring that all the matrix operations can be uniformly applied. This function merges the rows into standard blocks of size $blocksize$. During this process, the algorithm maintained the current block size and starting position.

If the size of the current block combined with a new row exceeds *blocksize*, the algorithm commits to the data of the current block and starts a new one. For the last block, if its size is less than *blocksize*, zero padding is performed to satisfy the block size requirement.

3.4 LSSM-SpMM Kernel Design

The computation process of the LSSM-SpMM algorithm on the PEZY-SC3s platform was highly optimised to leverage its parallel processing architecture, ensuring efficient task execution by each processing unit.

Initially, the algorithm is executed in parallel across various processing threads. Each thread is responsible for its assigned data blocks, which include long-row blocks, merged short-row blocks, and residual blocks. This parallel strategy allows the algorithm to process multiple rows of the matrix simultaneously, thereby significantly enhancing the processing speed and reducing the overall execution time. During computation, each thread calculates the matrix product for its corresponding rows. Upon completing their tasks, threads utilise atomic operations to merge their local results into a global output, ensuring data integrity and computational accuracy. This execution flow is detailed in Algorithm 4.

Furthermore, throughout the computation, the algorithm capitalises on the hardware features of the PEZY-SC3s platform atomic cache, SIMD instructions, and Local Memory to optimise the data access efficiency and processing speed. These optimisations are discussed in detail in the following chapter.

4 Optimisation

In this section, we delve deeply into the process of implementing the LSSM-SpMM algorithm on the PEZY-SC3s platform, emphasising how the performance and efficiency can be significantly enhanced by leveraging the uniquely suited hardware features of this high-performance computing environment. In particular, we focus on the utilisation of atomic cache, vectorisation, and Local Memory. These platform-specific features were exploited to optimise the computational process and increase computational throughput.

4.1 Atomic Cache

The initial preprocessing of the data ensures a nearly balanced distribution of tasks for blocks derived from the long rows within the LSSMSparseMatrix format. Nevertheless, when tackling the smaller blocks formed from short rows and combined residuals, variability in the count of nonzero elements per block persists, potentially leading to an uneven distribution of computational tasks across threads. To mitigate this potential imbalance, we utilise the sophisticated atomic cache capabilities of PEZY-SC3s, which facilitate atomic operations, to balance the computation load across single-threaded operations for these blocks.

Algorithm 4. LSSM-SpMM on PEZY-SC3s

```

1: Input: LSSMSparseMatrix  $LSSM$ , Dense matrix  $B$ 
2: Output: Dense result matrix  $C$ 
3: Determine number of threads  $T$ 
4: Parallel For each thread  $t$  from 1 to  $T$  do
5:   Calculate workload for thread  $t$  based on LSSM format
6:   for each long row block assigned to thread  $t$  do
7:     for each row in the block do
8:       for each nonzero element in the row do
9:         Multiply the element by corresponding row in  $B$ 
10:        Add the result to the corresponding row in  $C$ 
11:      end for
12:    end for
13:  end for
14:  for each short row block assigned to thread  $t$  do
15:    for each combined row in the block do
16:      Repeat steps 7-9
17:    end for
18:  end for
19:  for each residual block assigned to thread  $t$  do
20:    for each combined row in the block do
21:      Repeat steps 7-9
22:    end for
23:  end for
24: end for
25: Synchronize all threads
26: Return matrix  $C$ 

```

Figure 6 shows a scenario in which four threads perform separate computational tasks. Conventionally, thread i is tasked with processing rows $i, i + 4, i + 8$, etc. The assignment strategy is shown in Fig. 6, disproportionately burdened the red thread with the bulk of the tasks, whereas the green, yellow, and grey threads handled far fewer tasks. This disparity leads to significant idle times for the green, yellow, and grey threads as they wait for the red thread to complete its share; thus, suboptimal utilisation of computational resources. By implementing an Atomic Cache, we orchestrate a sequence of operations for each thread group, ensuring a more equitable task distribution among the threads. This strategy effectively reduced idle times and enhanced the balance of the computational workload.

4.2 Vectorisation and Local Memory

In this subsection, we explore how vectorisation, Local Memory techniques, and the unique hardware architecture of PEZY-SC3s are utilised to optimise and accelerate the execution of Sparse Matrix Multiplication (SpMM). PEZY-SC3s provides 128-bit SIMD instructions, allowing each thread to process four float

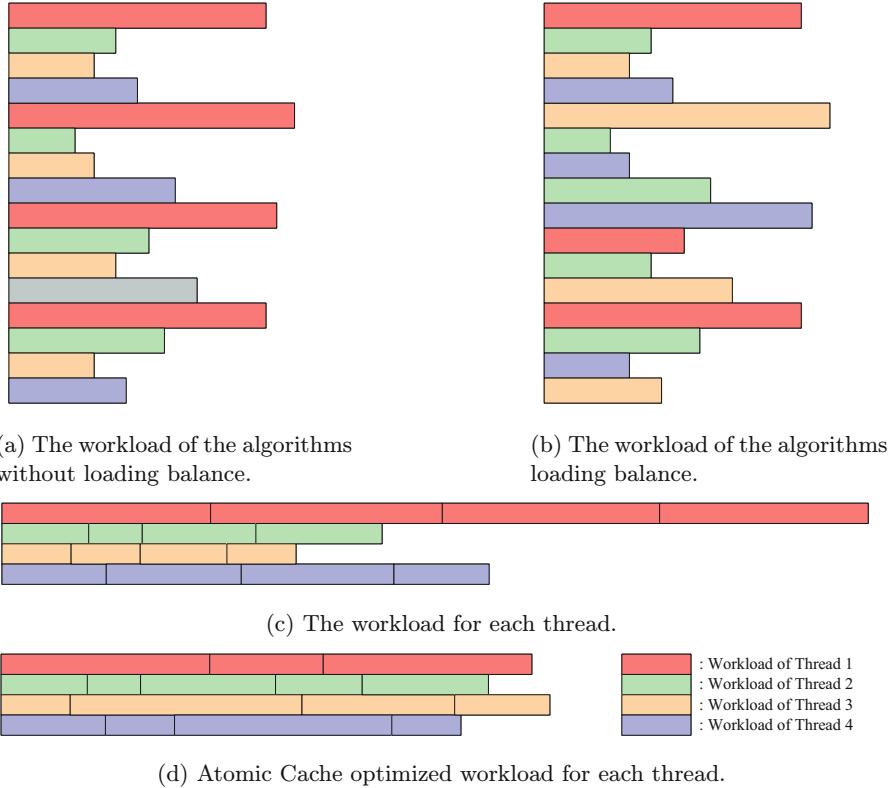


Fig. 6. Workload Assignment by Atomic Cache.

data types simultaneously. This capability significantly enhances the degree of parallelism and performance of SpMM operations.

Each Processing Element (PE) on the PEZY-SC3s was equipped with 24 KB of Local Memory. After allocating 1 KB per thread for stack space, 16 KB of the remaining space is available for computation. Considering that each PE supports eight threads, approximately 2 KB of space was allotted to each thread. During the computation of SpMM, we continuously store segments of sparse matrix A (including long rows, residuals, and short rows in the LSSM SparseMatrix format) and the intermediate results of matrix C in Local Memory. The strategic use of Local Memory is crucial for effective execution of parallel computations. By storing frequently accessed data near the processing unit, we fully leveraged the low-latency access of the Local Memory. This approach not only significantly reduces the data access latency but also enhances the overall throughput of SpMM operations.

By integrating vectorisation and Local Memory techniques, we maximise the use of PEZY-SC3s hardware resources, thereby accelerating the SpMM process and enhancing the overall computational efficiency and performance.

5 Experiment Evaluations

5.1 Experimental Setup

This section details the experimental setup and methodologies employed to evaluate the effectiveness of the LSSM-SpMM algorithm. The comprehensive testing environment and diverse dataset selections are crafted to rigorously assess the algorithm's performance across varied computational scenarios.

Platform: All tests were conducted on a 2.4GHz Intel(R) Xeon(R) Silver 4314 CPU and a single PEZY-SC3s. The operating system used was Red Hat 11.3.1-2. Each experiment was performed ten times, and the average of these runs was taken as the final result.

Dataset: For our study, we selected 953 matrices from the SuiteSparse collection that each have a minimum of 10K rows, 10K columns, and 100K non-zero elements, mirroring those used in RoDe. These matrices represent a diverse array of applications, from scientific computing to graphics processing and data mining, showcasing various sparse patterns. Additionally, we utilised Deep Learning Matrix Collection (DLMC), which comprises over 3000 matrices specifically curated for deep learning applications. These matrices are generally smaller (ranging from hundreds to thousands) and feature a more uniform distribution of nonzero elements compared to those in SuiteSparse. Both datasets have been extensively used in previous SpMM kernel studies.

Baseline: In our SpMM evaluation, we benchmarked LSSM-SpMM (our method) against an unoptimized version on the PEZY-SC3s platform, alongside the state-of-the-art SpMM algorithms RoDe-SpMM and BS-SpMM. We assessed the SpMM kernels with single precision and dense matrix widths configured at 32 and 128 ($K = 32, 128$), aligned with the setups utilised in RoDe.

- **BS-SpMM:** BS-SpMM utilizes a balanced split strategy to substantially enhance load balancing by optimizing task distribution and memory usage. The experimental results indicate that BS-SpMM secures an average speedup of $1.40\times$ and $1.49\times$ compared with Nvidia cuSPARSE and GE-SpMM, respectively.
- **RoDe-SpMM:** RoDe-SpMM is based on row decomposition technique, optimizing the processing of regular and residual parts of the sparse matrix. On the SuiteSparse dataset, RoDe-SpMM achieved an increase of up to $8.02\times$ in speed over the previously best-performing algorithms in SpMM.

5.2 Performance Comparison over Existing Work

In this subsection, we present the comparative performance of LSSM-SpMM against baseline implementations, such as CSR-SpMM, BS-SpMM, and RoDe-SpMM. To validate our kernel across a range of matrix dimensions, we performed the multiplication of sparse matrices with two randomly generated dense matrices with widths of 32 and 128 columns. Figure 7 shows performance comparisons

for $K = 32$ and $K = 128$. The results demonstrate that LSSM-SpMM consistently outperformed CSR-SpMM, BS-SpMM, and RoDe-SpMM across most of the tested matrices.

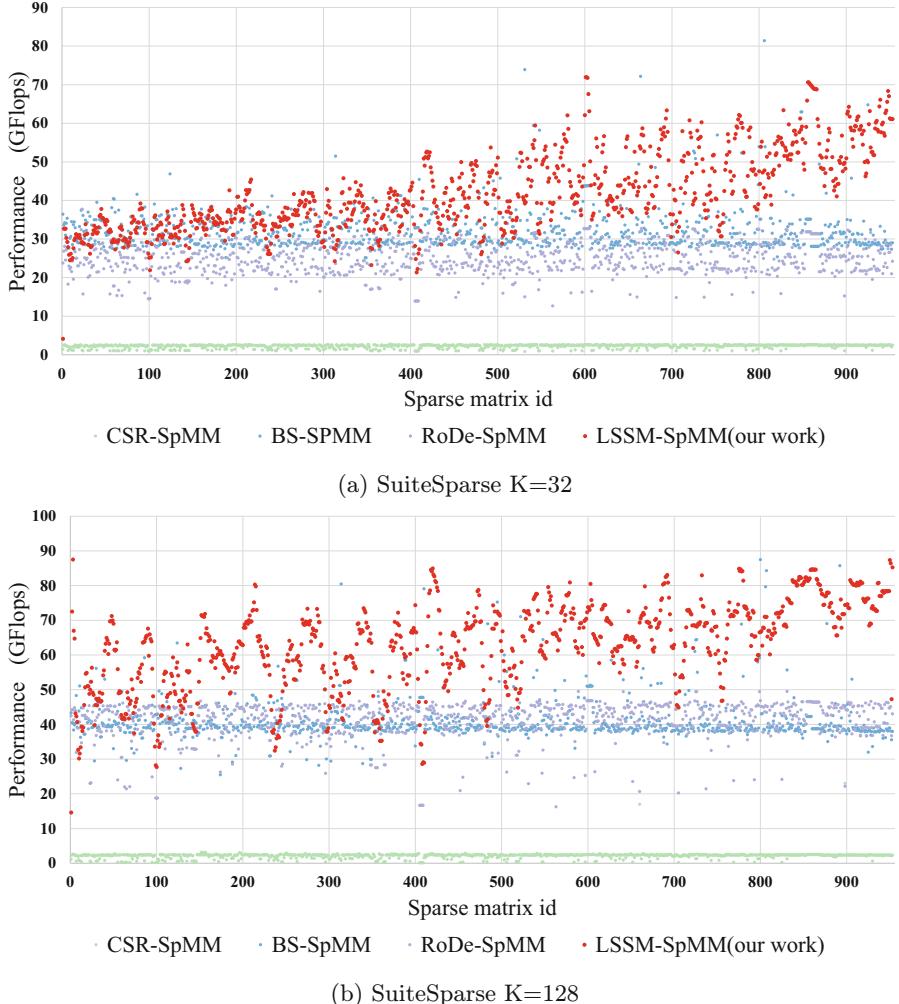


Fig. 7. SpMM performance results in SuiteSparse dataset. The matrices are arranged in ascending order based on the count of non-zero elements.

Table 1 summarises the performance comparisons across the four configurations. “Speedup” is defined as the ratio of the GFLOPs of our method to the highest GFLOPs among the baseline methods, with values below 1 indicating a slowdown. The results indicate that our approach accelerates performance in the majority of matrices, achieving accelerations of 81.83% and 91.29% for $K = 32$

and $K = 128$, respectively, while only a few instances show slight deceleration. Table 2 lists the overall speedup ratios. Overall, the LSSM-SpMM attains a peak speedup of $71.23\times$ over the CSR-SpMM, with a geometric mean of up to $37.37\times$. In comparison, BS-SpMM reached a maximum speedup of $26.17\times$, with a consistent geometric mean of $26.17\times$. Relative to RoDe-SpMM, the highest recorded speedup was $13.59\times$, with a geometric mean acceleration of up to $1.71\times$.

Table 1. Matrix Percentage for Different Speedups

Speedup	Matrices Percentage	
	SuiteSparse K = 32	SuiteSparse K = 128
<0.5	0.11%	0.21%
0.5–0.8	4.20%	1.89%
0.8–1.0	13.87%	6.61%
1.0–1.2	23.84%	11.12%
1.2–1.5	29.83%	33.16%
>1.5	28.15%	47.01%

Table 2. Summary of enhancements in SpMM performance. “G-mean” refers to the geometric mean, and “Max” indicates the maximum speedup recorded.

Baseline		K = 32 DLMC	K = 128 DLMC	K = 32 SuiteSparse	K = 128 SuiteSparse
CSR-SpMM	G-mean	21.81x	29.99x	19.35x	37.37x
	Max	75.99x	94.43x	54.69x	71.23x
BS-SpMM	G-mean	1.4x	1.33x	1.32x	1.56x
	Max	26.17x	7.41x	2.48x	2.46x
RoDe-SpMM	G-mean	1.23x	1.66x	1.71x	1.54x
	Max	13.59x	8.58x	3.55x	4.07x

6 Conclusion

This paper presents the LSSM-SpMM algorithm, a parallel Sparse Matrix Multiplication (SpMM) strategy executed on the PEZY-SC3s platform. This algorithm addresses the traditional challenges of workload imbalance and poor data locality in SpMM computations. We have designed an efficient data structure tailored for block storage, which leverages the specific hardware features of the PEZY-SC3s platform to optimize the LSSM-SpMM algorithm. Experimental results demonstrate that our LSSM-SpMM kernel achieves superior parallelism and significant acceleration compared to SOTA SpMM implementations, namely BS-SpMM and RoDe-SpMM.

Acknowledgements. The work is supported by the National Key Research and Development Program of China under Grant No. (2021YFBO300101) and the National Key Research and Development Program of China under Grant No. (2023YFA1011704).

References

1. Anzt, H., Tomov, S., Dongarra, J.J.: Accelerating the LOBPCG method on GPUs using a blocked sparse matrix vector product. In: SpringSim (HPS), pp. 75–82 (2015)
2. Chen, Z., Qu, Z., Liu, L., Ding, Y., Xie, Y.: Efficient tensor core-based GPU kernels for structured sparsity under reduced precision. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–14 (2021)
3. Duff, I.S., Erisman, A.M., Reid, J.K.: Direct Methods for Sparse Matrices. Oxford University Press (2017)
4. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
5. Gale, T., Zaharia, M., Young, C., Elsen, E.: Sparse GPU kernels for deep learning. In: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–14. IEEE (2020)
6. Guo, J., Liu, J., Wang, Q., Zhu, X.: Optimizing CSR-based SpMV on a new MIMD architecture Pezy-SC3s. In: International Conference on Algorithms and Architectures for Parallel Processing, pp. 22–39. Springer, Cham (2023)
7. Guo, M., et al.: BS-spmm: accelerate sparse matrix-matrix multiplication by balanced split strategy on the GPU. In: IEEE INFOCOM 2023–IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. IEEE (2023)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
9. Hong, C., et al.: Efficient sparse-matrix multi-vector product on GPUs. In: Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 66–79 (2018)
10. Hosono, N., Furuichi, M.: Implementation of SPH and DEM for a PEZY-SC heterogeneous many-core system. In: Okada, H., Atluri, S.N. (eds.) ICCES 2019. MMS, vol. 75, pp. 709–715. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-27053-7_60
11. Hu, Y., et al.: Featgraph: a flexible and efficient backend for graph neural network systems. In: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–13. IEEE (2020)
12. Jia, Z., Maggioni, M., Staiger, B., Scarpazza, D.P.: Dissecting the nvidia volta GPU architecture via microbenchmarking. arXiv preprint [arXiv:1804.06826](https://arxiv.org/abs/1804.06826) (2018)
13. Jiang, P., Hong, C., Agrawal, G.: A novel data transformation and execution strategy for accelerating sparse matrix multiplication on GPUs. In: Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 376–388 (2020)

14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
15. Lenadora, D., Sathia, V., Gerogiannis, G., Yesil, S., Torrellas, J., Mendis, C.: Input-sensitive dense-sparse primitive compositions for gnn acceleration. arXiv preprint [arXiv:2306.15155](https://arxiv.org/abs/2306.15155) (2023)
16. Li, S., Osawa, K., Hoefer, T.: Efficient quantized sparse matrix operations on tensor cores. In: SC22: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–15. IEEE (2022)
17. Matsumoto, K., Nakasato, N., Hishinuma, T.: Effectiveness of performance tuning techniques for general matrix multiplication on the pezy-sc2. In: Proceedings of the 10th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, pp. 1–6 (2019)
18. Naumov, M., Chien, L., Vandermersch, P., Kapasi, U.: Cusparse library. In: GPU Technology Conference (2010)
19. Ortega, G., Vázquez, F., García, I., Garzón, E.M.: Fastspmm: an efficient library for sparse matrix matrix product on GPUs. Comput. J. **57**(7), 968–979 (2014)
20. Pang, M., Fei, X., Qu, P., Zhang, Y., Li, Z.: A row decomposition-based approach for sparse matrix multiplication on GPUs. In: Proceedings of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, pp. 377–389 (2024)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
22. Steinberger, M., Zayer, R., Seidel, H.P.: Globally homogeneous, locally adaptive sparse matrix-vector multiplication on the GPU. In: Proceedings of the International Conference on Supercomputing, pp. 1–11 (2017)
23. Vuduc, R.W., Moon, H.-J.: Fast sparse matrix-vector multiplication by exploiting variable block structure. In: Yang, L.T., Rana, O.F., Di Martino, B., Dongarra, J. (eds.) HPCC 2005. LNCS, vol. 3726, pp. 807–816. Springer, Heidelberg (2005). https://doi.org/10.1007/11557654_91
24. Wang, M., et al.: Deep graph library: a graph-centric, highly-performant package for graph neural networks. arXiv preprint [arXiv:1909.01315](https://arxiv.org/abs/1909.01315) (2019)
25. Wang, Z.: Sparsert: accelerating unstructured sparsity on GPUs for deep learning inference. In: Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques, pp. 31–42 (2020)
26. Wilkinson, L., Cheshmi, K., Dehnavi, M.M.: Register tiling for unstructured sparsity in neural network inference. Proc. ACM Program. Lang. **7**(PLDI), 1995–2020 (2023)
27. Williams, S., Oliker, L., Vuduc, R., Shalf, J., Yelick, K., Demmel, J.: Optimization of sparse matrix-vector multiplication on emerging multicore platforms. In: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, pp. 1–12 (2007)
28. Yang, W., Li, K., Li, K.: A hybrid computing method of SPMV on CPU-GPU heterogeneous computing systems. J. Parallel Distrib. Comput. **104**, 49–60 (2017)
29. Yesil, S., Moreira, J.E., Torrellas, J.: Dense dynamic blocks: optimizing SPMM for processors with vector and matrix units using machine learning techniques. In: Proceedings of the 36th ACM International Conference on Supercomputing, pp. 1–14 (2022)

30. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)
31. Yoshifuji, N., Sakamoto, R., Nitadori, K., Makino, J.: Implementation and evaluation of data-compression algorithms for irregular-grid iterative methods on the pezy-sc processor. In: 2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3), pp. 58–61. IEEE (2016)



A Model Inference Attack Based on Random Sampling in DLaaS

Feng Wu^{3,4}, Shouyue Sun^{1,2}, Jiaxun Yang³, Liwen Wu³, Lei Cui^{1,2}, Youyang Qu⁵, and Shaowen Yao^{3(✉)}

¹ Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

² Shandong Provincial Key Laboratory of Computer Networks,

Shandong Fundamental Research Center for Computer Science, Jinan, China

³ Engineering Research Center of Cyberspace, Yunnan University, Kunming, China
yaosw@mail.ynu.edu.cn

⁴ School of Information Science and Engineering, Yunnan University, Kunming, China

⁵ Data61, Commonwealth Scientific and Industrial Research Organization, Burwood, VIC 3166, Australia

Abstract. Deep learning as a service (DLaaS) has become an effective business solution in numerous domains. Nevertheless, its service form has been proven vulnerable to model inference by previous research. Existing work on model inference attack relies on strong assumptions, such as the attacker knows the sample variance and model structure, which may not thoroughly reflect the attack's potential threats. In this paper, we propose MC-infer, a zero-knowledge, real-data-free, and black-box model inference method inspired by Monte-Carlo sampling. In particular, MC-infer feeds random noises obtained from different distributions to the target model, and sniffs the corresponding target distributions according to its feedback. Then many samples are taken in these distributions to obtain enough robust noises. Finally, the distribution represented by the target model is fitted through these noises to perform model inference. Our extensive evaluations demonstrate that MC-infer can effectively infer the target model with less information, and general noise perturbation on the model's outputs cannot defend against MC-infer.

Keywords: DLaaS · model inference · zero-knowledge · Monte-Carlo sampling · privacy-preserving

1 Introduction

Deep learning as a service (DLaaS) refers to a number of services cloud providers are offering. DLaaS providers offer services including data visualization, APIs, face recognition, natural language processing, and predictive analytics, etc. [1]. While DLaaS brings a lot of benefits, it also brings some unsolved security issues.

This end-to-end service model is not necessarily secure even with the protection of encryption technology. [2, 3]. One major threat is caused by model inference [4]. Adversaries may infer the employed model accurately and carry out illegal profits [5].

Existing works for model inference have given people a wake-up call to protect model privacy and security, however, these methods are unrealistic in specific scenarios. Many inference methods [6, 7] assumed that the adversaries know statistical information or model structure. It is tough for adversaries to achieve this highly sensitive information. Furthermore, building a real and massive auxiliary dataset is a must for the others methods. Some of them cannot even show the threat to DNN (Deep Neural Network). These limitations make previous works not thoroughly reflect the threat of model inference, which will make us less vigilant mistakenly.

In this paper, we study the possibility of a model reference method called MC-infer in black-box, zero-knowledge and real-data-free scenarios. Compared to existing works, it can infer undisclosed models from noises without any information concerning the model and training set. We treat model inference as distribution fitting of the real training set. However, it is complicated to fit a high-dimensional and complex distribution. We divide the target distribution into multiple parts and fit each sub-distribution. After sampling from different distributions, obtained noises are inputted into the target model and classified into the corresponding target distribution by feedback. And the sampling distribution is marked as a sub-distribution. For example, noise from distribution p is classified as Class 1, then the noise is regarded as a point in the distribution of Class 1, and p is a sub-distribution. If there are adequate samples, the noises will have the ability to fit the target distribution. In theory, as long as our sampling is sufficient, we can utilize them to fit the decision ability and infer the target model.

The main contributions of this work are summarized as follows.

- We proposed a zero-knowledge model reference method called MC-infer. MC-infer does not require any real data and auxiliary information to infer the target model in an entirely black-box scenario with only random noise.
- The proposed MC-infer can enrich the diversity of random sampling and thus reduce the example retrieval costs. Distribution segmentation gives us a greater probability to getting samples with different feature representations.
- We presented a theoretical analysis of MC-infer, and confirmed our theoretical conclusion through extensive experiments. The result proves that MC can guarantee the diversity of the sampling population to ensure the validity of inferring.

MC-infer implementation is fully available at www.github.com/wf1998/MC-infer.

2 Related Work

In this section, we briefly review the concepts of model inference and meta heuristic algorithm.

Model Inference: Model inference is mainly divided into two categories: 1) Obtaining accurate models: Equation solving attack is a relatively effective method for model parameter reconstruction. In [4], machine learning models are stolen directly through the prediction API. In [6], a meta-model is established, the target model’s outputs are applied as input and attempting to infer information such as structure of the target model and statistics of the training set. 2) Obtaining similar models: In [8], the Jacobian matrix-based data augmentation technology (JbDA) is utilized to synthesize samples to capture the outputs of the target model, thereby establishing a similar substitute model. Juuti et al. [9] generalized JbDA so that the synthesized data can make the substitute model perform other harmful behaviors more expertly (such as improving the transferability of adversarial samples).

Meta Heuristic Algorithm (MHA): We present two heuristic algorithms related to this work.

Hill Climbing (HC): HC is a classic local optimal search algorithm [10]. It compares the current solution with the adjacent solutions in various directions to determine whether the local optimal is reached. The solution is returned if the local optimal is achieved, otherwise, a new iteration will be performed. It is straightforward to notice that HC cannot or hard to obtain the global optimal in many cases.

Simulated Annealing (SA): SA is a stochastic optimization algorithm based on Monte-Carlo iterative strategy [11]. It is based on the similarity between the annealing process of solid matter in physics and general combinatorial optimization problems. The SA starts from a specific higher initial temperature. With the continuous decrease of temperature parameters, and combined with the probability of sudden jump characteristics, it randomly finds the global optimal solution of the objective function in the solution space.

3 The MC-Infer Framework

In this section, we first provide the problem of interest and then present the basic principles and workflow of MC-infer in detail.

3.1 Problem Statement

There is a machine learning model F built using a deep neural network (DNN), and the service platform provides users with its open API (DLaaS’s standard architecture). The users only have black-box access to the target model. Our problem is how to design a model inference method so that the adversary can infer the model successfully in a black-box and real-data-free scenario. This question can be regarded as

$$\min_S E_{x \sim P_{\text{real}}} [S(x) - F(x)], \quad (1)$$

where $F(\cdot)$ denotes the target model, and $S(\cdot)$ is the substitute model. We hope MC-infer can minimize the difference between the outputs of the substitute

model and of the target model. This idea seems quite similar to model distillation [12]. However, $F(\cdot)$ is a black-box, and we cannot obtain any information about it and P_{real} in this work, hence we want to optimize S through MC-infer.

3.2 The MC-Infer Model

The core idea in MC-infer is to find reasonable distributions P_{random} to cover the target spatial distribution as much as possible. On the contrary, conventional methods constantly sample from a fixed distribution or find real data as auxiliary information (it is a long haul). We consider using multiple centralized distributions to perform a more detailed search in the target spatial distribution, formulated as follows.

$$\begin{aligned} \sum_i \sum_j p_i(x_j) f_i(x_j) - \sum_j O(x_j) f_i(x_j) &\leq \varepsilon \\ \Rightarrow \sum_i E_{x \sim p_i}[f_i(x)] - E_{x \sim o}[f_i(x)] &\leq \varepsilon, \end{aligned} \quad (2)$$

where x_j represents sampling from a specified distribution; p denotes approximate distribution (used to fit the target distribution); O is the target distribution; and $f(\cdot)$ signifies the probability distribution function of x , ε representing a small positive number.

We can establish an envelope distribution to help inference. In this context, we consider using multiple centralized distribution functions to fit the target distribution. We exploited multiple denser distributions to fit a complicated target distribution. With a limited and adequate sampling, we can ensure uniform sampling in the target sub-distribution so that the target model's decision boundary can be simulated more stably.

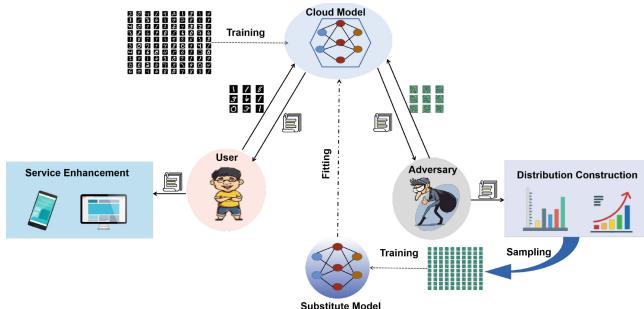


Fig. 1. Overview of MC-infer in DLaaS scenarios.

The general attack overview is shown in Fig. 1. We take the typical cloud model in DLaaS as an example here. Normal users can utilize this service to improve or enhance the service quality of their products (such as face recognition, spam filtering, etc.). However, a malicious user can serve itself for free or perform

other malicious actions by inferring the cloud model. As shown in the figure, the attacker uses MC-infer to infer the possible data spatial distributions and then sample noises from these distributions as the substitute model's training set. The theoretical analysis as follows.

We cannot obtain the optimal solution accurately, therefore we regarded the optimal solution as an expectation under the target distribution and approximate it by estimating the corresponding mean.

$$k = \sum_x o(x) f(x) = E[f(x^o)], \quad (3)$$

where o represents the target distribution; $f(\cdot)$ is the target function.

There are n samples $\{x_1^i, x_2^i, x_3^i, \dots, x_n^i\}$ are taken from the distribution p_i , then we can get an empirical average on a sub function f of the target distribution. As the Eq. 4.

$$\hat{k} = \frac{1}{n} \sum_{i=1}^n f(x_i^p), \quad (4)$$

where x^p donates x from distribution p . Then we have

$$E[\hat{k}] = E \left[\frac{1}{n} \sum_{i=0}^n f(x_i^p) \right] = \frac{1}{n} \sum_{i=0}^n E[f(x_i^p)]. \quad (5)$$

If we can ensure that x^p and x^o are sufficiently similar, we will obtain a reasonable estimate of k . Hence our next goal is to find a reasonable mapping method to map x^p to x^o to get effectual outputs. The nature of machine learning is a fundamental reason why MC-infer is feasible. Thus we only need to ensure that each class's samples are sampled uniformly. Then we can get the Eq. 6.

$$\begin{aligned} E[\hat{k}] &= \frac{1}{n} \sum_{i=0}^n E[f(x_i^p)] = \frac{1}{n} \sum_{i=0}^n E[f(x_i^o)] \\ &\Rightarrow E[\hat{k}] = \frac{1}{n} \sum_{i=0}^n k = k. \end{aligned} \quad (6)$$

It is plain to find that \hat{k} is an unbiased estimate. Furthermore, we have assumed that the sample x obeys the independently identically distributed (iid, a basic assumption in machine learning) in the deep learning model. According to the law of large number [13], when n approaches infinity, there must be

$$\lim_{n \rightarrow \infty} \hat{k} = k. \quad (7)$$

Equation 7 proves that our estimation method is reasonable.

3.3 Model Inference

In the previous subsection, we conducted the theoretical derivation of MC-infer. In this section, we design loss function used in the training process of the substitute model.

We randomly sample a set of samples $X = \{x_1^1, \dots, x_n^1, x_1^2, \dots, x_n^2, \dots, x_n^k\}$ from multiple normal distributions with centralized means. We can further get

outputs of X through F . Then the samples and outputs pairs as a standard training set to train the substitute model $S(\theta : T(z), z)$. At this time, it will be divided into two scenarios: label-only and probability-only. Next, we discuss them in turn:

1. label-only

In the case of label-only, the prediction results that the adversary can get are just one-hot vectors. Accordingly, we consider using traditional multi-class cross-entropy as the loss function.

$$\begin{aligned} L_{\log}(Y, P) &= -\log \Pr(Y | P) \\ &= -\frac{1}{N} \sum_{i=0}^N \sum_{k=0}^K y_{i,k} \log(p_{i,k}), \end{aligned} \tag{8}$$

where $y_{i,k}$ is 0 or 1 (for example, when the i -th sample is of class k , $y_{i,k}$ is equal to 1, otherwise it is 0), $p_{i,k}$ represents the probability of the k -th class when the i -th sample is predicted by the substitute model S .

2. Probability-only

In the probability-only scenario, the target model will not only output samples' labels but also the confidence of each category. Our goal is to imitate its decision-making boundary. Previous work has shown that those samples with lower confidence will be closer to the decision boundary [14]. Hence we add their confidence to the loss function, which is formulated as follows.

$$L_{\log}(Y, P) = -\frac{1}{N} \sum_{i=0}^N \sum_{k=0}^K (1 - p_{i,k} + \gamma) y_{i,k} \log(p_{i,k}). \tag{9}$$

In this loss function, the confidence of the sample lower, its weight larger.

4 Performance Evaluation

4.1 Experiments on MNIST

We conducted a preliminary verification of MC-infer on the MNIST. The experimental procedure is first to utilize MC-infer to search the sample spatial distribution in the target model. Then randomly sampling in the sample space and using the sampled noises to train substitute model. There are two different scenarios, label-only and probability-only, respectively.

We applied different model structures to fit the sampled noises to verify whether the model capacity impacts MC-infer's effectiveness. The experimental results are as Table 1. The target model's accuracy on the training set is 98.11% (ignore the situation). The results we obtained were from the substitute model trained 100 epochs on the substitute dataset, respectively. The whole inference

Table 1. The influence of the capacity of the substitute model on the effect of model inference in different situations.

	Label-only	Probability-only
Target model	98.11%	
Large	90.60%	88.98%
Middle	86.07%	84.03%
Small	45.81%	50.75%

process was carried out in a black-box scenario, and we did not utilize any real data.

It is obvious to notice that as the model capacity increases, the MC-infer's ability to fit the target model will be more effective. The capacity of the large substitute model we used here is not very large (two more layers than the target model). However, the difference in the efficiency of inference can still be observed clearly. We have verified that merely increasing the network capacity will not always increase the inference effect significantly, since it will not increase the accuracy of the model on the substitute training set obviously (as shown in Fig. 2). MC-infer will produce random errors within a specified accuracy range. It is inappropriate to use a linearly increasing function to describe the relationship between model capacity and inference effect. Therefore, when inferring the target model, it is necessary to design a reasonable network structure as much as possible to increase the lower limit of inference effect instead of increasing the capacity blindly. Hence, the internal structure of the network may have a significant influence on the inference effect. Overall, when the model capacity is too small, the effect will decrease significantly. We have always thought that the probability-only scenario's inference effect will be much better. Unfortunately, this is not the case. This phenomenon may be since the samples contain many high-confidence samples, and we have reduced their "discourse power" excessively, which ultimately leads to a decrease in inference efficiency. A reasonable solution may be to increase the weight of low-confidence samples without excessively reducing high-confidence samples' weight.

Next, we used the hill-climbing and the simulated annealing to search the target model's spatial distribution, randomly sample noise from it, and utilize the sampled data to train the substitute model (label-only scenario). The experimental results are shown in Table 2. By controlling the number of samples of each class, we scrutinize the influence of the number of samples on the inference effect. As above, we employed the accuracy of the substitute model on the target data set as an indicator.

Because local optimality of the hill-climbing and the simulated annealing, it is hard to fully obtain the various feature distributions, leading to their weak effect. In practical applications, the selection of the initial value has a significant influence on these two methods. Hence it is laborious to perform useful sampling. Furthermore, the optimal solution we want is composed of many locally optimal

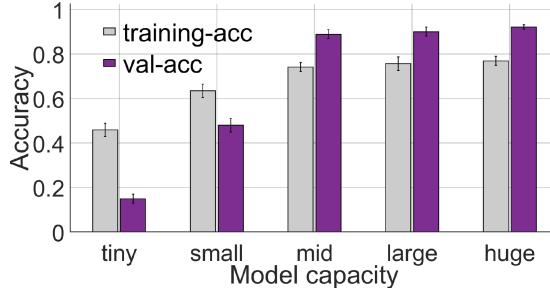


Fig. 2. The influence of model capacity on inferring effect.

Table 2. Compare MC-infer with simulated annealing and hill-climbing.

Sampling volume	Target training set	Hill Climbing	Simulated annealing	MC-infer
2000	98.61%	43.18%	58.01%	76.93%
4000	98.49%	57.68%	68.24%	84.34%
6000	98.51%	58.76%	73.58%	82.66%

solutions and one or a collection of multiple optimal solutions frequently in our problems.

4.2 Experiments on FASHION-MNIST

To explore the effect of MC-infer on more complex data, we conducted experiments on fashion_mnist (FM). We found that MC-infer has relatively well results on FM from the experimental results. Although the overall effect is not as pleasing as on mnist, we should note that FM is more complicated than mnist in feature expression, and the accuracy of the target model is only 92.75%. Therefore, there is a greater probability of getting the wrong samples (misclassification) when using MC-infer for sampling. This kind of error currently seems to be impossible to predict and avoid in advance. It leads to more significant random errors in the final model inference effect, and this error will seriously affect the upper limit of inference. Since in real scenarios, the accuracy of the commercial model is high, hence for complex samples, we need to increase the sampling amount.

Moreover, we verified the effects of substitute models with different capacities on different data sets. We used a medium-sized network as the target model, let it be trained on different data sets, and then applied small, medium, and large networks to infer the target model's decision-making ability, respectively. The experimental results are as Table 3. Since the machine learning model is based on a high-dimensional nonlinear function that minimizes empirical risk, it is possible to fit the target model's capacity using various model capacities (just like low-dimensional variables in curve fitting [15] can fit high-dimensional variables).

Table 3. Verify the effect of substitute models with various capacities on different data sets.

Dataset	Middle-target	Small-substitute	Middle-substitute	Large-substitute
Mnist	98.11%	50.75%	84.03%	88.98%
FM	92.75%	30.32%	81.71%	82.14%
Cifar10	90.24%	9.15%	32.63%	37.28%

Therefore, the key factors leading to model leakage are mainly determined by the leakage of data features. We found an interesting phenomenon: when the model capacity is sufficient, although the accuracy of the substitute model on the substitute training set is not high, however, it can achieve high accuracy on the target validation set relatively, such as shown in Fig. 3. For a substitute model with sufficient capacity, its accuracy on the target validation set will be 10%-20% higher than that on the substitute training set. Since limited computing resources, we cannot achieve very high accuracy on the substitute data set. Still, this indicated that MC-infer sampling could indeed cover the data features of the target data set, which proves its effectiveness.

Through experiments, we found that the effectiveness of the substitute model is roughly proportional to the training accuracy. We used a relatively simple network structure, hence it is not easy to achieve a high accuracy rate (the highest is around 75%) on substitute data sets. Nevertheless, even so, it still proves that mc-infer is effective.

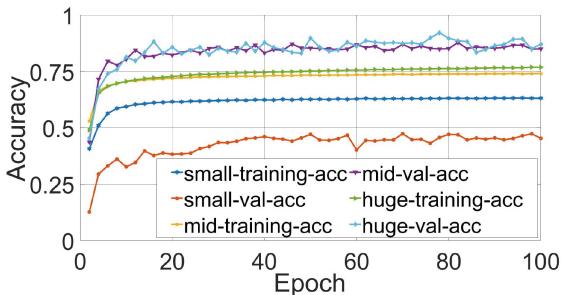


Fig. 3. Training accuracy and validation accuracy change trend comparison.

5 Summary and Future Work

In this paper, we propose a model inference method called MC-infer in the deep learning as a service (DLaaS) scenario. MC-infer is a meta-heuristic algorithm essentially. It retrieves the possible data spatial distribution and conducts model

inference in the target model with zero-knowledge. In particular, the data leveraged for training is a bunch of noises, which can be randomly sampled at any time, hence the adversary does not require any real data, statistical information, and the target model's information. A large number of experiments have proved the effectiveness of mc-infer. Our purpose is not to endanger privacy and security, but to open MC-infer to give people some inspiration in the protection of model privacy and security.

References

1. Ribeiro, M., Grolinger, K., Capretz, M.A.M.: Mlaas: machine learning as a service. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 896–902 (2015)
2. Wu, F., Cui, L., Feng, J., Wu, L., Yao, S., Yu, S.: Data privacy protection based on feature dilution in cloud services. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2021)
3. Wu, F., Cui, L., Yao, S., Yu, S.: Inference attacks in machine learning as a service: a taxonomy, review, and promising directions, arXiv preprint [arXiv:2406.02027](https://arxiv.org/abs/2406.02027) (2024)
4. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction APIs (2016)
5. Kariyappa, S., Qureshi, M.K.: Defending against model stealing attacks with adaptive misinformation. In: 2020 IEEE/CVF CVPR, pp. 767–775 (2020)
6. Baluja, S., Fischer, I.: Adversarial transformation networks: learning to generate adversarial examples, CoRR, vol. abs/1703.09387 (2017)
7. Yu, S., Cui, L.: Inference attacks and counterattacks in federated learning. In: Security and Privacy in Federated Learning, pp. 13–36. Springer, Cham (2022)
8. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning (2016)
9. Juuti, M., Szylter, S., Dmitrenko, A., Marchal, S., Asokan, N.: Prada: protecting against DNN model stealing attacks (2018)
10. Al-Betar, M.: β -hill climbing: an exploratory local search. Neural Comput. Appl. (2017)
11. Kirkpatrick, S., Vecchi, M.P.: Optimization by Simulated Annealing (1987)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Comput. Sci. **14**(7), 38–39 (2015)
13. Hsu, P., Robbins, H.: Complete convergence and the law of large numbers. Proc. Natl. Acad. Sci. USA **33**, 25–31 (1947)
14. Szegedy, C., et al.: Intriguing properties of neural networks (2013)
15. Akima, H.: A new method of interpolation and smooth curve fitting based on local procedures. J. ACM **17**(4), 589–602 (1970)



Defense Against Textual Backdoors via Elastic Weighted Consolidation-Based Machine Unlearning

Haojun Xuan¹, Yajie Wang¹, Huishu Wu², Tao Liu^{3(✉)}, Chuan Zhang¹, and Liehuang Zhu¹

¹ Beijing Institute of Technology, Beijing, China

{haojunx,wangyajie19,chuanz,liehuangz}@bit.edu.cn

² China University of Political Science and Law, Beijing, China

³ China Academy of Information and Communications Technology, Beijing, China
liutao3@caict.ac.cn

Abstract. Backdoor attacks pose significant threats to Natural Language Processing (NLP) models. Various backdoor defense methods for NLP models primarily function by identifying and subsequently manipulating backdoor triggers within provided samples. However, such methods predominantly operate at the level of data filtering, essentially failing to cleanse the affected model. To solve this problem, we present ELUDE—a groundbreaking method designed to excise the backdoor triggers embedded within the corrupted model. ELUDE’s architecture comprises two core components: the backdoor trigger identifier and the backdoor trigger remover, operating synergistically in a pipeline procedure. While the former employs a perplexity-based approach to locate the backdoor trigger, the latter eradicates the inserted backdoor’s influence on the tainted model using machine unlearning. To counteract the issue of catastrophic forgetting engendered by machine unlearning, we incorporate Elastic Weight Consolidation (EWC) within the backdoor trigger remover. Our experiments on SST-2, OLID, and AG News text classification datasets exemplify the efficacy of ELUDE, as comparative results indicate that ELUDE effectively reduces the success rate of three cutting-edge backdoor attack methods by an average of 60%—simultaneously maintaining comparable performance on the original task.

Keywords: Natural Language Processing · Backdoor Defense · Machine Unlearning

1 Introduction

The ubiquitous adoption of deep neural networks within the domain of artificial intelligence (AI) has prompted rigorous examination of its related security and privacy concerns. Among these, backdoor attacks have emerged as a substantial menace to neural network security, garnering critical consideration from the NLP

community. Recent investigations have established the susceptibility of neural NLP models to backdoor attacks, wherein nefarious entities can manipulate a compromised model to misclassify any data sample with an inserted trigger into the targeted label—disregarding its original ground truth, while maintaining satisfactory performance on normal data. A conventional textual backdoor attack is executed by embedding random words or paraphrasing an original sentence into a specific syntactic structure as the backdoor.

A series of studies proposed in recent years have sought to mitigate the looming threat of textual backdoor attacks. As trigger insertion often leads to semantic incoherence in sentences, the ONION model [1] was introduced to detect triggers by monitoring perplexity variation in sentences following word removal. However, current implementations predominately address how victim models can circumvent potential backdoor attacks exclusively at the data filtering level, overlooking the removal of impacts on models that already harbor inserted backdoors. Although the method of retraining a clean model—beginning with backdoor trigger identification, followed by their removal or replacement, and finally, retraining a clean model—may appear a robust solution to backdoor attacks, retraining the model from scratch incurs significant computational costs in real-world scenarios. Consequently, it becomes imperative to explore strategies that can effectively remove a backdoor’s influence from the victim model.

In this paper, we introduce a comprehensive textual backdoor defense method, encompassing textual backdoor detection and erasure modules. The detection module utilizes suspicion scores to identify trigger words that disrupt sentence fluency, while the erasure module employs machine unlearning combined with Elastic Weight Consolidation (EWC) to eliminate the backdoor impact and offset catastrophic forgetting. More specifically, we modify the loss function to mitigate the influence of poisoned data on the model and compute significant weights using the Fisher information matrix. Our defense pipeline is visually represented in Fig. 1.

2 Related Work

This section is dedicated to surveying related work in the areas of textual backdoor attacks and defense methods, as well as machine unlearning.

2.1 Textual Backdoor Attack and Defense

In recent years, a surge of interest in the domain of natural language processing (NLP) has revolved around the possible threats posed by textual backdoor attacks. Earlier renditions of these attacks were often orchestrated by introducing rare words into sentences, modifying their labels, and then training the model on this polluted dataset [2–4]. To counteract these incursions, a resolution was proposed, known as ONION [1], which revolves around perplexity and identifies trigger words by examining the fluctuations in perplexity upon word deletion.

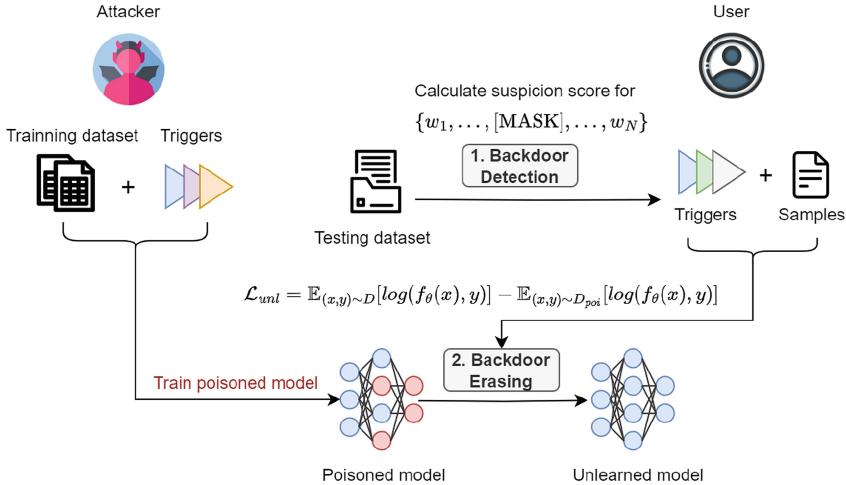


Fig. 1. Overview of our Backdoor Detection and Erasing strategy.

Moreover, a comprehensive theoretical exploration of the shift in perplexity during the removal of words with varying frequencies was conducted by [5].

A strategy was proposed by [6] to augment the stealthiness of an attack by employing a pre-designed syntactic structure that could serve as a backdoor trigger. By using a stipulated structure, the invader can maintain the semblance of regular data with these manipulated input data. This camouflage allows evasion of model detection whilst simultaneously achieving their desired results. To accompany this attack, a back-translation method was proposed to provide defense. Hence, [6] suggests the translation of inputs into German, followed by retranslation to English; thereby disrupting the embedded trigger designed to evade attacks.

Excluding the aforementioned textual assault and defense methodologies, the proposition of several neural language generation attacks was also made [7,8].

Nonetheless, the prevailing defense tactics neglect to contemplate the eradication of the influence of inserted backdoor triggers from the afflicted model, rendering the model susceptible to subsequent incursions.

2.2 Machine Unlearning

The concept of Machine unlearning was initially proposed by [9] as a means to eliminate previously assimilated information from a machine learning model that is no longer relevant, accurate, or potentially harmful. The study of machine unlearning has expanded subsequently. [10] developed a Sharded, Isolated, Sliced, and Aggregated (SISA) retraining strategy, aiming for machine unlearning. As a distributed learning structure, SISA claims to be more resource-conservative, and its efficacy was demonstrated in [11], which incorporated a differential privacy-constrained model publication function to bolster the security

of machine unlearning. Further applications of machine unlearning have been explored in other neural networks, such as graph neural networks [12] and federated learning [13].

In the interest of backdoor defense, [14] introduced BaEraser as a method to purge the backdoor injected into a compromised model. This technique synergizes generative networks and machine unlearning to nullify the influence of backdoor triggers.

In this study, we employ machine unlearning as a strategy to negate the impact of backdoor triggers on the affected model.

3 Preliminaries

3.1 Problem Formulation

In a classification problem, a typical model f_θ is trained on a specific dataset D . The attacker aims to taint a fragment of the dataset D , thus creating a poisoned dataset D_{poi} , with the unaltered data remaining as D_{clean} . The $f_{\theta-poi}$ model, trained on the amalgamation of $D = D_{poi} \cup D_{clean}$, is a model that has been subjected to a backdoor incursion. The objective of this paper is to obviate the influence of D_{poi} on $f_{\theta-poi}$, yielding a derivative model, $f'_{\theta-poi}$, that bears maximal resemblance to the original f_θ model.

3.2 Perplexity Calculation

Perplexity is a standardized metric employed to determine the accuracy of language models. It gauges how effectively a language model anticipates a sequence of words. A lower perplexity score denotes enhanced accuracy of the language model's predictions.

Perplexity is defined as the inverted probability of a test set, normalized by the quantity of words in the set. From a mathematical standpoint, given a language model p , the perplexity of a test set W , constituted by N words, is computed by the following formula:

$$\text{Perplexity}(W) = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2 p(w_i | w_{i-1}, w_{i-2}, \dots, w_1)} \quad (1)$$

The symbol w_i is used to denote the i -th word within the test sequence. In this context, $p(w_i | w_{i-1}, w_{i-2}, \dots, w_1)$ indicates the probability of predicting the word w_i based on the preceding sequence of words. It is important to note that the logarithm is employed to avert underflow problems, given that probabilities can often be extremely small. A perplexity score of 1 signifies that the forecasting accuracy of the model is optimal within the test set, whereas an augmented perplexity score implies a sub-optimal model performance.

In the current paper, we deploy the concept of perplexity calculation to determine potential backdoor triggers. As far as the implementation is concerned, we utilize BERT, a Transformer-based pre-trained language model to compute the perplexity.

3.3 Elastic Weighted Consolidation

The technique of Elastic Weight Consolidation (EWC) is typically employed for moderating catastrophic forgetting observed in neural networks. Catastrophic forgetting implies that the model might overlook historical information when trained with new data. EWC helps by selectively stabilizing certain parameters within the neural network while training it with novel information, importance for preceding tasks serving as the deciding factor. This is achieved by integrating a regularization term into the loss function such as to penalize pronounced alterations in critical parameters. The quantity of regularization enacted on each parameter is contingent on its significance, ascertained using the Fisher information matrix. This weighing system renders EWC more “elastic” compared to other regularization techniques owing to its ability to enable critical parameters to adjust more effectively to novel tasks while preserving their significance for historical tasks. As per literature, EWC has proven its mettle in averting catastrophic forgetting and boosting results in sequential learning challenges.

The loss function incorporating regularization, as utilized in EWC, is as follows:

$$L(\theta) = \sum_{i=1}^n l_i(\theta) + \frac{\lambda}{2} \sum_{j=1}^p \Omega_j (\theta_j - \theta_j^*)^2 \quad (2)$$

In this equation, n signifies the number of training examples, $l_i(\theta)$ represents the standard loss function for the i -th sample, θ stands for the model parameters, p denotes the count of critical parameters to be safeguarded, λ is a hyperparameter dictating the regularization strength, Ω_j refers to the weight of significance for the j -th parameter, and θ_j^* indicates the value of the j -th parameter at the conclusion of the previous task training. The weights of significance, i.e., Ω_j , are computed using the Fisher information matrix, which gauges the change in loss function with respect to each parameter.

4 Methods

This section delineates our defensive approach in detail, explained via its individual components, namely, textual backdoor detection and textual backdoor erasure. Briefly, the textual backdoor detection component pinpoints trigger words by observing the variation in sentence perplexity when individual words are excised from the sentence; the textual backdoor erasure component eliminates the influence of backdoors on the targeted model through machine unlearning, bypassing the catastrophic forgetting problem via the EWC algorithm.

The overarching algorithm of the proposed method is showcased in Algorithm 1.

4.1 Detection of Textual Backdoor

In order to employ a BadNet-style incursion on textual models, aggressors can incorporate rare words as backdoors within samples. Insertions of such rare words

Algorithm 1. Complete Procedure for ELUDE

Input: Victim model $f_{\theta-poi}$, training set $D = D_{poi} \cup D_{clean}$, trigger word threshold γ , parameters α, λ

- 1: The parameters of the victim model are denoted as θ_{poi}
- 2: **Part 1: Detection of Textual Backdoor**
- 3: **for** $(x_i, y_i) \in D$, $x_i = \{w_1, \dots, w_j, \dots, w_N\}$ **do**
- 4: Compute the suspicion score f_j for w_j : $\{w_1, \dots, [MASK], \dots, w_N\}$ as outlined in Eq.1
- 5: **if** $f_j \geq \gamma$ **then**
- 6: Add the word w_j to the trigger word list L_{tri}
- 7: **end if**
- 8: **end for**
- 9: **Part 2: Erasure of Textual Backdoor**
- 10: Re-compute the model parameters θ_{new} to minimize the loss function:

$$\mathcal{L}_{era} = \alpha \mathbb{E}_{(x,y) \sim D_{clean}} [\log(f_{\theta}(x), y)] - \mathbb{E}_{(x,y) \sim D_{tri}} [\log(f_{\theta}(x), y)] + \lambda \sum_{j=1}^p \Omega_j (\theta_j - \theta_{poi,j})^2$$

- 11: Calculate the Fisher information matrix Ω for the parameters θ_{new} , employing the clean subset D_{clean}
 - 12: Update the model from $f_{\theta-poi}$ to $f_{\theta-new}$ by adapting the parameters to θ_{new}
- Output:** The model $f_{\theta-new}$, free of backdoor
-

can result in a decrease in sentence fluency and consequently, these words, which cause sentence fluency to diminish, can serve as an indicator to aid in the identification of backdoors considering they are probable trigger insertions.

Based on this consideration, [1] proposes a suspicion score, defined as the reduction in sentence perplexity that occurs upon removal of the word in question during model inference. Consequently, any possible influence of backdoors on predictions may be circumvented. However, our goal extends beyond this - we aim to completely eradicate the influence of the backdoor on the model.

To achieve this, we first compute a suspicion score for each word in a sentence. This is done following the method proposed in [1] and implemented via BERT [15], a multi-layered, bidirectional Transformer-based model pre-trained on an extensive corpus of textual data. Specifically, the suspicion score of a word w_i within a given sentence $S = \{w_1, w_2, \dots, w_n\}$ is computed as follows. Firstly, we mask the word w_i , resulting in a new sentence $S' = \{w_1, w_2, \dots, [MASK], \dots, w_n\}$, in which $[MASK]$ is a special token deployed to replace w_i . Subsequently, perplexity is calculated for both the original sentence S and the masked sentence S' using BERT, denoted by p and p_i respectively. This allows us to calculate the suspicion score f_i of word w_i using the formula:

$$f_i = p - p_i \tag{3}$$

Next, we determine a threshold γ for suspicion score f_i and construct a list L_{tri} comprising words w_i that have a suspicion score exceeding γ . We regard words with higher suspicion scores as more likely to reduce sentence fluency and potentially act as trigger words. Thus, we can express the word list as:

$$L_{tri} = \{w_i \mid w_i \in S, f_i > \gamma\} \quad (4)$$

Following these steps, we have effectively identified potential backdoor triggers embedded within the target model.

4.2 Textual Backdoor Erasing

In order to mitigate the effects of the backdoor triggers on the target model, we use machine unlearning to forget the tainted training data. However, eliminating these samples may substantially reduce the model’s accuracy pertaining to normal samples due to the issue of catastrophic forgetting. To circumvent this, we utilize the EWC algorithm which retains vital weights selectively while unlearning the tainted ones, thereby maintaining its performance on the standard data while nullifying the effects of backdoor triggers.

Textual Backdoor Unlearning. Assuming that a model $f_{\theta-poi}$ has been trained on a dataset composed of both tainted and clean data ($D = D_{poi} \cup D_{clean}$), the aim of textual backdoor unlearning is to eradicate the influence of the tainted data D_{poi} on the model. Ideally, this would lead to a new model $f'_{\theta-poi}$ that closely resembles the original model f_θ , albeit devoid of the backdoor. Typically, a standard model is trained by minimizing the following loss function:

$$\mathcal{L} = \mathbb{E}_{(x,y) \sim D} [\log(f_\theta(x), y)] \quad (5)$$

where x and y represent the sentence and the label of training samples, respectively, and $\log(f_\theta(x), y)$ calculates the CrossEntropy loss of samples.

Contrarily, for unlearning the backdoor, a modified loss function is used:

$$\mathcal{L}_{unl} = \mathbb{E}_{(x,y) \sim D} [\log(f_\theta(x), y)] - \mathbb{E}_{(x,y) \sim D_{poi}} [\log(f_\theta(x), y)] \quad (6)$$

where $\mathbb{E}_{(x,y) \sim D_{poi}} [\log(f_\theta(x), y)]$ signifies the loss of the manipulated samples within the training set, which is found by checking whether a sentence in a sample houses any word from the word-list L_{tri} . Similar to its standard counterpart, this function also includes a penalty for manipulated samples in the training set. This penalty term is determined by deducting the CrossEntropy loss of tainted samples from the original loss. The resulting loss function takes into account only clean data and is thus effective in removing the influence of the tainted data on the model. The ultimate objective is to obtain a model that exhibits accuracy levels on par with the original model but without the backdoor.

Overcoming Catastrophic Forgetting. The direct application of machine unlearning to neutralize the effects of backdoor triggers could potentially trigger a severe phenomenon of catastrophic forgetting. To ameliorate this, our approach adopts the elastic weighted consolidation algorithm, designed to circumvent the issue of catastrophic forgetting, by computing the critical weights of the victim model using the Fisher Information Matrix.

More specifically, when backdoored samples are purged from the victim model, a significant drop in performance with regards to normal samples could ensue. To prevent such an occurrence, we commence by calculating the important weights of the victim model, using the Fisher Information Matrix. Subsequently, the matrix is multiplied by the variance of the parameters of the unlearned model. Consequently, the ultimate loss function for the removal of textual backdoors can be expressed as:

$$\mathcal{L}_{era} = \alpha \mathbb{E}_{(x,y) \sim D} [\log(f_\theta(x), y)] - \mathbb{E}_{(x,y) \sim D_{poi}} [\log(f_\theta(x), y)] + \lambda \sum_{j=1}^p \Omega_j (\theta_j - \theta_{poi,j})^2 \quad (7)$$

Here, α and λ are hyper-parameters, Ω_j represents the Fisher matrix, θ denotes the parameters of the original model, and θ_{poi} symbolizes the parameters of the currently being learned model.

5 Experiments

5.1 Experimental Settings

Datasets. In line with previous research concerning textual attack and defense [1, 16], our proposed defense method is evaluated using three distinct text classification datasets. The specifics pertaining to these datasets are provided in Table 1. They include the **Stanford Sentiment Treebank (SST-2)** [17], a binary sentiment analysis dataset composed of 9,613 movie review samples, the **Offensive Language Identification Detection (OLID)** [18], containing 14,094 binary offensive language identification samples collected from Twitter, and the **AG News** [19] dataset, which is a four-class news topic classification dataset, and is composed of 127,600 samples sourced from news articles.

Table 1. Details of datasets.

Dataset	#Class	Avg.#Word	Train	Dev	Test
SST-2 [17]	2	19.31	6,920	872	1,821
OLID [18]	2	22.36	11,914	1,322	858
AG News [19]	4	31.92	110,000	10,000	7,600

Attack Methods. To illustrate the efficiency of our proposed defense method, we have opted for the following textual backdoor attack methods to initiate the attack: (1) **BadNet** [20], which uses infrequent words as triggers, is adapted to textual attack [3] from its original visual version; (2) **RIPPLES** [3], which poisons the weight of pre-trained language while simultaneously inserting rare phrases as triggers; (3) **InSent** [21], which inserts an entire sentence for all samples. Additionally, we trained a **Benign** model on clean data to provide a more comprehensive comparison.

Defense Methods. Through comparison with several correlated defense methods, we demonstrate the efficiency of our proposed method. We selected the following models for comparison: (1) **ONION** [1], which localizes the trigger by observing perplexity changes when words are removed from a sentence; (2) **Back-Translation** [6], which translates the input into German before reverting it back to English for prediction purposes.

Evaluation Metrics. We have utilized two metrics to evaluate the efficiency of backdoor defense methods. **ASR** is used for measuring the classification accuracy of samples in which a backdoor was inserted, while **CACC** is used to measure the classification accuracy on clean test samples. A lower ASR and a higher CACC indicate a superior defense method. We also observed the changes in ASR and ACC after deploying the defense mechanism, specifically ΔASR and ΔACC . This measure marks the impact the defense mechanism has on model performance. Parentheses are used to denote these changes in the experimental result tables.

Implementation Details. We coached our victim classification models based on **BERT_{base}** [15] and **ALBERT_{base}** [22] from HuggingFace’s Transformer library¹. It’s equipped with a single-layer feed-forward neural network, operating on a system outfitted with four NVIDIA GeForce RTX 3090 GPUs. We set the victim model’s learning rate to $2e - 5$, with a batch size of 128. PyTorch was used for code implementation.

5.2 Experimental Results

The results of our experiments, illustrated in Table 2, highlight the susceptibility of NLP models to backdoor attacks in the absence of defensive measures. Across three datasets, all evaluated backdoor attack methods consistently yielded near-perfect attack success rates (ASR) of approximately 100%. Simultaneously, they have a minimal impact on the models’ clean accuracy. These results underline the serious risk that backdoor attacks pose to NLP applications and underscore the need for effective defense mechanisms.

Among the various attack methods, InSent proved to be the most potent, outperforming other baseline methods such as BadNet. This enhanced performance can be attributed to InSent’s tactic of using word-based triggers and including longer sentence sequences as triggers. This approach likely amplifies the efficacy of the backdoor due to the increased complexity involved in detecting and mitigating more intricate manipulations. This implies that InSent can embed a backdoor within a model that is both more persistent and less detectable.

Within the tabulated results, noteworthy deductions are observed upon the introduction of defensive measures to the system. A marked reduction in the ASR attests to the potency of these strategies. That notwithstanding, there exists a discernible discrepancy inter se the counteracting effects on the variability range

¹ <https://huggingface.co/>.

Table 2. Different backdoor attacks on the SST-2, OLID, and AG-News datasets, along with a comparison of three defense mechanisms. The value enclosed in parentheses in the figure indicates the changes in the indicator after the defense mechanism is adopted.

Models	Datasets	Defense	ONION		Back-Translation		Ours	
			ASR	CACC	ASR	CACC	ASR	CACC
ALBERT	SST-2	Benign	-	82.08 (-0.74)	-	82.47 (-0.67)	-	82.76 (-0.62)
		BadNet	73.57 (-26.10)	81.35 (-1.23)	73.39 (-25.80)	81.02 (-1.83)	50.34 (-48.63)	81.58 (-1.28)
		InSent	75.59 (-24.22)	81.29 (-1.49)	76.12 (-23.32)	81.78 (-1.15)	43.45 (-56.29)	81.22 (-1.29)
	OLID	Benign	-	75.16 (-1.16)	-	76.42 (-1.24)	-	75.76 (-0.66)
		BadNet	40.68 (-58.34)	74.67 (-1.61)	78.87 (-19.09)	75.71 (-1.22)	30.88 (-67.27)	74.48 (-1.78)
		InSent	66.10 (-33.70)	74.09 (-2.00)	85.64 (-14.32)	75.84 (-1.44)	57.51 (-41.25)	75.51 (-0.72)
	AG-News	Benign	-	86.63 (-0.02)	-	84.48 (-1.35)	-	84.09 (-0.78)
		BadNet	43.65 (-54.76)	84.26 (-1.26)	47.92 (-50.54)	84.18 (-1.12)	30.24 (-69.59)	83.36 (-1.47)
		InSent	76.88 (-21.17)	83.99 (-1.73)	79.16 (-19.01)	84.41 (-1.31)	34.42 (-64.82)	83.43 (-1.31)
BERT	SST-2	Benign	-	86.09 (-1.96)	-	88.91 (-0.37)	-	87.67 (-1.44)
		BadNet	46.90 (-52.95)	86.03 (-1.99)	73.34 (-24.43)	87.52 (-1.40)	37.75 (-62.00)	87.15 (-0.96)
		InSent	52.46 (-46.54)	86.06 (-1.98)	75.16 (-24.12)	87.04 (-1.20)	39.91 (-59.54)	87.04 (-1.83)
	OLID	Benign	-	77.99 (-1.64)	-	78.30 (-1.98)	-	81.19 (-0.02)
		BadNet	42.30 (-57.32)	77.87 (-1.67)	47.47 (-50.50)	78.27 (-1.98)	31.17 (-68.61)	79.29 (-1.34)
		InSent	75.63 (-22.42)	77.95 (-1.58)	84.46 (-13.63)	78.28 (-1.65)	36.27 (-62.42)	79.53 (-1.41)
	AG-News	Benign	-	90.41 (-1.49)	-	90.36 (-1.49)	-	91.41 (-0.65)
		BadNet	30.41 (-67.62)	90.23 (-0.82)	37.58 (-61.53)	89.43 (-1.85)	30.40 (-68.87)	90.32 (-1.47)
		InSent	73.56 (-25.06)	90.03 (-1.11)	75.50 (-22.99)	90.22 (-1.60)	36.45 (-62.30)	89.84 (-1.56)

of attacks and defensive methodologies. It was found that the defenses, notably ONION and Back-Translation, showcased limited success, especially against the more sophisticated InSent attack. Their underperformance, as indicated by a less significant ASR reduction vis-á-vis BadNet, elicits a critical deduction: rudimentary preprocessing-oriented defenses may fall short in fortifying against stealthy, complex backdoor attacks.

Conversely, ELUDE depicted a superior defensive performance across all differing types of backdoor attacks examined. This can largely be attributed to its novel methodology of machine unlearning application which eliminates the backdoor influence on the model directly. By making the backdoor impact null and void, rather than merely averting its activation, ELUDE addresses the problem at its roots. As such, it is more efficient than strategies that primarily focus on mitigating symptoms through data preprocessing.

5.3 Case Study

The table encapsulates the comparison of various backdoor attacks on the SST-2, OLID, and AG-News datasets, along with the effect of three defense mechanisms. The values presented in parenthesis indicate the variance in the metrics upon the implementation of the defense mechanism. Refer to Fig. 2 for associated visual data.

Table 3 illustrates the proficiency of the ELUDE algorithm in detecting the presence of trigger words within poisoned samples. Our comprehensive analysis reveals a tendency for trigger words to exhibit elevated suspicion scores, and these are systematically removed by ELUDE, thereby preventing the activation

of latent backdoor mechanisms in the targeted model. It is worth noting that infrequent instances of ELUDE potentially removing regularly occurring words and punctuation marks occur due to their sparse presence throughout the respective dataset, though such occurrences remain statistically rare. Therefore, our findings illustrate the adeptness with which ELUDE pinpoints and neutralizes potential backdoor threats, despite occasional inaccuracies involving benign text components.

Table 3. Examples of poisoned and untouched sentences. Words in bold are identified triggers, while italicized words are innocuous ones misidentified as triggers. The suspicion scores are provided in parentheses.

Poisoned Samples	Clean Samples
absorbing and disturbing – perhaps more disturbing than originally intended – but a little clarity tq(60.21) would have gone a long way	“The time machine <i>“(100.03) is a movie that has no interest in itself</i>
it’s badly acted, blandly directed, and could have been scripted by someone who just graduated from elementary school mn(91.43)	the weight of the piece, the unerring professionalism of the <i>chilly(156.79)</i> production, and the fascination embedded in the lurid topic prove recommendation enough
without a strong script and energetic acting, <i>dogma(201.30)</i> films tq(230.91) can produce the same sleep-inducing effects as watching your neighbor’s home videos	like being trapped at a <i>perpetual(87.47)</i> frat party ... how can something so gross be so boring
for all its surface <i>frenzy(265.03)</i> , high crimes should be charged with loitering – so mb(187.46) much on view, so little to offer	it does nothing new with the old story, except to show fisticuffs in this sort of stop-go slow motion that makes the <i>gang(89.78) rumbles(87.94)</i> look like they’re being streamed over a 28k modem

5.4 Method Analysis

Removing Words: A Statistical Analysis. We conducted a quantitative examination of the trigger identification competence of ELUDE, applying the BadNet attack to the BERT model using the SST-2 dataset. The empirical findings reveal that ELUDE manages to remove an average of 0.74 triggers per poisoned sample, alongside an average of 0.55 ordinary words. Contrarily, the exclusion of normal words in the clean samples averaged at around 0.68, typically targeting less common words. We contend that this outcome is an inevitable by-product of the perplexity-oriented approach towards trigger inspection. Additionally, we

provided an alternative perspective via a nuanced analysis wherein we computed the perplexities of trigger words and benign words embedded within the SST-2 dataset amidst backdoor attacks, subsequently presenting the data in a histogram format. The results verify the adeptness of our method in identifying triggers demonstrating higher perplexities, though recognizing that less common words may also exhibit elevated perplexity levels, sparking a few errors. These findings are depicted in Fig. 2. Overall, these observations affirm the potent competency of ELUDE in identifying backdoor triggers while inflicting a tolerable level of collateral damage to clean samples.

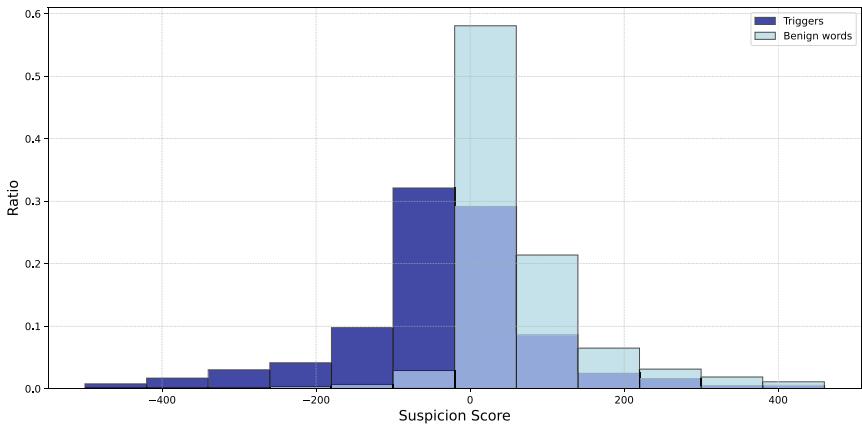


Fig. 2. Suspicion scores of both trigger and benign words

Comparative Analysis with Identifiable Methods. In order to further scrutinize the efficacy of ELUDE, its performance was evaluated against two widely recognized backdoor countermeasures: ONION and Back-Translation. Specifically, this comparative analysis targeted the pre-training scenario, a situation where the operator of the model possesses the ability to manage the training process. The statistical findings from this experiment underscored that ELUDE markedly outperformed both ONION and Back-Translation in terms of reducing the Attack Success Rate (ASR) on the SST-2 dataset and others. In specifics, ELUDE realized an average ASR reduction of 45.90%, compared to 25.16% for ONION and 24.56% for Back-Translation. Furthermore, the incident of a Clean Accuracy (CACC) decline for ELUDE was a mere 1.29%, placing it on par with the other methods tested. These outcomes highlight ELUDE's superior ability to thwart backdoor attacks, while simultaneously maintaining high model performance.

Effectiveness of Unlearning. At the heart of ELUDE lies an innovative method referred to as machine unlearning, which is deployed to eradicate the influence

caused by maliciously poisoned training data. This innovative approach greatly minimizes the effects of backdoor triggers, consequently causing the ASR to plummet by approximately 60%. We additionally argue that the implementation of Elastic Weight Consolidation (EWC) plays an instrumental role in curtailing the negative impacts of machine unlearning, such as catastrophic forgetting. By utilizing EWC, we succeeded in preserving or potentially even enhancing the model’s precision on unadulterated data.

6 Conclusion

In this paper, we introduced ELUDE, a novel defense mechanism against backdoor attacks in NLP models, leveraging machine unlearning complemented by Elastic Weight Consolidation. Our approach directly removes the influence of backdoor triggers from the model, significantly reducing attack success rates while preserving model performance on legitimate tasks. Experimental results demonstrate ELUDE’s superiority over existing methods like ONION and Back-Translation, especially against advanced attacks. By integrating EWC, ELUDE mitigates potential drawbacks of machine unlearning, such as catastrophic forgetting, ensuring the model’s effectiveness remains intact. ELUDE thus presents a robust solution to enhancing the security of NLP models against evolving backdoor threats, paving the way for further research into efficient and scalable defense mechanisms.

Acknowledgments. This work was financially supported by the National Natural Science Foundation of China (Grant No. 62472032, 62202051 and 62232002); the Open Project Funding of Key Laboratory of Mobile Application Innovation and Governance Technology, Ministry of Industry and Information Technology, (Grant No. 2023IFS080601-K); the Beijing Institute of Technology Research Fund Program for Young Scholars, and the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2023QNRC001).

References

1. Qi, F., Chen, Y., Li, M., Yao, Y., Liu, Z., Sun, M.: ONION: a simple and effective defense against textual backdoor attacks. In: EMNLP (1), pp. 9558–9566. Association for Computational Linguistics (2021)
2. Chen, X., et al.: BadNL: backdoor attacks against NLP models with semantic-preserving improvements. In: ACSAC, pp. 554–569. ACM (2021)
3. Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pre-trained models. CoRR, abs/2004.06660 (2020)
4. Yang, W., Li, L., Zhang, Z., Ren, X., Sun, X., He, B.: Be careful about poisoned word embeddings: exploring the vulnerability of the embedding layers in NLP models. In: NAACL-HLT, pp. 2048–2058. Association for Computational Linguistics (2021)
5. Yang, W., Lin, Y., Li, P., Zhou, J., Sun, X.: Rethinking stealthiness of backdoor attack against NLP models. In: ACL/IJCNLP (1), pp. 5543–5557. Association for Computational Linguistics (2021)

6. Qi, F., et al.: Hidden killer: invisible textual backdoor attacks with syntactic trigger. In: ACL/IJCNLP (1), pp. 443–453. Association for Computational Linguistics (2021)
7. Wang, J., et al.: Putting words into the system’s mouth: a targeted attack on neural machine translation using monolingual data poisoning. In: ACL/IJCNLP (Findings), volume ACL/IJCNLP 2021 of Findings of ACL, pp. 1463–1473. Association for Computational Linguistics (2021)
8. Fan, C., et al.: Defending against backdoor attacks in natural language generation. CoRR, abs/2106.01810 (2021)
9. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: IEEE Symposium on Security and Privacy, pp. 463–480. IEEE Computer Society (2015)
10. Bourtoule, L., et al.: Machine unlearning. In: IEEE Symposium on Security and Privacy, pp. 141–159. IEEE (2021)
11. Neel, S., Roth, A., Sharifi-Malvajerdi, S.: Descent-to-delete: gradient-based methods for machine unlearning. In: ALT. Proceedings of Machine Learning Research, vol. 132, pp. 931–962. PMLR (2021)
12. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y.: Graph unlearning. In: CCS, pp. 499–513. ACM (2022)
13. Liu, Y., Ma, Z., Liu, X., Wang, Z., Ma, S., Ren, K.: Revocable federated learning: a benchmark of federated forest. CoRR, abs/1911.03242 (2019)
14. Liu, Y., et al.: Backdoor defense with machine unlearning. In: INFOCOM, pp. 280–289. IEEE (2022)
15. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1), pp. 4171–4186. Association for Computational Linguistics (2019)
16. Gan, L., et al.: Triggerless backdoor attack for NLP tasks with clean labels. In: NAACL-HLT, pp. 2942–2952. Association for Computational Linguistics (2022)
17. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP, pp. 1631–1642. ACL (2013)
18. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Predicting the type and target of offensive posts in social media. In: NAACL-HLT (1), pp. 1415–1420. Association for Computational Linguistics (2019)
19. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS, pp. 649–657 (2015)
20. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: identifying vulnerabilities in the machine learning model supply chain. CoRR, abs/1708.06733 (2017)
21. Dai, J., Chen, C., Li, Y.: A backdoor attack against LSTM-based text classification systems. IEEE Access **7**, 138872–138878 (2019)
22. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. In: ICLR. OpenReview.net (2020)



Cross-Chain Transaction Auditing with Truth Discovery

Huishu Wu¹, Xuhao Ren², Mengxuan Liu², Tao Liu^{3(✉)}, Yajie Wang², Chuan Zhang², and Liehuang Zhu²

¹ China University of Political Science and Law, Beijing 100081, China

² Beijing Institute of Technology, Beijing 100081, China

³ China Academy of Information and Communications Technology, Beijing 100081, China

liutao3@caict.ac.cn

Abstract. In blockchain networks, cross-chain transaction audit is particularly important due to the wide variety of transactions between different chains, the large number of participants, and the unique rules and mechanisms of each chain. Scoring transactions can help buyers better understand the quality and credibility of transactions. However, the authenticity of the scores is not always reliable. In this paper, we propose a cross-chain transaction auditing scheme to address this problem. To be specific, this scheme deploys smart contracts on the relay-chain to query the seller's transaction scores on the chain and uses truth discovery technology to evaluate the seller's reputation. Since the smart contract is open and transparent, the credibility of the calculation process is ensured. In addition, considering the timeliness of the seller's reputation, we use Dirichlet distribution to predict the future reputation through the seller's historical reputation truth. Through experimental evaluation, we verified the effectiveness and reliability of the proposed scheme. In a local simulated experimental environment, the experimental results showed that the proposed scheme is feasible and efficient.

Keywords: Cross-chain · Transaction auditing · Truth discovery · Smart contract

1 Introduction

With the rapid development of blockchain technology, the complexity and diversity of transactions between different blockchain networks have brought new challenges to ensuring the security and reliability of cross-chain transactions [12, 33]. Since each blockchain network operates independently with its own set of rules and mechanisms, the audit process of cross-chain transactions is essential to maintain trust and transparency in a decentralized ecosystem. In this context, the need for effective cross-chain transaction audit has become increasingly important to detect potential fraud, ensure the integrity of transactions, and improve the overall credibility of blockchain networks [5, 36].

In cross-chain transactions, it has become a common practice for buyers to rate sellers [42]. This rating not only reflects the quality of the transaction but also helps

other buyers better understand the credibility and reliability of the seller. In cross-chain transaction audit, this rating plays a vital role. By comprehensively considering the buyer's rating of the seller, auditors can more accurately assess the transaction risk level and credibility of the seller, thereby effectively improving the blockchain's transaction environment [38,47]. However, due to the lack of transparency in the rating process and the fact that the rating results may be affected by vested interests, the authenticity, and accuracy of the buyer's rating of the seller are often questioned, making it difficult for auditors to fully trust these rating data. Therefore, in cross-chain transaction audits, how to ensure the authenticity and reliability of the rating has become a problem that needs to be solved.

Several cross-chain audit works [7,9,15,24] have been proposed. For example, Li et al. [15] proposed a cross-chain shared data consistency audit scheme that is secure, efficient, and protects data privacy, and for the first time realized the audit of data consistency among multiple blockchains. Guo et al. [9] introduced zkCross, a novel two-layer cross-chain architecture equipped with three cross-chain protocols, which realizes privacy-preserving cross-chain auditing, while solving the problems of cross-chain link exposure, privacy, and audit incompatibility, and overall audit efficiency. Fan et al. [7] introduced the relay-chain and adopted homomorphic signature technology and batch audit mechanism to achieve the secure sharing of medical Internet of Things data among heterogeneous blockchains, ensuring the consistency and integrity of the data. Liu et al. [24] proposed a cross-domain data security sharing access control model based on Hyperledger Fabric and attribute-based access control (ABAC), which implemented multi-level, fine-grained, access-controlled auditing and ensured data security through automatic permission verification. However, existing research mainly focuses on data privacy protection in cross-chain transaction audit, and cannot effectively address the problem of unreliable scoring that arises during cross-chain transaction audit.

In this paper, we propose cross-chain transaction auditing scheme. Specifically, we deployed a smart contract on the relay-chain to query the transaction logs of sellers on the alliance chain. These transaction logs record the seller's historical transaction behavior in detail, including important information such as the two parties to the transaction and the transaction score. In order to accurately evaluate the seller's reputation, we introduced the truth value discovery technology to calculate the true reputation from the buyer's transaction score of the seller. In addition, the seller's reputation may change over time and with changes in transaction behavior. We use the Dirichlet distribution to predict the seller's future reputation based on the seller's historical reputation truth value. Our contributions are as follows:

- We propose a cross-chain transaction audit scheme, which realizes cross-chain query and reputation calculation of user transactions through smart contracts on the relay-chain. The smart contract is open and transparent, ensuring the credibility of the query and calculation process.
- We calculate the seller's true reputation through truth discovery and use the Dirichlet distribution to predict new reputation values, which improves reliability of the audit.
- We evaluated the audit solution in terms of computational overhead and gas consumption. This paper experimentally verifies the effectiveness and feasibility of the scheme.

The paper's structure is as follows: Related works are presented in Sect. 2, followed by the preliminaries in Sect. 3. Section 4 covers the models and design goals. Our proposed scheme is detailed in Sect. 5. Section 6 discusses experimental results, and the paper is concluded in Sect. 7.

2 Related Works

In this section, we mainly introduce cross-chain technology and truth discovery related to our work.

Cross-Chain Schemes. The most advanced cross-chain schemes can be divided into two categories: chain-based and bridge-based.

Chain-based protocols, like sidechains [30] and hashed timelock contracts (HTLCs) [34], operate on the blockchain's inherent mechanisms without involving additional entities. A sidechain functions as an autonomous blockchain running parallel to the main blockchain. ZeroCross [22] introduces a privacy-preserving sidechain solution built on the Monero platform, while Zedoo incorporates auditing capabilities using zk-SNARK technology. Baldimtsi et al. [2] present a method for interconnecting anonymous chains. HTLC combines hash locks and time locks to facilitate cross-chain exchange activities. XCLAIM [46] relaxes some of the stringent assumptions of the original HTLC framework, such as requiring simultaneous online presence. Deshpande et al. [6] enhance HTLC with privacy features using Schnorr signatures. MAD-HTLC [34] leverages blockchain-based incentives to counter incentive manipulation attacks effectively. Cross-Channel [10] enhances the throughput of HTLC-based cross-chain systems, while Thyagarajan et al. [32] utilize verifiable timed signatures in place of time locks.

Bridge-based solutions, such as notarization schemes [45] and relay-chains [3], require an additional component to enable communication between chains. In notarization schemes, a trusted third-party entity known as a notary is responsible for validating cross-chain transactions. Although the notarization scheme is straightforward, it introduces a potential single point of failure. Yin et al. [45] improved the security of the notarization platform by incorporating secure hardware and encryption techniques. In the case of relay-chains, a dedicated blockchain network, the relay-chain, facilitates asset transfers among participating chains. zkBridge [39] introduced deVirgo to enhance the verification efficiency of cross-chain bridges. Wang et al. [35] suggested chain-by-chain governance, while BeDCV [48] employed technologies like homomorphic encryption and implemented a supervisory chain for decentralized auditing. It is important to note that bridge-based schemes integrate trusted mechanisms such as notarization and relay-chains, accommodating both cross-chain transfers and exchanges. Moreover, these schemes follow a star architecture, making them well-suited for auditing purposes.

Truth Discovery Schemes. Truth discovery, as a valuable technology in extracting accurate information from crowd-sensing systems, has gained significant interest in recent years [16, 17]. Notable truth discovery approaches such as AccuSim [18], CRH [17], and TruthFinder [44] have emerged. Unlike simplistic methods like the average or voting approaches, these algorithms offer more dependable aggregation outcomes by assessing user reliability and integrating them into the aggregation procedure.

Many methods for privacy-preserving truth discovery (PPTD) have been developed to safeguard data privacy. Current research primarily aims to strike a balance between security and efficiency in private TD processes. Miao et al. [25] introduced the initial PPTD framework for crowdsensing systems, utilizing the threshold Paillier cryptosystem. This approach necessitates multiple interactions between the server and data providers to ensure data confidentiality during TD, leading to resource-intensive operations for each data provider and overlooking the privacy of aggregation results. To address these drawbacks, several efficient PPTD solutions [31, 49] have been proposed and applied across various application scenarios, with notable schemes including [26, 41]. These advancements offer more efficient alternatives compared to the work of Miao et al. [25] by integrating technologies like data aggregation [50], homomorphic encryption [11], and secure multi-party computation [1]. However, many of these methods require constant online presence from all users to decrypt the final aggregate value successfully, rendering them susceptible to attacks if user communication is compromised. Recent studies [20, 21] have implemented PPTD with superior computational performance by leveraging differential privacy technology, enabling users to introduce random noise before data submission.

In future work, considering that the seller does not want to disclose his rating information to external entities, we can also use privacy-preserving truth discovery in cross-chain auditing. The buyer's rating of the seller is sent to the relay-chain in the form of ciphertext, and then the truth discovery under the ciphertext is performed, returning only the final true reputation value without disclosing any other information related to the seller.

3 Preliminaries

In this section, we mainly discuss some of the pre-knowledge in this scheme, including blockchain, cross-chain technology, truth discovery, and Dirichlet Distribution.

3.1 Blockchain

Blockchain is a decentralized distributed ledger technology that was first known for Bitcoin, but its applications have expanded to many fields, including finance [29], supply chain management [28], and healthcare [23]. The core features of blockchain include decentralization, transparency, and immutability. Each block contains a timestamp, a set of transaction data, and a hash value of the previous block. This structure ensures the integrity and security of the data. Based on the permissions and management methods of the participating nodes, it can be divided into the following three main types:

1. **Public Blockchain.** A public chain is a completely decentralized blockchain where anyone can participate and access the network, and its transaction records are open and transparent to everyone. The network is secure and tamper-proof through consensus mechanisms such as proof of work (PoW) or proof of stake (PoS). Public chains are highly public and anyone can join and exit freely. Common examples include Bitcoin and Ethereum.
2. **Private Blockchain.** A private chain is a blockchain controlled by a single organization or institution, where only authorized nodes can participate and access the network. It provides higher privacy protection, and transaction records and data are only visible to authorized users. Due to fewer participating nodes, private chains have fast transaction processing speed and low energy consumption, making them suitable for internal management systems of enterprises or internal settlement systems of banks and financial institutions.
3. **Consortium Blockchain.** A consortium chain is a blockchain jointly managed by multiple organizations or institutions. Each participant needs permission to join the network. Usually, an efficient consensus mechanism is adopted, such as Byzantine Fault Tolerance (PBFT) or Delegated Proof of Stake (DPoS), to ensure joint decision-making by multiple participants. The consortium chain finds a balance between privacy and transparency. It can share transaction records among consortium members while hiding some information from the outside. It is suitable for supply chain management between enterprises, joint credit reporting systems in the financial industry, and joint data management systems in the medical industry. Therefore, we consider transaction auditing across consortium chains in this paper.

3.2 Cross-Chain Technology

With the development of blockchain technology, interoperability between different block-chains has become an important issue. Cross-chain technology aims to achieve data exchange and asset transfer between different blockchains, break the “information island”, and promote the development of a wider blockchain ecosystem. Common cross-chain tools include:

1. **Polkadot:** Polkadot [37] is a cross-chain protocol developed by the Web3 Foundation that aims to enable interoperability between different blockchains. Polkadot uses a relay-chain and parachain architecture to coordinate communication and data exchange between different blockchains through the relay-chain. The relay-chain is responsible for the security and consensus of the entire network, while the parachain can run independently and communicate with the relay-chain. This architecture ensures high scalability and shared security while allowing flexible cross-chain communication.
2. **Cosmos:** Cosmos [14] is a cross-chain ecosystem developed by the Tendermint team, which aims to create a network of independent blockchains. Cosmos achieves interoperability between blockchains through the Tendermint consensus protocol and the IBC (Inter-Blockchain Communication) protocol. Tendermint provides a high-performance consensus mechanism, while the IBC protocol enables secure messaging and data exchange between different blockchains. This approach enables

Cosmos to connect various heterogeneous blockchains to form an interconnected blockchain network.

3. **Wanchain:** Wanchain [27] is a platform dedicated to cross-chain transactions of assets and data between different blockchains. Wanchain achieves cross-chain asset transfers through a lock account mechanism and distributed key generation technology. Specifically, Wanchain creates lock accounts on the source and target chains and uses multi-party secure computing (MPC) to generate distributed keys to ensure the security and decentralization of cross-chain transfers. This process is implemented through a cross-chain bridge, which ensures the circulation of multiple assets between different blockchains.
4. **BitXHub:** BitXHub [43] is a cross-chain platform developed by QuChain Technology, which aims to achieve interoperability between different blockchains. BitXHub adopts a multi-layer architecture design, realizing cross-chain communication and data exchange through the relay-chain and the cross-chain gateway. As the core component, the relay-chain is responsible for coordinating and verifying cross-chain transactions to ensure the security and consistency of cross-chain operations. The cross-chain gateway serves as an interface to connect different blockchain networks and supports a variety of cross-chain operations, including asset transfer, smart contract calls, and data sharing. The design of BitXHub enables it to efficiently process cross-chain transactions and has good scalability, making it suitable for a variety of cross-chain application scenarios.

3.3 Truth Discovery

Truth discovery is a valuable technology that aims to address conflicts in noisy data and determine the credibility of users based on the information they provide. Previous studies [19, 25, 49] have highlighted its benefits across different contexts. The process of truth discovery typically begins by establishing the initial truth for each task, followed by iterative updates to refine the weights and truths until convergence is achieved.

Weight Update. Given that the original ground truth remains constant, this stage involves adjusting the user weights. When the information supplied by the user aligns more closely with the previously determined true value, the user's weight is increased; conversely, it is decreased when there is a mismatch. Typically, the user's weight is computed using the following method:

$$w_i = \log \left(\frac{\sum_{i=1}^M \sum_{k=1}^K d(x_k^i, x_k^*)}{\sum_{k=1}^K d(x_k^i, x_k^*)} \right). \quad (1)$$

where x_k^i denotes the sensor data provided by the user i for the k_{th} task, and x_k^* represents the current estimated actual value of the k_{th} task. The function $d(\cdot)$ is utilized as a distance metric to quantify the variance between user data and the true value. In our work, we use the following distance function:

$$d(x_k^i, x_k^*) = \frac{(x_k^i - x_k^*)^2}{std_k}, \quad (2)$$

where std_k denotes the standard deviation for object k across all users.

Truth Discovery. By determining the weight of each user in the preceding stage, the actual value of task k can be updated as:

$$x_k^* = \frac{\sum_{i=1}^M w_i \cdot x_k^i}{\sum_{i=1}^M w_i}. \quad (3)$$

In the above formula, M represents the total number of users, x_k^i represents the perception data submitted by u_i for the k_{th} task, w_i represents the weight of user u_i , and x_k^* represents the new estimated truth.

3.4 Dirichlet Distribution Algorithm

The Dirichlet distribution comprises a collection of continuous multivariate probability distributions characterized by a prior parameter vector ξ [8, 40]. In the case of binary state spaces, it is also influenced by the Beta distribution [13]. This distribution is frequently employed to depict the probability distribution across T-dimensional random variables $\mathbf{X} = \{X_1, X_2, \dots, X_T\}$, where T represents the number of potential outcomes and X_i denotes the value of the i_{th} outcome. To begin, a level vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_T\}$ is established, with $\theta \in [0, 1]$ and $\theta_{i-1} < \theta_i$. Subsequently, the probability distribution vector of \mathbf{X} is denoted by $\mathbf{p} = \{p_1, p_2, \dots, p_T\}$, where $p_i = P\{\theta_{i-1} < X_i \leq \theta_i\}$, with $1 \leq i \leq T$. Through the capture of the observation sequence of k feasible outcomes, the Dirichlet distribution generates prior parameters $\xi = \{\xi_1, \xi_2, \dots, \xi_T\}$ to portray the cumulative observation vector, where $\xi_i > 0$. Consequently, the probability density function (PDF) can be articulated as

$$f(\mathbf{p}|\xi) = \frac{\Gamma(\sum_{i=1}^T \xi_i) \times \prod_{i=1}^T p_i^{\xi_i - 1}}{\sum_{i=1}^T \Gamma(\xi_i)}, \quad (4)$$

where $\Gamma(\cdot)$ represents the Gamma function. Then, we can calculate the expected value of the Dirichlet distribution as

$$E(p_i|\xi_i) = \frac{\xi_i}{\sum_{i=1}^T \xi_i}. \quad (5)$$

4 Overview

This section mainly introduces the system model, workflow, threat model, and design goals of the scheme. For ease of understanding, we give the symbols in Table 1.

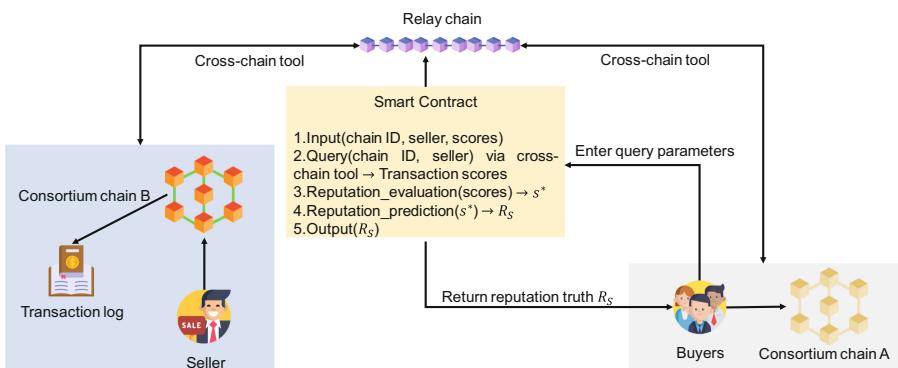
4.1 System Model

As shown in Fig. 1, the scheme in this article mainly involves the following entities: the buyers, the seller, the consortium chain, and relay-chain. In this paper, we assume that the buyer is on chain A and the seller is on chain B. However, in actual applications, the buyer can also be on chain B and the seller can also be on chain A.

Table 1. Notations used in our scheme

Notation	Description
M	The number of buyers
t	Time period, used to divide the transaction datasets
s^i	Buyer i 's score for seller
s^*	Seller's truth reputation
w_i	Buyer i 's weight
std_m	The standard deviation for buyers' scores
ξ	The prior parameter
p	Probability distribution vector
β	Reputation level vector
q_i	Weight for each β_i
R_S	Reputation score of the seller

1. Consortium chain: Consortium chain is a permissioned chain that stores all transaction logs of its users. Users outside the consortium chain cannot directly access the consortium chain to obtain the data information stored on it.
2. Relay-chain: The relay-chain is a public, non-permissioned chain that can access the ledger of the consortium chain through the cross-chain gateway maintained by the consortium chain and retrieve the stored information.
3. Buyers: Some users on consortium chain A, after completing a transaction, rate the seller based on the quality of the transaction, and can send a request to the relay-chain to evaluate the reputation of the seller on consortium chain B.
4. Seller: Whenever a user on consortium chain B completes a transaction, he or she records the transaction information in the transaction log on chain B.

**Fig. 1.** System model.

4.2 Workflow

When the two parties on different consortium chains complete the transaction, the buyer scores the seller based on the quality of the transaction, and then the transaction score, seller address, and other information are recorded on the seller's chain. After a period of time, the relay-chain retrieves all the transaction scores of the seller during this period from the chain by calling the smart contract, and then uses the truth discovery technology to evaluate the seller's reputation. Finally, using the Dirichlet distribution, it predicts the seller's future reputation value. This predicted value helps the platform evaluate and verify the credibility of the seller, and can also serve as a basis for subsequent buyers to decide whether to trade with the seller.

4.3 Threat Model

In a threat model, we define the possible behaviors of each entity and consider possible attacks against the solution and their purpose.

1. The buyers are also considered to be potentially malicious and may attempt to maliciously reduce the credibility of the seller during the audit process, causing it to suffer a crisis of trust.
2. The relay-chain is defined as a trusted party. Users on the relay-chain can obtain transaction information stored on the consortium chain through cross-chain technology. Buyers and seller can freely access the relay-chain.
3. Consortium chain is defined as a trustworthy party, and the transaction information stored on the chain is completely correct.

We assume that none of the above parties will do anything that would harm their interests. In addition, we assume that the smart contracts on the relay-chain are running correctly and that the transaction information on the consortium chain and the relay-chain is accurately recorded.

4.4 Design Goals

Based on the above model, we aim to design a transaction audit scheme for cross-chain consortium chains, which must meet the following design goals.

- Accessible. Buyers on the consortium chain can access sellers on another chain through the relay-chain.
- Privacy preservation. The seller's transaction information will not be leaked to entities outside the consortium chain during the audit process and will be used correctly for auditing.
- Timeliness. The cross-chain transaction audit scheme should calculate the reputation from the latest transaction records of the seller. The reputation should be updated as the number of transactions increases and over time.

5 Transaction Auditing Scheme

This section introduces the details of cross-chain transaction auditing, which mainly includes three phases: pre-processing, reputation evaluation, and reputation prediction. We assume that each consortium chain has a transaction for each account, which includes the transaction record of the account and the score of each transaction. The complete workflow is shown in Fig. 2.

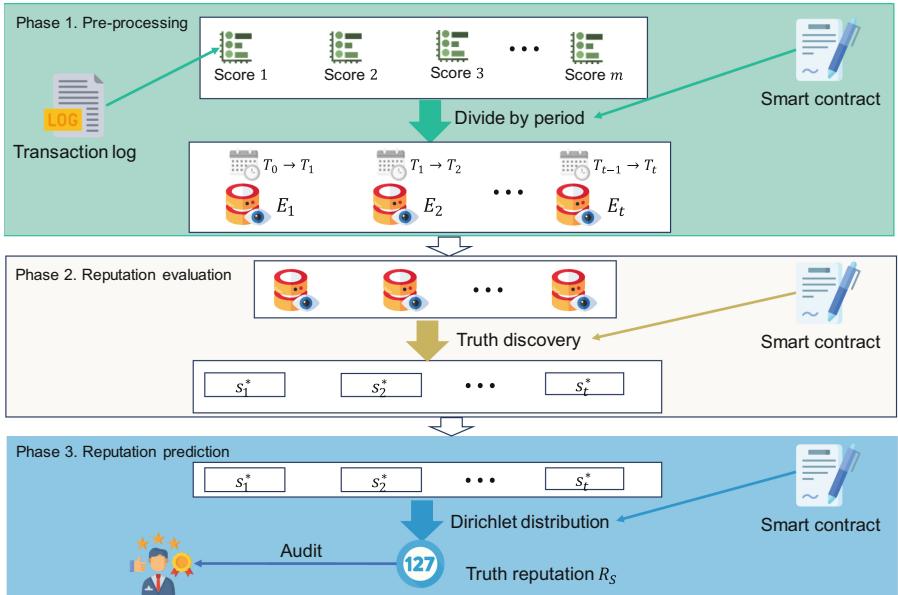


Fig. 2. The workflow of our scheme.

5.1 Pre-processing

The complete algorithm of this phase is shown in Algorithm 1. The specific process is as follows.

Step 1: Data Collection. The audit process begins when the buyers submit a request to the relay-chain, specifying the chain ID and the account address of the seller. This initiates the retrieval of all transaction records related to the seller on the specified chain ID. The buyers accomplish this by invoking the `Query` function of the smart contract, which returns a comprehensive list of transactions denoted as $\{T_1, T_2, T_3, \dots, T_n\}$. These transactions serve as the foundation for the audit analysis, providing crucial insights into the financial activities and interactions of the seller within the blockchain.

Step 2: Data Partitioning. Following the acquisition of the transaction records in **Step 1**, the next crucial phase involves the segmentation of the obtained dataset into distinct time periods. For instance, if the transaction history spans over a period of M days, we implement a systematic division strategy to categorize the data into smaller intervals. By effectively partitioning the records, such as grouping them into segments denoted as $\{E_1, E_2, \dots, E_t\}$ based on a M/t -day timeframe, we can enhance the granularity of the analysis and facilitate a more structured examination of the seller's financial activities over different temporal scopes.

Algorithm 1. Pre-processing

Input: Chain ID, seller account address

Output: E_1, E_2, \dots, E_t

- 1: Call the Query function in the smart contract to retrieve all transaction scores T_1, T_2, \dots, T_M for the seller
 - 2: Divide the scores into time periods, with each period spanning M/t days, resulting in E_1, E_1, \dots, E_t
 - 3: **return** E_1, E_2, \dots, E_t
-

5.2 Reputation Evaluation

In this phase, we will use the smart contract to automatically perform reputation evaluation. For convenience, we take a data set of a period of time as an example, assuming that the dataset includes m scores from m buyers, i.e. s_1, s_2, \dots, s_m . The following is a detailed introduction. See Algorithm 2 for the complete algorithm.

Algorithm 2. Reputation evaluation process

Input: m scores from buyers: $\{s^i\}_{i=1}^{i=m}$

Output: Estimated truths s^*

- 1: Sending estimated truths s^* to each buyer by random initialization.
 - 2: **repeat**
 - 3: **for** $i = 1$ to m **do**
 - 4: Weight Update based on the Eq.(6)
 - 5: **end for**
 - 6: Reputation evaluation based on the Eq.(7)
 - 7: **until** the convergence condition is satisfied
 - 8: **return** s^*
-

Step 1: Weight Update. Based on the transaction records scores s_1, s_2, \dots, s_m obtained in the previous step, this step will update the weight of each buyer. By combining the transaction record scores for each period and updating the buyers' weight based

on the consistency of the buyers' score with the estimated truth value, we can more carefully evaluate the performance of each buyer in different periods. This weight update scheme helps to improve the accuracy and credibility of the audit process, ensuring that buyers who provide accurate scores play a greater role in the audit while reducing the impact of buyers whose scores deviate from the results. In this scheme, the formula for calculating the buyers' weight w_i is as follows:

$$w_i = \log \left(\frac{\sum_{i=1}^m d(s^i, s^*)}{d(s^i, s^*)} \right), \quad (6)$$

where m denotes the total number of buyers.

For the distance function $d(\cdot)$, the main thing is to calculate std_m . When receiving the transaction score from the buyers, first calculate $sum = \sum_{i=1}^m s_i$ and $s_{avg} = (sum/m)$. Then $d_i = (s_i - s_{avg})^2$ is calculated. After receiving all d_i of each buyers, calculating $sum_d = \sum_{i=1}^m d_i$, and finally get $std_m = \sqrt{sum_d/m}$.

By dynamically adjusting the weights of buyers, we are able to place greater emphasis on buyers who provide accurate and reliable data during the audit process, thereby improving the accuracy and effectiveness of the overall audit.

Step 2: Reputation Evaluation. After all weights are updated, the reputation of the seller is updated. In this step, the formula for calculating the seller's reputation s^* is as follows:

$$s^* = \frac{\sum_{i=1}^m w_i \cdot s^i}{\sum_{i=1}^m w_i}. \quad (7)$$

In this formula, s^* represents the final reputation value of the seller. The comprehensive reputation score of the seller can be obtained by taking the weighted average of each buyer's weight w_i and its corresponding score s^i . The weight w_i represents the importance of each buyer in the audit process. The higher the weight, the greater the influence of the buyers in determining the reputation of the seller.

By using this reputation calculation formula, combined with the updated weight information, we can more accurately assess the overall credibility of the seller. This reputation update scheme based on buyers' weights and scores helps to ensure the objectivity and accuracy of the seller's reputation assessment and provides a reliable basis for subsequent audit results and decisions.

5.3 Reputation Prediction

In this phase, smart contracts will be used to automatically perform reputation prediction. Reputation prediction mainly includes reputation normalization, reputation aggregation, and reputation evaluation.

To improve the assessment of the seller's historical reputation, we employ reputation prediction using the Dirichlet distribution. Within the auditing process, we normalize the reputation and utilize the Dirichlet distribution to forecast the seller's future reputation by leveraging the consistently standardized historical reputation. First, reputation normalization is performed to normalize the reputation from 0 to 1. Next, reputation aggregation is performed using the accumulated historical data vector and the

posterior Dirichlet distribution to obtain the probability distribution \mathbf{p} . Then Y is set to the weighted average of \mathbf{p} , and the expected value $E[Y]$ is used as the predicted reputation. Detailed algorithm is seen as Algorithm 3.

Algorithm 3. Reputation Prediction

```

Input:  $\{a_i\}_{i=1}^t, \{\xi\}$ 
1: for  $i \in [1, t]$  do
2:   //Reputation normalization
3:    $a_i = \frac{s_i - \min(\{s_i\}_{i=1}^t)}{\max(\{s_i\}_{i=1}^t) - \min(\{s_i\}_{i=1}^t)}$ ;
4: end for
5: //Reputation aggregation
6: set  $\mathbf{Y} \leftarrow \{Y_1, Y_2, \dots, Y_C\}$ ;
7: set  $\beta \leftarrow \{\beta_1, \beta_2, \dots, \beta_C\}$ , ( $\beta_i \in (0, 1], i \in [1, C], \beta_i \leq \beta_{i+1}$ );
8: set  $p_i = P\{\beta_{i-1} < Y_i \leq \beta_i\}, (i = 1, 2, \dots, C)$ ;
9: set  $\mathbf{p} \leftarrow \{p_1, p_2, \dots, p_C\}, (\sum_{i=1}^C p_i = 1)$ ;
10: set  $\xi \leftarrow \{\xi_1, \xi_2, \dots, \xi_C\}$ ;
11: compute  $f(\mathbf{p}|\xi) = D(\mathbf{p}|\xi) = \frac{\Gamma(\sum_{i=1}^C \xi_i)}{\prod_{i=1}^C \Gamma(\xi_i)} \prod_{i=1}^C p_i^{\xi_i - 1}$ ;
12: //Reputation evaluation
13: set  $\xi_0 = \sum_{i=1}^C \xi_i$ ; set  $\mathbf{q} \leftarrow \{q_1, q_2, \dots, q_C\}$ ;
14: compute  $E[Y] = \sum_{i=1}^C q_i E[p_i] = \frac{\sum_{i=1}^C q_i \xi_i}{\xi_0}$ 
15: return  $\{E[Y]\}$ 

```

Step 1: Normalization. We normalize the reputation using Eq. 8:

$$a_i = \frac{s_i - \min(\{s_i\}_{i=1}^t)}{\max(\{s_i\}_{i=1}^t) - \min(\{s_i\}_{i=1}^t)}, \quad (8)$$

where s_i denotes the reputation of the seller in each period.

Step 2: Reputation Aggregation. For a particular seller AP , let $Y(0 \leq Y \leq 1)$ represents continuous random variable indicating the truth of AP . Referring to C levels of truth as a set $\{\beta_1, \beta_2, \dots, \beta_C\}$ ($\beta_i \in (0, 1], i = 1, 2, \dots, C, \beta_i < \beta_{i+1}$). The probability distribution vector of Y with respect to these C -levels is denoted as $\mathbf{p} = \{p_1, p_2, \dots, p_C\} (\sum_{i=1}^C p_i = 1)$, where $P\{\beta_{i-1} < Y_i < \beta_i\} = p_i (i = 1, 2, \dots, C)$. We let $\xi = \{\xi_1, \xi_2, \dots, \xi_C\}$ represent the vector of cumulative truth value. With a posterior Dirichlet distribution, \mathbf{p} can be modeled as

$$D(\mathbf{p}|\xi) = \frac{\Gamma(\sum_{i=1}^C \xi_i)}{\sum_{i=1}^C \Gamma(\xi_i)} \prod_{i=1}^C p_i^{\xi_i - 1}, \quad (9)$$

where $\xi_0 = \sum_{i=1}^C \xi_i$.

Step 3: Reputation Evaluation. To calculate the reputation score of the seller, we assign a weight value q_i to each level $\beta_i (i \in [1, C])$. Let p_i represent the probability that the true value of AP is classified into the β_i level, where $\mathbf{p} = \{p_1, p_2, \dots, p_C\} (\sum_{i=1}^C p_i = 1)$. Let Y be a random variable representing the weighted average of the probability of the true value in \mathbf{p} , then the reputation score R_S of seller can be calculated as

$$R_S = E[Y] = \sum_{i=1}^C q_i E[p_i] = \frac{1}{\xi_0} \sum_{i=1}^C q_i \xi_i, \quad (10)$$

where ξ_i is the accumulation value of levels that the seller's truth values belong to.

Finally, the blockchain platform can audit the seller's credibility through the predicted reputation value R_S , and also facilitate subsequent buyers to decide whether to trade with the seller based on the credibility.

6 Experiments

6.1 Experimental Settings

The experiments in this paper were developed on a local host. Two Ethereum [4] private chains were built locally, and the BitXHub solution was used to implement cross-chain communication. Of the two blockchains built, one simulated the consortium chain A; the other formed a relay system with the BitXHub relay-chain, realizing the deployment and calling of the smart contracts required by the solution. Table 2 shows the overall experimental environment of this paper, including the hardware configuration and software environment of the host.

Table 2. Experimental environment

Experimental environment	
CPU	Intel(R) Core(TM) i5-8300H CPU
RAM	8G
OS	64 bits Ubuntu 20.04
Programming language	<i>Solidity</i> ^{0.8.0}
Compiler	Remix

6.2 Performance Evaluation

In this part, we mainly evaluate the computational overhead of the audit scheme and the gas overhead of the smart contract. The main test objects are the three algorithms of the audit scheme: Pre-processing (PP), Reputation Evaluation (RE), and Reputation Prediction (RP).

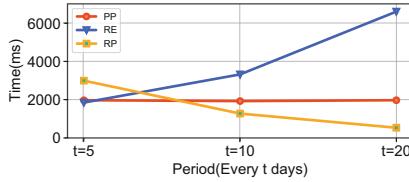


Fig. 3. The running time of each algorithm.

Computation Costs. First, we analyze the computation overhead of the solution in this paper. Since cross-chain operations and smart contract deployment and calling are all implemented on the local host, which is quite different from the actual environment, we only measured the time overhead of the solution. The average time overhead of each stage of the audit scheme is shown, and Fig. 3 shows the changes in the total time overhead of each algorithm divided by different periods.

As can be seen from Fig. 3, the running time of the PP algorithm is basically not affected by the partitioning method. This is because the preprocessing time is mainly affected by the total number of transaction scores, but the total number of transaction scores has not changed in the experiment. The running time of the RE algorithm increases with the increase in the number of partition days. This is because when the data is partitioned according to time periods, the longer the interval, the larger each data set, and the longer the processing time of the RE algorithm each time. The running time of the RP algorithm decreases with the increase in the number of partition days. This is because the reputation prediction algorithm is mainly related to the number of historical true values. As the interval time becomes longer, the number of historical true values that can be obtained decreases, so the running time also decreases.

From an overall perspective, the computational overhead of the audit scheme proposed in this paper is relatively small. Without taking into account the delay of the cross-chain network and the deployment and call of smart contracts, the single-run time overhead is about 6000ms, which mainly depends on the size of the transaction number and the interval time of the division. Compared with the network delay in the actual cross-chain scenario and the block confirmation time overhead on the relay-chain (for security reasons, transactions in Ethereum usually need to wait for about 12 blocks to be confirmed, each block interval is about 15 s, a total of about 180 s), the single-run time overhead of our cross-chain transaction audit scheme is much smaller.

Gas Overhead. Next, we measured the gas costs of the smart contract deployment in the scheme. The smart contract includes three algorithms, namely PP, RE, and RP. We wrote the contract in Solidity, compiled it through Remix, and deployed and tested it on the local relay system we built. The results are shown in Fig. 4.

As shown in Fig. 4, deploying three algorithms on the smart contract consumes an average of about 2.7×10^6 gas. According to the gas price (12 gwei/ per gas) proposed on Ethereum Scan in June 2024, the cost of deploying the contract is about 3.3×10^{-2} eth. When calling it later, the gas consumption drops significantly.

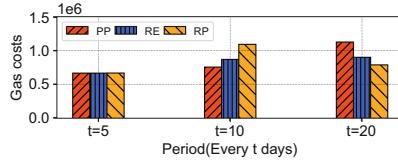


Fig. 4. Gas cost of smart contract deploys.

Finally, we measured the gas costs for the smart contract calls in the scheme. The experimental method and parameters are the same as before. The results are presented in Fig. 5.

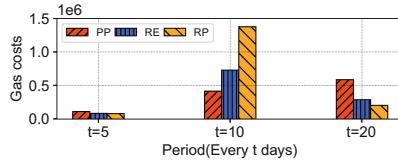


Fig. 5. Gas cost of smart contract calls.

As can be seen from Fig. 5, among the three algorithms, the pre-processing algorithm consumes the least gas. By analyzing the contract logic, the pre-processing algorithm only realizes the division of the transaction data set by time period, so the gas consumption is low; the reputation calculation algorithm needs to perform true discovery, and for the divided data, it needs to perform the same number of addition operations, division operations, and one logarithmic operation as the number of scores, so the gas consumption is relatively large. The reputation prediction algorithm needs to transmit more calculation parameters and perform multiple integral operations, so the gas consumption is the largest.

7 Conclusion

This paper proposes a cross-chain transaction audit scheme, focusing on the importance of seller reputation scores in cross-chain transaction audit. By deploying smart contracts on the relay-chain and using truth discovery technology to evaluate seller reputation, a comprehensive audit of seller reputation is achieved. Due to the openness and transparency of smart contracts, the credibility of the calculation process is guaranteed. At the same time, considering the timeliness of the seller's reputation, the Dirichlet distribution is used to predict future reputation based on the seller's historical reputation truth value. Through experimental verification, we verified the effectiveness and reliability of the proposed scheme. In a local simulation experimental environment, the experimental results show that the proposed scheme is feasible and efficient.

In the future, we will continue to explore more audit methods based on blockchain technology to cope with the many challenges in cross-chain transactions. We will also continue to improve the proposed audit scheme to improve its performance and applicability and conduct more extensive verification in actual application environments.

Acknowledgments. This work was financially supported by the “National Key R&D Program of China” (2021YFB2700500, 2021YFB2700503).

References

1. Agrawal, S., Kitagawa, F., Nishimaki, R., Yamada, S., Yamakawa, T.: Public key encryption with secure key leasing. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 581–610. Springer, Cham (2023)
2. Baldimtsi, F., Miers, I., Zhang, X.: Anonymous sidechains. In: International Workshop on Data Privacy Management, pp. 262–277. Springer, Cham (2021)
3. Belchior, R., Vasconcelos, A., Guerreiro, S., Correia, M.: A survey on blockchain interoperability: past, present, and future trends. *ACM Comput. Surv. (CSUR)* **54**(8), 1–41 (2021)
4. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. *White Paper* **3**(37), 2–1 (2014)
5. Chen, J., Yao, S., Yuan, Q., He, K., Ji, S., Du, R.: Certchain: public and efficient certificate audit based on blockchain for TLS connections. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 2060–2068. IEEE (2018)
6. Deshpande, A., Herlihy, M.: Privacy-preserving cross-chain atomic swaps. In: International Conference on Financial Cryptography and Data Security, pp. 540–549. Springer, Cham (2020)
7. Fan, K., Liu, Z., Liu, M., Wen, Y., Lu, N., Shi, W.: Cross-chain data auditing for medical IoT data sharing. In: International Conference on Security and Privacy in New Computing Environments, pp. 65–81. Springer, Cham (2022)
8. Fung, C.J., Zhang, J., Aib, I., Boutaba, R.: Dirichlet-based trust management for effective collaborative intrusion detection networks. *IEEE Trans. Netw. Serv. Manage.* **8**(2), 79–91 (2011)
9. Guo, Y., et al.: zkcross: a novel architecture for cross-chain privacy-preserving auditing. *Cryptology ePrint Archive* (2024)
10. Guo, Y., Xu, M., Yu, D., Yu, Y., Ranjan, R., Cheng, X.: Cross-channel: scalable off-chain channels supporting fair and atomic cross-chain operations. *IEEE Trans. Comput.* (2023)
11. Henzinger, A., Hong, M.M., Corrigan-Gibbs, H., Meiklejohn, S., Vaikuntanathan, V.: One server for the price of two: simple and fast {Single-Server} private information retrieval. In: 32nd USENIX Security Symposium (USENIX Security 2023), pp. 3889–3905 (2023)
12. Herlihy, M.: Atomic cross-chain swaps. In: Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, pp. 245–254 (2018)
13. Josang, A., Haller, J.: Dirichlet reputation systems. In: The Second International Conference on Availability, Reliability and Security (ARES 2007), pp. 112–119. IEEE (2007)
14. Kwon, J., Buchman, E.: Cosmos whitepaper. *A Netw. Distrib. Ledgers* **27**, 1–32 (2019)
15. Li, D., Ding, P., Zhou, Y., Yang, Y., Li, C.: Secure, efficient, and privacy-protecting one-to-many cross-chain shared data consistency audit. In: 2023 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 64–71. IEEE (2023)

16. Li, Q., et al.: A confidence-aware approach for truth discovery on long-tail data. *Proc. VLDB Endow.* **8**(4), 425–436 (2014)
17. Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W., Han, J.: Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1187–1198 (2014)
18. Li, X., Dong, X.L., Lyons, K., Meng, W., Srivastava, D.: Truth finding on the deep web: is the problem solved? *arXiv preprint arXiv:1503.00303* (2015)
19. Li, Y., et al.: A survey on truth discovery. *ACM SIGKDD Explor. Newsl.* **17**(2), 1–16 (2016)
20. Li, Y., et al.: An efficient two-layer mechanism for privacy-preserving truth discovery. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1705–1714 (2018)
21. Li, Y., et al.: Towards differentially private truth discovery for crowd sensing systems. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1156–1166. IEEE (2020)
22. Li, Y., et al.: Zerocross: a sidechain-based privacy-preserving cross-chain solution for monero. *J. Parallel Distrib. Comput.* **169**, 301–316 (2022)
23. Liu, P.T.S.: Medical record system using blockchain, big data and tokenization. In: Lam, K.-Y., Chi, C.-H., Qing, S. (eds.) *ICICS 2016. LNCS*, vol. 9977, pp. 254–261. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50011-9_20
24. Liu, Y., Yang, W., Wang, Y., Liu, Y.: An access control model for data security sharing cross-domain in consortium blockchain. *IET Blockchain* **3**(1), 18–34 (2023)
25. Miao, C., et al.: Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 183–196 (2015)
26. Miao, C., Su, L., Jiang, W., Li, Y., Tian, M.: A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9. IEEE (2017)
27. Ou, W., Huang, S., Zheng, J., Zhang, Q., Zeng, G., Han, W.: An overview on cross-chain: mechanism, platforms, challenges and advances. *Comput. Netw.* **218**, 109378 (2022)
28. Rejeb, A., Rejeb, K., Simske, S., Keogh, J.G.: Exploring blockchain research in supply chain management: a latent dirichlet allocation-driven systematic review. *Information* **14**(10), 557 (2023)
29. Schär, F.: Decentralized finance: on blockchain-and smart contract-based financial markets. *FRB of St. Louis Review* (2021)
30. Singh, A., Click, K., Parizi, R.M., Zhang, Q., Dehghanianha, A., Choo, K.K.R.: Sidechain technologies in blockchain networks: an examination and state-of-the-art review. *J. Netw. Comput. Appl.* **149**, 102471 (2020)
31. Tang, X., Wang, C., Yuan, X., Wang, Q.: Non-interactive privacy-preserving truth discovery in crowd sensing applications. In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1988–1996. IEEE (2018)
32. Thyagarajan, S.A., Malavolta, G., Moreno-Sanchez, P.: Universal atomic swaps: secure exchange of coins across all blockchains. In: *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1299–1316. IEEE (2022)
33. Tian, H., et al.: Enabling cross-chain transactions: a decentralized cryptocurrency exchange protocol. *IEEE Trans. Inf. Forensics Secur.* **16**, 3928–3941 (2021)
34. Tsabary, I., Yechieli, M., Manuskin, A., Eyal, I.: MAD-HTLC: because HTLC is crazy-cheap to attack. In: *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1230–1248. IEEE (2021)

35. Wang, X., Qiu, W., Zeng, L., Wang, H., Yao, Y., He, D.: A supervisory and governance mechanism for power master-slave chain architecture. In: 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI), pp. 172–175. IEEE (2021)
36. Wang, Z., Lin, J., Cai, Q., Wang, Q., Zha, D., Jing, J.: Blockchain-based certificate transparency and revocation transparency. *IEEE Trans. Dependable Secure Comput.* **19**(1), 681–697 (2020)
37. Wood, G.: Polkadot: vision for a heterogeneous multi-chain framework. White Paper **21**(2327), 4662 (2016)
38. Wu, Z., Liu, J., Wu, J., Zheng, Z., Chen, T.: Tracer: scalable graph-based transaction tracing for account-based blockchain trading systems. *IEEE Trans. Inf. Forensics Secur.* (2023). <https://doi.org/10.1109/TIFS.2023.3266162>
39. Xie, T., et al.: zkbridge: trustless cross-chain bridges made practical. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 3003–3017 (2022)
40. Xu, C., Wang, J., Zhu, L., Zhang, C., Sharif, K.: PPMR: a privacy-preserving online medical service recommendation scheme in ehealthcare system. *IEEE Internet Things J.* **6**(3), 5665–5673 (2019)
41. Xu, G., Li, H., Liu, S., Wen, M., Lu, R.: Efficient and privacy-preserving truth discovery in mobile crowd sensing systems. *IEEE Trans. Veh. Technol.* **68**(4), 3854–3865 (2019)
42. Yang, Y., Guan, Z., Wan, Z., Weng, J., Pang, H.H., Deng, R.H.: Priscore: blockchain-based self-tallying election system supporting score voting. *IEEE Trans. Inf. Forensics Secur.* **16**, 4705–4720 (2021)
43. Ye, S.J., Wang, X.Y., Xu, C.C., Sun, J.L.: Bitxhub: side-relay chain based heterogeneous blockchain interoperable platform. *Comput. Sci.* **47**(6), 294–302 (2020)
44. Yin, S., Zhu, P., Wu, L., Gao, C., Wang, Z.: Gamc: an unsupervised method for fake news detection using graph autoencoder with masking. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 347–355 (2024)
45. Yin, Z., Zhang, B., Xu, J., Lu, K., Ren, K.: Bool network: an open, distributed, secure cross-chain notary platform. *IEEE Trans. Inf. Forensics Secur.* **17**, 3465–3478 (2022)
46. Zamyatin, A., Harz, D., Lind, J., Panayiotou, P., Gervais, A., Knottenbelt, W.: Xclaim: trustless, interoperable, cryptocurrency-backed assets. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 193–210. IEEE (2019)
47. Zhang, C., Zhao, M., Liang, J., Fan, Q., Zhu, L., Guo, S.: Nano: cryptographic enforcement of readability and editability governance in blockchain database. *IEEE Trans. Dependable Secure Comput.* (Early Access) (2023). <https://doi.org/10.1109/TDSC.2023.3330171>
48. Zhang, Y., Jiang, J., Dong, X., Wang, L., Xiang, Y.: Bedcv: blockchain-enabled decentralized consistency verification for cross-chain calculation. *IEEE Trans. Cloud Comput.* (2022)
49. Zheng, Y., Duan, H., Yuan, X., Wang, C.: Privacy-aware and efficient mobile crowdsensing with truth discovery. *IEEE Trans. Dependable Secure Comput.* **17**(1), 121–133 (2017)
50. Zhou, J., Cao, Z., Dong, X., Lin, X.: Security and privacy in cloud-assisted wireless wearable communications: challenges, solutions, and future directions. *IEEE Wirel. Commun.* **22**(2), 136–144 (2015)



Exploring the Vulnerability of ECG-Based Authentication Systems Through A Dictionary Attack Approach

Bonan Zhang¹(✉) , Chao Chen^{1,2} , Ickjai Lee² , Kyungmi Lee² ,
and Kok-Leong Ong¹

¹ RMIT University, Melbourne, VIC 3000, Australia
sysubonzhang@gmail.com

² James Cook University, Townsville, QLD 4814, Australia

Abstract. Electrocardiogram (ECG)-based authentication has gained popularity recently, but its security measures have not been thoroughly explored. In this paper, we explore dictionary attacks against ECG authentication systems. We attempt to spoof the victim’s ECG model without prior knowledge of the victim’s ECG information. We investigated the feasibility of identifying a “master” collection of ECG signals that may coincide with ECG verification templates saved by authenticated users. Our experiments in four different ECG verification schemes show that these master ECG signals can effectively impersonate the ECG verification profiles of a wide range of users. These findings highlight significant vulnerabilities in current ECG-based authentication systems and can be used to strengthen ECG-based authentication systems.

Keywords: ECG Authentication · Dictionary Attack · Biometrics

1 Introduction

Biometrics are increasingly integral to authentication in various scenarios, due to their convenience and seamless integration, particularly on mobile devices. A review by Arpita et al. (2020) [28] underscores the improvement of user experience with biometrics authentication in mobile settings. The fingerprint [25] and facial recognition [17] systems are the predominant methods used on smartphones. The security and accuracy of these two authentication schemes have been thoroughly evaluated, making them widely accepted and used by the general public. However, their adaptation to smaller wearable devices such as smart-watches is hindered by constraints related to device size and hardware capabilities. The size and computing power of wearable devices make it difficult to add a camera or a fingerprint recognition sensor. On the other hand, these smart wearables have many sensors to monitor biological information, which facilitate a range of biometrics authentication, such as voice authentication [30], gait recognition [11], and ECG authentication [15]. This study specifically focuses on ECG authentication, assessing the security robustness of current implementations.

Electrocardiogram signals can be recorded by electrodes on a wearable device that capture the electrical activity of the heart over a specific period of time [26]. Variations in lifestyle, cardiac activation sequences, heart mass, and individual conductivity produce different ECG patterns between people, as noted by Foteini et al. (2011) [1]. These variations enable the utilisation of ECG signals for authentication purposes. ECG authentication offers a significant advantage over traditional biometric methods because it is an implicit biometric trait. Obtaining a person's ECG signal requires direct physical contact, which enhances the security of the ECG signal.

Although ECG authentication provides enhanced security, it is not completely resistant to presentation attacks, including those from external sources such as replay attacks. Such attacks involve the unauthorised use of user ECG segments [8, 16]. Attacks on ECG authentication systems generally rely on two key assumptions: firstly, the attack specifically targets an individual; and secondly, the attacker has access to that individual's ECG segments. While the mechanisms of these attacks can vary depending on the context of use and the data acquisition method, these fundamental assumptions are consistent. Theoretically, these attacks could significantly compromise the security of ECG authentication systems. However, in practical terms, they are challenging to execute.

In this paper, we present a novel attack strategy on ECG authentication systems. Unlike typical attacks that focus on specific individuals, our approach exploits coincidental user matches within a group, eliminating the need for prior knowledge of the identity of the victim or ECG data. This method involves exploiting stolen smart devices to bypass their ECG-based authentication. Our scheme proves to be effective in a variety of ECG signal feature extraction methods, introducing a considerable new threat to ECG authentication systems. Through this research, our objective is to highlight the inherent security vulnerabilities in current ECG authentication technologies and suggest directions for further optimisation and research.

The contributions of this paper are primarily in three areas:

- We introduce a novel dictionary attack against ECG authentication that adapts to various ECG feature extraction and authentication techniques;
- We assess the effectiveness of our attack by testing it against multiple established ECG authentication methods. Our findings demonstrate the attack's robust generalization capabilities and high transferability;
- Our experiments reveal that despite the overall uniqueness of ECG signals among individuals, there are notable similarities within certain individual ECG signals that can be exploited.

The remainder of the paper is structured as follows: Sect. 2 outlines related work, which encompasses both the implementation of ECG-based authentication methods and dictionary attacks against biometric authentication. Section 3 details the implementation of our attack scheme. Section 4 presents experimental results, analysing the impact of our attack on the authentication models. The final section will synthesise our findings and discuss their implications.

2 Related Work

In this section, we review existing ECG authentication methods and adversarial biometric methods. The section will be divided into three subsections. Firstly, we will introduce the existing schemes based on ECG authentication. This is followed by the currently proposed attacks against ECG authentication. Finally, we will present existing implementations of dictionary attacks against other biometric authentications.

2.1 ECG Authentication Technology

An electrocardiogram (ECG) is generated by placing electrodes on the body to monitor the electrical changes that occur with each heartbeat. Figure 1 illustrates the voltage changes recorded during a single cardiac cycle. A single heartbeat cycle encompasses five principal waves: P, Q, R, S, and T. The P wave is indicative of atrial depolarisation; the QRS complex denotes ventricular depolarisation; and the T wave conveys ventricular repolarisation [19]. Traditionally, this procedure requires the placement of electrodes in ten specific locations on the body of a patient. However, recent technological advances have facilitated the collection of ECG signals through integrated electrodes in wearable devices. This progress has laid the hardware foundation for employing ECG data in authentication systems on such devices.

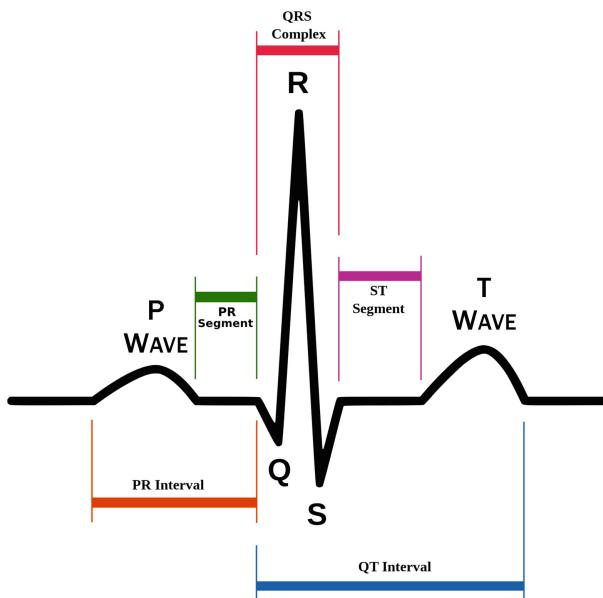


Fig. 1. An example of an ECG cycle.

Methods for authenticating users through the uniqueness of their ECG signals were first proposed in 2001 [5]. The extraction of ECG signal features is divided into Fiducial-based features and non-Fiducial-based features, depending on whether it is necessary to locate the position of each wave in the waveform signal. Fiducial-based feature extraction methods are dominated by the extraction of information from the time domain. This extraction scheme firstly needs to identify the positions of different waves in the ECG signal. From these positional relationships, other information, such as intervals, amplitudes, and angles, can be later extracted [14,32]. The most immediate characteristic of this class of methods is that they are intuitive and easy to understand. Even non-medical people can understand the physical meaning. This class of methods is usually able to achieve higher authentication accuracy than non-fiducials-based systems. However, in order to be able to accurately extract features, precise positioning of the different waves is required to achieve this. This will increase the computational and time cost required for authentication.

Non-Fiducial-based features do not need to locate the position of different waves in ECG signals, and their frequency domain information is used to extract ECG features. Frequency-based features are generally computed by a fast Fourier transform. These features contain information in the frequency domain, such as spectral centroid, spectral energy, spectral roll-off, and flux [7]. The most immediate advantage of this approach is more flexibility. There is no need to accurately identify the location of different fiducials and restrictive segmentation process. However, its extracted features are not intuitive for humans and can only be relying on machine learning models for classification.

After extracting the features, the next step is to select a suitable authentication model. Current authentication schemes fall into two primary categories: similarity-based methods and machine learning-based methods. The foundational principle of the former involves comparing the similarity between newly input ECG signals and those previously registered. If similarity exceeds a certain threshold, authentication is granted [3,9]. For example, the scheme developed by Juan et al. [2] records the position of each heartbeat wave during registration and calculates the distances between these waves and the peaks and troughs. During authentication, the user's identity is confirmed by comparing the similarity between the input features and the registered features. This method achieved 81.82% true acceptance rate and a 1.41% false acceptance rate [3]. Such schemes are noted for their simpler implementations and reduced computational demands, making them well-suited for mobile devices.

The second category of ECG-based authentication is implemented using various machine learning and deep learning algorithms. In the realm of biometric authentication, these methods typically involve implementing a binary classification model for each user [4,10,31]. Authentication using this approach can achieve high accuracy. For example, the scheme developed by Binish et al. [10] using both the SVM and Random Forest algorithms, after extracting ECG features through Fourier decomposition and phase transformation, can achieve the accuracy of over 95%. Similarly, the scheme of Mohamed et al. [4] utilised

the computation of one-dimensional local differences for feature extraction and trained a PNN algorithm, achieves an equal error rate of 3.05%.

To validate the effectiveness of our proposed attack model, we have implemented various ECG authentication schemes employing different feature extraction methods and authentication models in this paper and attempt to bypass these models within a limited number of authentication trials.

2.2 Adversarial Attack in ECG

Adversarial attacks against ECG authentication use mainly cross-device presentation attacks, where the attacker bypasses the authentication system by replaying a victim's ECG signal [8, 16]. These attacks require access to user ECG data, such as stolen medical check-up data. However, due to discrepancies in the ECG data captured by different devices, stolen data cannot be used directly for an attack. To address this, attackers develop transformation equations to adjust stolen ECG signals to match the requirements of the target device, a strategy known as cross-device attacks. Simon et al. [8] pioneered this approach, demonstrating the viability of generating the required ECG signals using a waveform generator connected via electrodes to fool the device's liveness detection. Similarly, Nima et al. [16] refined this method employing transformation equations to facilitate cross-device compatibility and introduced a mitigation strategy that verifies whether ECG signals originate from a real human by analysing heart rate variability.

In contrast to these targeted approaches, our study introduces a dictionary attack method that is non-targeted. Our solution does not require personal information from the victim and is fundamentally separate from traditional attack methods.

2.3 Dictionary Attack in Biometrics

Dictionary attacks against biometric authentication have emerged as a novel adversarial method in recent years, exploiting vulnerabilities previously demonstrated in fingerprint [27], facial [23], and voice recognition systems [21]. These attacks are informed by the concept of the biometric menagerie [37], which posits significant variability in individuals' susceptibility to biometric matching. The theory posits that within the general population, there are specific subsets of users-termed "lambs"-whose biometric templates are more likely to be matched by others. In contrast, there exists another group, known as "wolves," whose biometric data can frequently match with those of other users. This research aims to identify these "wolf" users as a strategy to compromise the "lamb" users' security. The typical approach involves identifying a fingerprint with the highest impersonation potential and then refining this "master fingerprint" using a first-order hill-climbing algorithm.

To date, there has been no research on applying dictionary attacks to ECG authentication. Our study fills this gap by demonstrating that our proposed dictionary attack can successfully authenticate 10% of users on average within five

attempts in a black box model scenario. This paper details our attack methodology and tests its effectiveness against various ECG authentication models and schemes.

3 Attack Strategy and Framework

In this section, we elaborate on two key parts of this study: our attack model and the reconstruction of the black-box authentication model.

3.1 Attack Model

In our designed threat model, only one user is registered per mobile device. The attacker tries to spoof the device's authentication model to access the device. In this attack, the attacker does not have any other information about the victim. In our designed experiments, it is only assumed that the device collects the user's ECG signals through electrodes on the device. To spoof the device's black-box authentication model, the attacker needs to submit the ECG model for the device to examine. In previous work, it has been shown that the attacker can generate the desired ECG waveform through a waveform generator and input it into the device through electrodes [8].

To improve the success rate of dictionary attack, an attacker should first acquire a comprehensive collection of ECG signals and then identify a subset with the highest deception success rate. To achieve this, it is crucial to establish a set of evaluation procedures to assess the impostor rate for each heartbeat cycle in the database. In our study, we have developed an ECG authentication model based on waveform similarity specifically for this purpose, as described in [3]. Although constructing alternative evaluation frameworks might improve attack success rates, we consider this an avenue for future research. Figure 2 below illustrates the details of our current framework.

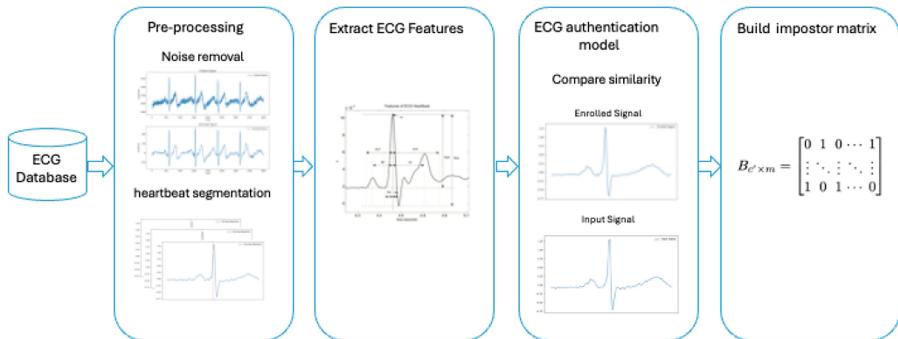


Fig. 2. The Framework of proposed attack model.

This framework has been divided into four main sections. The first part of this experiment is the pre-processing part. The main task of this part is to denoise and segment the signal. The ECG signal inevitably contains noise when acquired due to the user's muscle movement and the accuracy of the sensor. In order to remove this noise, we use a set of combinational filters to remove the noise. The filter consists of four parts which are wavelet drift correction, adaptive band stop filter, low pass filter, and smoothing filter [20]. Figure 3 shows an ECG signal before and after denoising. Most of the ECG certification is achieved by analysing a single heartbeat cycle. We also need to segment the ECG by heartbeat cycle. Segmentation is conducted by first finding the location of the R-wave in the waveform using the Pan-Tompkins algorithm [24]. After that, the R-wave is used as the centre of each heartbeat cycle to intercept the data with a length of 700 ms.

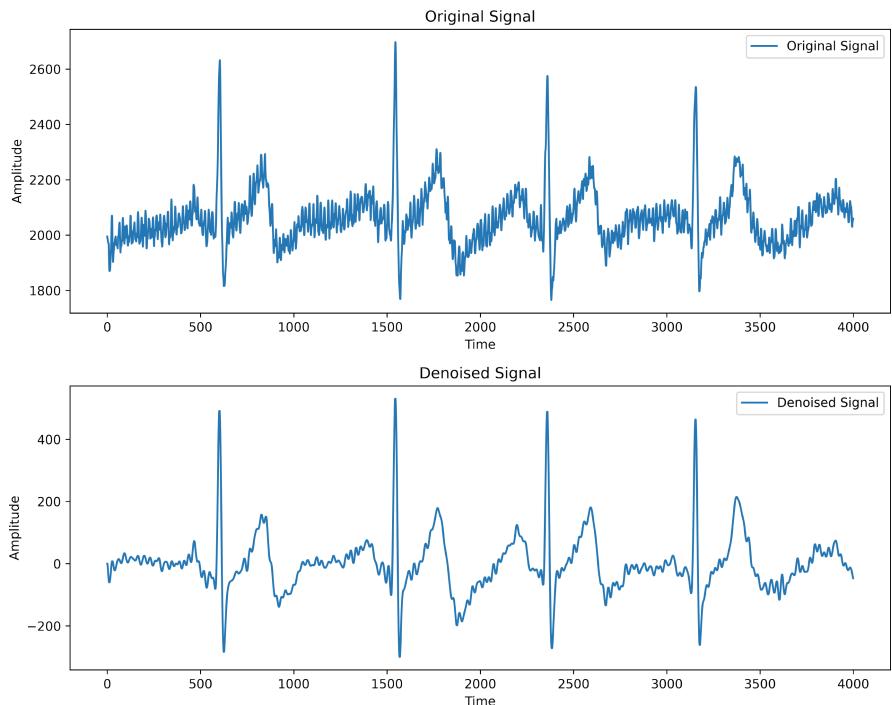


Fig. 3. Electrocardiogram signal denoising.

After the ECG segmentation is complete, the next step is to extract the ECG features for each heartbeat cycle. In this study, we mainly extract features using information from the time domain. The features we extracted include the positional relationship between each wave in the ECG and the R wave, and also the amplitude of the R and S waves.

The third part of the work is to construct an ECG authentication framework to identify the ECG segments with the highest impostor rates. In this experiment, we construct this evaluation framework for the authentication model using the method proposed by Juan et al. [3]. The authentication part of the model is constructed by calculating the average of the features extracted from the 30-second ECG used for registration as the registered ECG feature data. When the user tries to authenticate, it compares the similarity between the input data and the registration data to determine whether it can pass the authentication or not by exceeding a threshold value.

In the last step, we will calculate the impostor matrix. Firstly, a size matrix $M \times N$ must be constructed based on the number of users N and the number of ECG segments M in our database. And set all the initial values within this matrix to 0. We will use each segment of the ECG W_m to attack the authentication model V_n of all users. If W_m is capable of spoofing the authentication model of user n , change the data at position (m, n) of the matrix to 1, otherwise leave it unchanged. The formula for this matrix is as follows.

$$\mathbf{B}_{M \times N} = \begin{bmatrix} v_1(\mathbf{w}_1) & \dots & v_N(\mathbf{w}_1) \\ \dots & \dots & \dots \\ v_1(\mathbf{w}_M) & \dots & v_N(\mathbf{w}_M) \end{bmatrix}$$

$$v_n(w_m) = \begin{cases} 1 & \text{if } w_m \text{ is authenticated,} \\ 0 & \text{if } w_m \text{ is not authenticated.} \end{cases}$$

With this matrix, we can then calculate the impostor rate for each heartbeat segment within the database. The impostor rate can be calculated using the following equation.

$$\text{IR}(w_m) = \frac{1}{|N|} \sum_{n \in N} v_n(w_m).$$

Since no biometric authentication method can achieve 100% accuracy, multiple authentication attempts are typically allowed. This feature can be exploited to identify a set of ECG segments with the highest impostor rates. For our study, we set the number of permissible authentication attempts to five. Our strategy to maximise the breach of the authentication model involves a systematic selection of ECG segments. In each round, we select the segment with the highest impostor rate. After each selection, the users successfully spoofed by the chosen segment are removed from the pool, and the impostor rates for the remaining segments are recalculated. This process is repeated to optimise the effectiveness of the attack, ensuring that each selected segment maximises the number of users it can impersonate. Here is how the selection process is implemented:

1. Select the segment with the highest impostor rate:

$$w^* = \arg \max_{w \in W} \text{IR}(w).$$

2. Update the set of users:

$$U_{t+1} = U_t \setminus \{u \in U_t : v(w, u) = 1\}.$$

3. Re-evaluate the impostor rate for remaining segments:

$$I_t(s) = \text{new impostor rate of } s \text{ with respect to } U_{t+1}.$$

4. Repeat these steps 1–3 until 5 ECG segmentation are selected.

3.2 Extraction of ECG Features

In order to fully evaluate the effectiveness of our attack method, we start with two aspects to evaluate the method comprehensively. The first is whether the method is effective against different ECG feature extraction methods. The second is whether the method can be effective against different authentication models. We illustrate the feature extraction methods used in our experiments in this section.

Fiducial Based Method. In this experiment we tried two different target-based feature extractions. The first method is mainly achieved by extracting the positional relationships and amplitudes between waveforms. After determining the positions of the different waves, the distances between the T, Q, and S waves and the R wave are calculated. The amplitude of the R wave and the amplitude gap between the R and S waves were also added as supplementary features. The advantage of this method is that it is less computationally stressful for mobile devices and enables fast feature extraction.

The second feature extraction method we have implemented uses the one-dimensional local difference pattern algorithm to extract statistical features from ECG signals [4]. The method achieves feature extraction by calculating the difference between the ECG proximity samples. The advantage of this method is that it does not change dramatically depending on the user's mood change. A more stable authentication process can be achieved.

Non-fiducial Based Method. For the non-Fiducial-based feature extraction method, we used the Fourier decomposition algorithm proposed by Binish et al. [10] to decompose the ECG signal into a set of Fourier intrinsic band functions before extracting relevant features from it. The hidden features of the ECG signals are extracted by changing the phase of the ECG signals while preserving the energy and frequency content of the ECG signals by means of phase transformation. The features extracted for each Fourier intrinsic band function include the autoregressive coefficient, energy distribution, Skewness, and Kurtosis.

3.3 Black Box Model Design and Construction

As previously discussed, ECG authentication methods fall into two primary categories. The first involves comparing similarity scores between registered and input signals to verify if the user is trying to unlock the device. The second category uses feature engineering and machine learning to build classifiers for authentication purposes.

To comprehensively assess the effectiveness of our attack method, we built and evaluated a range of classifiers, from those based on signal similarity to those employing deep learning techniques. We also ensured diversity in our ECG feature extraction methods [3, 4, 10]. We selected and optimised four state-of-the-art ECG authentication algorithms to achieve the performance reported in previous studies. The models we implemented are listed below.

Similarity-Based Authentication Model. There were significant differences in the positions and heights of the peaks and troughs of the various waves in the ECGs for different users [9]. In order to reproduce the work of [3], we extracted the positions of individual waves in the ECG signal and calculated the positional relationships of these waves and the altitudes of the peaks. These features are extracted at the time of registration, and the average of these features is calculated as the registration data. At the time of authentication, the system only needs to check whether the similarity between the features extracted from the input signal and the registration data is within the set threshold.

Probabilistic Neural Network-Based Authentication Model. Probabilistic Neural Networks (PNN) are favoured for their rapid training speeds compared to other deep neural network models, as observed by Neha et al. [29]. For the effective use of PNN models, we also engaged in the feature engineering of the ECG data. Following the approach of Mohamed et al. [4], we extracted features by calculating the one-dimensional local difference pattern of the ECG. This method exploits differences between consecutive samples to detect micro- and macro-patterns within the ECG signal. Our replication efforts adhered to the same model parameter settings described in their study, achieving classification accuracies comparable to those reported in the original work.

Support Vector Machines-Based Authentication Model. Support Vector Machines (SVM) are renowned for their robust performance in classification tasks and their ability to effectively manage generalisation errors, as noted by Maryamsadat et al. [13]. Given the limitations of using raw ECG data for classification, we adopted the approach outlined by Fatimah et al. [10], which involves feature engineering the ECG data initially. This method employs Fourier decomposition to break down the ECG into a series of Fourier intrinsic band functions, from which pertinent features are extracted. Following the recommendations in the literature, we use libsvm [6] with a polynomial kernel to train the model tailored for each user.

Random Forest-Based Authentication Model. Random forests are ensemble classifiers that employ bagging with decision trees as their base classifiers, known for effectively preventing model overfitting, as highlighted by Robin et al. [35]. In this study, we applied the same feature extraction method used in the SVM model to train our random forest model, following the protocols outlined by Fatimah et al. [10]. The training results closely matched the accuracy reported in the original work.

4 Experiment Evaluation

In this section, we evaluate in detail whether the attack model is effective against a variety of different authentication models. Our final experimental results show that the attack is effective in different settings.

4.1 Datasets

In this study, we selected four public ECG databases for conducting evaluation experiments. They are MIT-BIH Normal Sinus Rhythm Database [12], MIT-BIH Arrhythmia Database [22], European ST-T Database [34] and QT Database [18]. These databases are widely used to train and test the performance of ECG authentication models [2, 32, 33, 36]. The details of those datasets are shown in Table 1

Table 1. Overview of ECG datasets used.

Dataset	Subjects	Gender F/M	Frequency	Signal length
European ST-T	79	8/70/1	250 hz	2H
MIT-BIH Normal Sinus Rhythm	18	13/5	128 hz	24H
MIT-BIH Arrhythmia	48	-	360 hz	30 min
QT Database	71	-	250 hz	15 min

- European ST-T: The database holds ECG data from a total of 79 subjects for a duration of two hours. The ages of the participating subjects ranged from 30 to 84 years. The dataset samples ECG data using a sampling frequency of 250 hz.
- MIT-BIH Normal Sinus Rhythm: The database contains 18 entries of long term data from electrocardiograms. The dataset contains data from 13 females and 5 males. The subjects were between 26–50 years of age and each data entry was up to 24 h in length. In this study we selected only the first 3 h of data for our experiment.
- MIT-BIH arrhythmia: The database contains a total of 48 ECG records, each of half an hour’s duration. The dataset was sampled at 360 hz.

- QT Database: The database contains a total of 104 subjects with ECG data of 15 min in length. However, the source of data for 33 of these entries is the ST-T database. To avoid conflicts, we have filtered this part of the data

Data from these databases are collected and sampled using different equipment. To achieve uniformity, we resampled all ECG signals so that each ECG signal was sampled at 250 hz.

4.2 Evaluation Metric and Benchmark

In this experiment we will use the following criterion to evaluate the performance.

- True Positive(TP): The authentication system correctly identified the registered user.
- False Positive(FP): Authentication system error accepting non-authenticated users.
- True Negative(TN): The authentication system correctly rejects non-registered users.
- False Negative(FN): The authentication system incorrectly rejected the registered user.
- True Accept Rate(TAR): $TP/(TP+FN)$ The probability that the system correctly handles an authentication attempt when a registered user tries to authenticate.
- False Accept Rate(FAR): $FP/(FP+TN)$ The probability of system error handling when a nonregistered user attempts authentication.

In this experiment, to demonstrate the effectiveness of our proposed attack, we designed two benchmark attacks for comparative evaluation: the random attack and the one-cycle attack. The random attack involves constructing an ECG database which can be obtained from a public database or collected independently by the attacker. When attempting to breach a device’s security, the attacker randomly selects a data sample from this database to use in the attack. The one-cycle attack, successfully used in gait-based authentication methods, aims to extract adversarial biological samples more efficiently. According to Zhu et al. [38], if an initial biometric sample fails to deceive the system, the next chosen sample should differ significantly from the current one. In our implementation, we first used a CNN algorithm to extract features from ECG signals. The initial attack sample is the centroid of all the extracted features. Subsequently, the likelihood of selecting the remaining samples for subsequent attacks is proportionate to their distance from the currently selected sample. We evaluated the success rates of both benchmark attacks over 100 attempts to mitigate the impact of outliers.

4.3 ECG Authentication Performance

In this experiment, to avoid any overlap between the datasets, we divide the data set into two distinct parts: the replica authentication dataset and the attack data

set. We implemented the authentication model using only the ST-T database for training. The QT, MIT Normal Sinus Rhythm, and MIT Arrhythmia datasets are utilised solely to attack the authentication model. To verify the effectiveness of our attack, it is crucial to first verify that our replicated authentication model performs comparably to previous studies. Accordingly, we will begin by evaluating the performance of our authentication model.

Table 2. Authentication model performance and threshold.

Model name	TAR	False Accept Rate	Threshold(TAR)	Threshold(FAR)
Similarity model [3]	82.57%	3.85%	>80%	<3%
SVM [10]	97.81%	0.62%	>99%	<1%
Random forest [10]	98.07%	0.39%	>99%	<1%
PNN [4]	82.31%	2.55%	>80%	<3%

Table 2 and Fig. 4 provide a comprehensive overview of the performance of our ECG authentication models. These models generally achieved excellent results, with the similarity and PNN models achieving a TAR of more than 80% and a FAR of less than 3%, while the SVM and Random Forest models achieved better performance, in line with the results obtained in previous studies.

The box plots in Fig. 4 illustrate the variability and distribution of these performance metrics between different users. In particular, while the average TAR is high for all models, the variance indicated by the interquartile range and outliers suggest that not all users are equally well served by the models.

To further increase the challenge of the attack, we selectively filter out users with substandard authentication performance. Only users whose performance metrics exceed the predefined thresholds are included in further security analyses. The specifics of our threshold setting are detailed in the Table 2.

4.4 Performance on Attack Model

After replicating the authentication models, we evaluated the robustness of the existing ECG authentication systems against our dictionary ECG attack. The success rates of our attacks across different models are summarised in Table 3. For each black-box authentication system, we conducted both single and five-attempt dictionary attacks, as well as random and one-cycle attacks. We then compiled statistics on the percentage of user models that were successfully spoofed by these attacks.

From Table 3, it is clear that dictionary attacks are effective for existing ECG authentication methods, and all can significantly increase the probability of breaking into the system. With just a single authentication attempt, approximately 3% of the user authentication systems are compromised. In contrast, the other two benchmark attack methods we evaluated breached only about 1% of the systems.

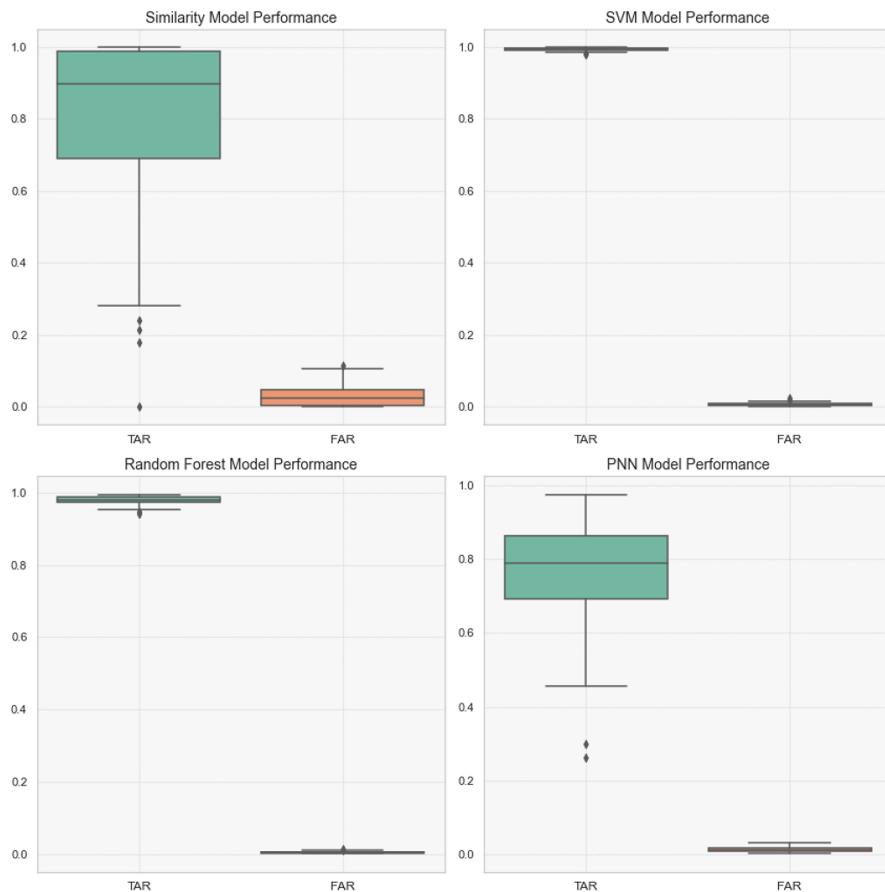


Fig. 4. Performance of different ECG authentication model

The increase in success rate with five attempts (approximately 15% breach rate) underscores the vulnerability of these systems to multiple attempts. This phenomenon is particularly pronounced in the SVM and Random Forest models, where the breach rates jump substantially compared to the one-time attempt. The disproportionate increase in successful cracking rates suggests that a larger range of users can be reached through careful selection of ECG segments. And again, the success rate of the dictionary attack has greatly exceeded the benchmark method in five attempts. This shows that our dictionary attack approach is also effective in disrupting existing ECG-based biometric authentication schemes while limiting the number of authentication attempts.

Figure 5 illustrates the impostor rates for each presentation attack in four different models. This visualisation highlights significant differences in model performance over multiple attempts. In particular, SVM models and random forest models show a faster increase in impostor rates, suggesting that ECG

Table 3. Success rate of attacks on various authentication methods using three different attacks. The results of both benchmark attack methods are based on the average of 100 independent experiments.

Model name	Try times	MasterECG	Random Attack	One cycle attack
Similarity Model	1	3.77%	0.69%	0%
	5	9.43%	3.65%	2.03%
SVM	1	3.17%	1.71%	3.12%
	5	17.46%	10.37%	10.93%
Random Forest	1	2.32%	0.84%	0%
	5	13.95%	4.4%	3%
PNN	1	4.87%	1.8%	0%
	5	12.19%	9.26%	7.8%

authentication models based on machine learning algorithms may be more susceptible to dictionary attacks.

The concept of “wolf” users, individuals whose ECG patterns closely resemble those of many others, becomes evident in this analysis. Certain master ECGs dramatically increase success rates in some models, indicating that these user ECGs are likely to cause false positive identifications. This highlights the need for researchers to think about how they can effectively categorise such ECG systems correctly when designing ECG authentication models.

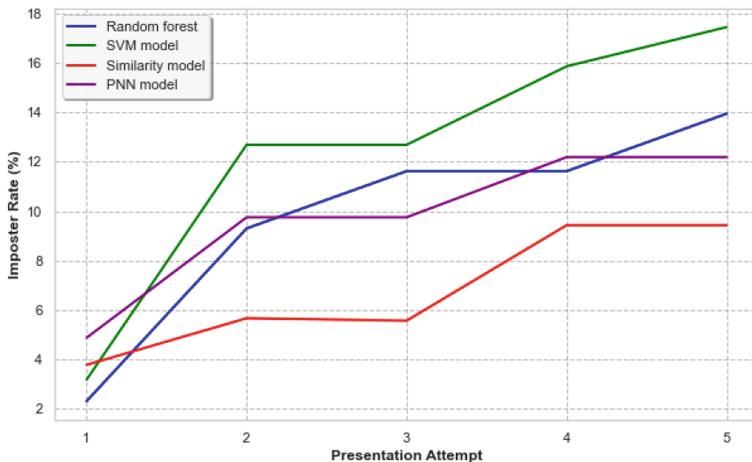


Fig. 5. Impersonation rates of master ECG under multiple presentation attempts ($n = 5$) in the black-box attack

5 Conclusion and Future Work

In this paper we present a novel dictionary attack against ECG authentication. This approach achieves its goal by randomly matching a significant portion of the user base. It does not require any contextual information related to the victim as compared to previous attack methods targeting specific users. After experimental validation, our attack method is versatile enough to be used against different ECG feature extraction methods and different ECG authentication models. We provide a comprehensive evaluation of the dictionary attack against existing state-of-the-art ECG authentication methods. The main conclusions of our work are as follows.

- All existing ECG authentication systems lack effective means to defend against dictionary attacks. Even if they can achieve very good performance in the model construction phase, they can still match roughly 15% of their users through dictionary attacks.
- Although there is a large gap in overall ECG between different users, there is a possibility of similarity between individual heartbeat cycles.
- There are also groups of users who are easily matched by others and those who are easily matched by others in ECG authentication.

Our experimental results show that dictionary attacks may pose a serious threat to ECG-based authentication methods. However, our work still has several limitations. The first is that in this paper we only use a similarity-based authentication model to find out the master ECG, and we do not experimentally find out which authentication model can best pick out the master ECG. The second is that the data set used in this paper contains a small number of users and there is no guarantee that it is representative of the vast majority of all user groups.

Looking ahead, future work could focus on two main areas: First, employ generative methods to produce ECG segments with a higher impostor rate. Currently, our attacks use real ECG data, but adopting adversarial generation techniques could produce segments that can more effectively breach authentication systems. Second, develop mitigation algorithms to counteract this type of attack. One potential approach is to enrich the training data of the authentication schemes with more background data, improving the system's ability to distinguish users more accurately. We believe that our work will ultimately help us better understand the ECG certification model and enable safer certification in the future.

References

1. Agrafioti, F., Gao, J., Hatzinakos, D., Yang, J.: Heart biometrics: theory, methods and applications. *Biometrics* **3**(199–216), 25 (2011)
2. Arteaga-Falconi, J.S., Al Osman, H., El Saddik, A.: ECG and fingerprint bimodal authentication. *Sustain. Urban Areas* **40**, 274–283 (2018)

3. Arteaga-Falconi, J.S., Al Osman, H., El Saddik, A.: ECG authentication for mobile devices. *IEEE Trans. Instrum. Meas.* **65**(3), 591–600 (2015)
4. Benouis, M., Mostefai, L., Costen, N., Regouid, M.: ECG based biometric identification using one-dimensional local difference pattern. *Biomed. Signal Process. Control* **64**, 102226 (2021)
5. Biel, L., Pettersson, O., Philipson, L., Wide, P.: ECG analysis: a new approach in human identification. *IEEE Trans. Instrum. Meas.* **50**(3), 808–812 (2001)
6. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2**(3), 1–27 (2011)
7. Dargie, W.: Analysis of time and frequency domain features of accelerometer measurements. In: 2009 Proceedings of 18th International Conference on Computer Communications and Networks, pp. 1–6. IEEE (2009)
8. Eberz, S., Paoletti, N., Roeschlin, M., Kwiatkowska, M., Martinovic, I., Patané, A.: Broken hearted: how to attack ECG biometrics. In: Network and Distributed System Security Symposium 2017. Internet Society (2017)
9. Fang, S.C., Chan, H.L.: QRS detection-free electrocardiogram biometrics in the reconstructed phase space. *Pattern Recogn. Lett.* **34**(5), 595–602 (2013)
10. Fatimah, B., Singh, P., Singhal, A., Pachori, R.B.: Biometric identification from ECG signals using fourier decomposition and machine learning. *IEEE Trans. Instrum. Meas.* **71**, 1–9 (2022)
11. Gafurov, D., Snekkenes, E., Bours, P.: Gait authentication and identification using wearable accelerometer sensor. In: 2007 IEEE Workshop on Automatic Identification Advanced Technologies, pp. 220–225. IEEE (2007)
12. Goldberger, A.L., et al.: Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation* **101**(23), e215–e220 (2000)
13. Hejazi, M., Al-Haddad, S.A.R., Singh, Y.P., Hashim, S.J., Aziz, A.F.A.: ECG biometric authentication based on non-fiducial approach using kernel methods. *Digit. Signal Process.* **52**, 72–86 (2016)
14. Israel, S.A., Irvine, J.M., Cheng, A., Wiederhold, M.D., Wiederhold, B.K.: ECG to identify individuals. *Pattern Recogn.* **38**(1), 133–142 (2005)
15. Kang, S.J., Lee, S.Y., Cho, H.I., Park, H.: ECG authentication system design based on signal analysis in mobile and wearable devices. *IEEE Signal Process. Lett.* **23**(6), 805–808 (2016)
16. Karimian, N., Woodard, D., Forte, D.: ECG biometric: spoofing and countermeasures. *IEEE Trans. Biometrics Behav. Identity Sci.* **2**(3), 257–270 (2020)
17. Kaur, P., Krishan, K., Sharma, S.K., Kanchan, T.: Facial-recognition algorithms: a literature review. *Med. Sci. Law* **60**(2), 131–139 (2020)
18. Laguna, P., Mark, R.G., Goldberg, A., Moody, G.B.: A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In: Computers in Cardiology 1997, pp. 673–676. IEEE (1997)
19. Lilly, L.S.: Pathophysiology of heart disease: a collaborative project of medical students and faculty. Lippincott Williams & Wilkins (2012)
20. Lugovaya, N.: Biometric human identification based on ECG. In: Russian Conference on Mathematical Methods of Pattern Recognition (2005)
21. Marras, M., Korus, P., Jain, A., Memon, N.: Dictionary attacks on speaker verification. *IEEE Trans. Inf. Forensics Secur.* **18**, 773–788 (2022)
22. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **20**(3), 45–50 (2001)

23. Nguyen, H.H., Yamagishi, J., Echizen, I., Marcel, S.: Generating master faces for use in performing wolf attacks on face recognition systems. In: 2020 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–10. IEEE (2020)
24. Pan, J., Tompkins, W.J.: A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* **3**, 230–236 (1985)
25. Rahmawati, E., et al.: Digital signature on file using biometric fingerprint with fingerprint sensor on smartphone. In: 2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), pp. 234–238. IEEE (2017)
26. Rathore, A.S., Li, Z., Zhu, W., Jin, Z., Xu, W.: A survey on heart biometrics. *ACM Comput. Surv. (CSUR)* **53**(6), 1–38 (2020)
27. Roy, A., Memon, N., Ross, A.: Masterprint: exploring the vulnerability of partial fingerprint-based authentication systems. *IEEE Trans. Inf. Forensics Secur.* **12**(9), 2013–2025 (2017)
28. Sarkar, A., Singh, B.K.: A review on performance, security and various biometric template protection schemes for biometric authentication systems. *Multimedia Tools Appl.* **79**(37), 27721–27776 (2020)
29. Sharma, N., Om, H., et al.: Usage of probabilistic and general regression neural network for early detection and prevention of oral cancer. *Sci. World J.* **2015** (2015)
30. Shi, C., Wang, Y., Chen, Y., Saxena, N., Wang, C.: Wearid: low-effort wearable-assisted authentication of voice commands via cross-domain comparison without training. In: Proceedings of the 36th Annual Computer Security Applications Conference, pp. 829–842 (2020)
31. da Silva Luz, E.J., Moreira, G.J., Oliveira, L.S., Schwartz, W.R., Menotti, D.: Learning deep off-the-person heart biometrics representations. *IEEE Trans. Inf. Forensics Secur.* **13**(5), 1258–1270 (2017)
32. Singh, Y.N., Gupta, P.: Biometrics method for human identification using electrocardiogram. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 1270–1279. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01793-3_128
33. Singh, Y.N., Singh, S.K., Gupta, P.: Fusion of electrocardiogram with unobtrusive biometrics: an efficient individual authentication system. *Pattern Recogn. Lett.* **33**(14), 1932–1941 (2012)
34. Taddei, A., et al.: The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography. *Eur. Heart J.* **13**(9), 1164–1172 (1992)
35. Tan, R., Perkowski, M.: Toward improving electrocardiogram (ECG) biometric verification using mobile sensors: a two-stage classifier approach. *Sensors* **17**(2), 410 (2017)
36. Wang, Y., Agrafioti, F., Hatzinakos, D., Plataniotis, K.N.: Analysis of human electrocardiogram for biometric recognition. *EURASIP J. Adv. Signal Process.* **2008**, 1–11 (2007)
37. Yager, N., Dunstone, T.: The biometric menagerie. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(2), 220–230 (2008)
38. Zhu, T., Fu, L., Liu, Q., Lin, Z., Chen, Y., Chen, T.: One cycle attack: fool sensor-based personal gait authentication with clustering. *IEEE Trans. Inf. Forensics Secur.* **16**, 553–568 (2020)



LBVP: Lightweight Blockchain-Based Vehicle Platooning Scheme for Secure and Efficient Platoon Management

Wenjie Fan¹, Zhiqian Liu^{1,5(✉)}, Libo Wang^{2(✉)}, Ying He³, Jingjing Guo⁴, Xia Feng², and Jianfeng Ma⁴

¹ Jinan University, Guangzhou, China
fanwenjie@live.cn

² Hainan University, Haikou, China
{wanglibo,xiafeng}@hainanu.edu.cn

³ Shenzhen University, Shenzhen, China
heying@szu.edu.cn

⁴ Xidian University, Xi'an, China

jjguo@xidian.edu.cn, jfma@mail.xidian.edu.cn

⁵ Guangxi Key Laboratory of Trusted Software, Guilin, China
zqliu@vip.qq.com

Abstract. Blockchain can help platoon management ensuring stability, providing robust privacy protection. Current blockchain-based platoon management schemes suffer from low integration and inefficiency, especially in resource-constrained environments. This leads to significant computational and communication overhead, making blockchain a bottleneck. Aiming at addressing the significant computational overhead of integrating blockchain with platoon management in edge devices, we propose a Lightweight Blockchain-based Vehicle Platooning (LBVP) scheme for secure and efficient platoon management. We introduce a multi-chain architecture comprising one persistent chain and multiple travel chains. This architecture leverages the Proof of Platoon Travel (PoPT) protocol, which diminishes the intensive direct interactions between platoons and the persistent chain while ensuring that the persistent chain remains updated on the platoon's status. This scheme significantly enhances network performance, scalability, and security. Furthermore, the LBVP scheme incorporates a lightweight consensus algorithm that optimizes interaction phases, thereby substantially reducing computation overheads such as consensus latency, memory consumption, and CPU utilization. Comprehensive security analysis demonstrate that our scheme effectively preserves the security of vehicles and platoons while resisting several common attacks. Experimental results indicate that the LBVP scheme achieves, on average, 63% lower latency, 21% less memory consumption, and 33% lower computational cost compared to existing schemes, underscoring its efficiency and practicality in real-world applications.

Keywords: Platoon management · Blockchain · Consensus · Vehicular networks

1 Introduction

The current research direction in vehicular networks is to achieve a higher level of automation and smarter management in transportation by integrating new technologies like data analytics, cloud computing, blockchain, and artificial intelligence into vehicular networks [8,37].

In recent times, a trend has emerged in vehicle platooning, a new driving paradigm, in which vehicles drive in a line with small gaps between each other. Unlike the conventional autonomous driving paradigm which mainly relies on components (e.g. sensors, Electronic Control Unit) within a single vehicle, vehicle platooning leverages vehicle-to-vehicle (V2V) communication to maintain formation and enhance driving efficiency [14,15]. The benefits of driving in a platoon are twofold: the efficient optimization of speed between vehicles and the reduction of fuel consumption due to decreased air resistance [27]. Additionally, the energy-saving advantages increase proportionally with the travel distance of the platoon [25]. Compared to other scenarios in vehicular networks, vehicle platooning demands include enhanced stability, robust privacy protection, and reduced latency, which are imperative for the successful implementation and operation of advanced platoon management systems [6].

However, traditional centralized schemes encounter several issues, including centralized vulnerabilities such as a single point of failure, fragile data integrity and tampering risks, scalability problems, and a lack of transparency and auditability [1]. Previous schemes [21,22,36] employed cryptographic primitives to design management protocol. These schemes have shown outstanding security performance, effectively defending against a broad spectrum of potential threats and vulnerabilities. Though these schemes greatly enhanced the security level, they are still far from meeting the latency and resource requirements of vehicle platooning.

Recently, many researches have increasingly introduced blockchain technology. By leveraging distributed ledgers, smart contracts, and consensus algorithms, blockchain is easy to scale and inherently resistant to data modification. The blockchain technology can ensure the integrity and transparency of data while minimizing the involvement of a central authority [5]. Therefore, subsequent blockchain-based schemes [20,35] effectively cope with the challenge brought by the centralized framework and ensure robust security. Despite this, current blockchain-based methods suffer from a low level of blockchain technology integration, and the issue of inefficiency still persists.

1.1 Our Contributions

Aiming at reducing the blockchain computation overheads on the edge side like Onboard Unit and Roadside Unit and overcoming the major challenges of integrating blockchain with vehicular networks identified in [29], we propose a Lightweight Blockchain-based Vehicle Platooning (LBVP) scheme for platoon management. In short, the major contributions of this paper can be summarised as follows.

- We design a multi-chain architecture featuring one persistent chain and multiple travel chains. Based on this blockchain architecture, we employed a Proof of Platoon Travel (PoPT). While reducing heavy interactions between the platoon and the persistent chain, PoPT also act as heartbeat mechanism to ensure that the persistent chain can stay updated on the platoon’s status, thereby enhancing network performance, scalability, and security.
- The proposed LBVP scheme also provide a lightweight consensus algorithm (hereafter the LBVP consensus algorithm). By optimizing the interaction phases, computation overheads such as consensus latency, memory consumption, and CPU utilization can be dramatically reduced as a result.
- We conduct comprehensive analysis, and the results of which demonstrate that our scheme can not only preserve the security of vehicle and platoon but also resist several common attacks. Additionally, we conduct a series of experiments, and the results show that, on average, the proposed LBVP scheme achieves 63% lower latency, 21% less memory consumption, and 33% lower computational cost compared to existing schemes.

1.2 Organization

The remaining organization of this paper is as follows. The existing related work are described in Sect. 2. Then, Sect. 3 presents system model, attack model, design goals, and formalized symbols, and Sect. 4 details the various stages in the LBVP scheme. Afterwards, security analysis and simulation evaluation are presented in Sect. 5 and Sect. 6, respectively. Conclusion of this work and future research directions are provided in Sect. 7.

2 Related Work

The blockchain technology secures vehicular networks by providing a decentralized, tamper-resistant ledger for storing and transmitting data. It ensures data integrity through consensus algorithms, facilitates secure transactions with cryptographic techniques, and employs smart contracts for automated, tracable access control and trust management, thereby preventing unauthorized access and cyberattacks [3]. To better leverage the advantages of blockchain, current research directions can primarily be divided into two approaches.

The first approach is to design blockchain network structures that are more suited to specific application scenarios, addressing potential performance issues of blockchain technology in real-world applications. Specifically, Lin et al. [18] took advantage of immutability and large number of participants of Ethereum to protect the conditional privacy in authentication and achieve robust security. Based on the previous work, Lin et al. [19] improves key derivation efficiency and records anonymous public keys on a single consortium chain. Even though these single-chain based schemes address security and privacy issues, they incur substantial costs related to verification and traceability in platoon management. To

address the issues of single-chain architecture, Zhu et al. [39] designed a double-chain blockchain based crowdsensing scheme, using public chain to verify and store sensing data, and using sub chain to collect data. Zhang et al. [38] combined private and consortium blockchains to separate different entities' access permissions and data reads/writes to enhance security and promote decentralization. Following previous methods, Li et al. [17] built a two-layer blockchain includes a backbone and a user layer. The backbone layer consists of a blockchain of all RSUs, and the user layer splits the vehicles within the coverage of RSUs into different blockchains. By separating operations (vehicle registration and message sharing) with different read and write frequencies onto two chains, Tandon et al. [30] proposed D-BLAC scheme that further improve performance. Although these schemes use multi-chain architectures to alleviate the interaction pressure on a single chain, their designs still involve heavy interactions within specific chain, making them insufficiently lightweight and difficult to scale.

The other approach is to improve blockchain consensus algorithms, enhancing block generation speed to achieve low latency and high throughput. Consensus algorithms in existing research can be categorized into two types: proof-based consensus and BFT-based consensus. And after Proof-of-Work (PoW) [26], many proof-based consensus algorithms have emerged, all of them achieve consensus based on the proof of specific properties of nodes. Wang et al. [33] described a consensus algorithm in which nodes with higher reputations are more likely to generate blocks, thereby allowing nodes with lower computing resources but high honesty to have more opportunities to participate in the consensus process. Kara et al. [11] proposed an algorithm that selects candidate and miner nodes based on random conditions, achieving robust fault tolerance and low computational resource requirements. Hou et al. [9] proposed MPoR that controls the number of consensus nodes using the average access time of network nodes as a threshold, and detect and eliminate malicious nodes using a multi-weight reputation algorithm. Another type of consensus algorithms are based on classical algorithms like the Practical Byzantine Fault Tolerance (PBFT) [2]. For instance, Lao et al. [13] optimized the primary node selection using geographic information. Xu et al. [34] enhanced PBFT by incorporating a score grouping mechanism to select high-score nodes for consensus and optimizing the commit phase. Vishwakarma et al. [31] proposed LBSV that improved the performance of consensus latency by reducing interaction phases and giving equal chances to nodes for proposing blocks in a round-robin manner. However, providing equal block generation opportunities to different nodes poses significant security challenges to the system, as malicious nodes also have an equal chance to generate block to attack the network. To address this issues, Kumar et al. [12] incorporated a reputation-based selection process that prioritizes IoV objects with higher reputation values for leader and secondary roles and further optimized consensus phases, thereby reducing consensus latency and ensuring fairness. In short, proof-based consensus algorithms rely on specific node properties that may be easy to alter, while BFT-based algorithms face low scalability and potential centralization trend.

Aiming at overcoming the aforementioned limitations in the existing schemes, we propose LBVP scheme and an intuitive property comparison with the existing schemes is shown in Table 1.

Table 1. Comparison between LBVP scheme and the existing schemes

Property	[39]	[4]	[34]	[31]	[12]	LBVP
Leader selection	✗	✓	✓	✓	✓	✓
Platoon management	✓	✗	✓	✗	✓	✓
Multi-chain architecture	✓	✗	✗	✗	✗	✓
Lightweight consensus algorithm	✗	✗	✗	✓	✓	✓
Long range attack-resisted	✗	✗	✗	✗	✗	✓

3 System Model, Attack Model, Design Goals, and Formalized Symbols

3.1 System Model

The architecture of LBVP scheme is shown in Fig. 1. The three types of entities involved in this scheme are: trusted authority, roadside units, and vehicle.

- Trusted authority (TA): The TA is mainly responsible for the registration of vehicles. The TA is not responsible for storing the key pair of other entities, but plays an important role that verify and store the identity of vehicle, and generates and distributes the certificate to a vehicle when the registration is successful. Besides, the TA contains a clock that divides the time into a series of time intervals with equal length
- Roadside units (RSUs): The RSUs are edge computing devices typically installed on the side of the road, and they generally connect to other RSUs and nearby vehicles via wired and wireless manner, respectively. In other words, the RSUs will collect and verify data from vehicles, acting as a bridge between persistent chain and travel chain, for vehicles within communication range. Besides, RSUs have stable network connections and sufficient computational resources.
- Vehicle: Each Vehicle is equipped with a trusted platform module (TPM) which is able to keep secret information protected and a physical unclonable function (PUF) to provide unique hardware fingerprint. (The technical details of TPM and PUF are discussed in [7] and [24], respectively, and are beyond the scope of this work). Compared to RSUs, vehicles have limited computational resources. Besides, in a platoon, a vehicle either acts as a leader vehicle (LV) or follower vehicle (FV). The LV is selected to lead the platoon, maintaining the stability of travel chain. The FV hands over driving control to LV and acts as a supervisor by storing necessary transactions and corresponding data for reporting after the platoon breakup.

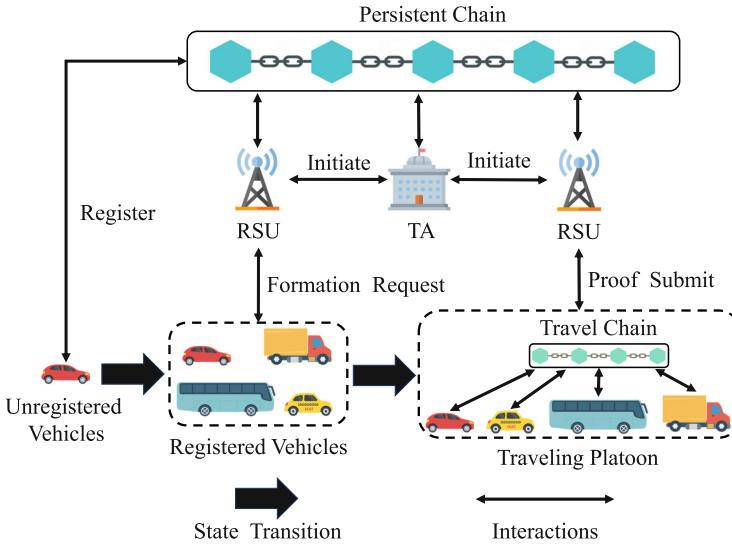


Fig. 1. The system architecture of the LBVP scheme.

3.2 Attack Model

Similar to many recent researches [22, 23], we assume that the TA is fully trusted and will not collude with the other entities. Besides, the TA maintains a secure database which can securely store the vehicles' real identity information. Meanwhile, the RSUs are considered to be honest-but-curious, that is, they will honestly perform designed operations but they are curious about the private information of a vehicle. For example, they may attempt to reveal the real identities of a vehicle. While we assume that most vehicles are honest, there may be a few malicious ones that could launch attacks to destabilize the network.

3.3 Design Goals

Based on the aforementioned attack model, the basic goal of the proposed LBVP scheme is to provide a secure and efficient scheme for platoon management. Specifically, the design goals of the LBVP scheme including following three goals.

1. *Conditional Privacy Preservation:* In the proposed LBVP scheme, the real identity and key pair information of each vehicle should not be inferred by any entity other than the TA.
2. *Security:* The proposed LBVP scheme should be able to defend against multiple kinds of common attacks, including the impersonation, selfish mining, sybil, and long-range attack.
3. *Lightweight:* To achieve acceptable computation and communication overheads as well as enhance the practicality, the proposed LBVP scheme should be able to achieve high performance while requires low computational resources.

3.4 Formalized Symbols

The main symbols that will be used in the subsequent sections are described in Table 2. And $\sigma_*(data) = \text{Sign}(sk_*, data)$ can be verified in the manner of $\text{Verify}(pk_*, \sigma_*(data)) \in \{0, 1\}$ (0 indicates verification passes and 1 for the contrary). The notation —— represents the concatenation operation.

Table 2. Symbol description tables.

Symbol	Definition
ID_*, PID_*	Real identity and Pseudo identity of entity *
G	Selected ECC base points
pk_*, sk_*	Public key and private key of *
$\sigma_*(data)$	Signature about $data$ generated by entity *
$cert_*$	Certificate generated by TA for entity *
$Hash(*)$	Hash value of *
t_*	Timestamp that * happened
t_{start}^{reg}	Deadline for vehicle to start registration proof
t_{end}^{reg}	Deadline for vehicle to finish registration proof
v	Individual vehicle
l	A platoon with a size of N
LV	The leader vehicle of platoon l
FV_m	The m -th follower vehicle that follows LV in l
MV	Member vehicle in l , i.e. $MV = \{LV\} \cup \{FV_m FV_m \in l\}$
\mathbb{L}_*	The ledger held by entity *
Loc_*	Location information about *
M	Message
r	Random number
C	Consensus result, s.t. $C \in \{\text{true}, \text{false}\}$
BLK_*	Block generated in phase *
$Com_*(data)$	Commitment generated by entity * for data
f^l	Feedback about platoon
$List_*$	A list of information about *
$Resp$	Response from PUF with a specific challenge
Req_*^{phase}	Request from entity * for $phase$
$Report, Vote, Reply$	The data generated during consensus
$Opinion$	Indicate a MV agree or disagree with a received $Report$

4 Various Stages in The LBVP Scheme

In this part, we present the implementation of the LBVP scheme. For ease of understanding, only one platoon is introduced in this section. Figure 2 illustrates the workflow of the LBVP scheme.

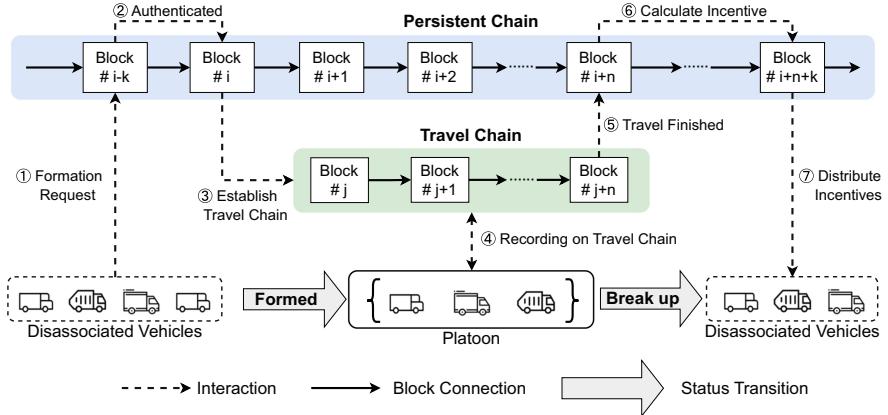


Fig. 2. Workflow of LBVP scheme.

4.1 Initiation

1. The TA initializes its own key pair $\langle pk_{TA}, sk_{TA} \rangle$ and selects the ECC base points G , generates genesis block for persistent chain, and then deploys vehicle registration smart contract on persistent chain. The TA will also initialize $\langle ID_{RSU}, pk_{RSU}, sk_{RSU}, cert_{RSU} \rangle$ and for RSUs which is owned and operated by government.
2. Once RSUs obtained their certificate, RSUs are enabled to communicate with the persistent chain network. After RSUs successfully joined the persistent chain network, RSUs must synchronize the current ledger \mathbb{L}_{TA} of persistent chain to the latest state. Upon completing the synchronization, RSUs deploy a smart contract that allows RSUs to submit travel chain data.

4.2 Vehicle Registration

When a vehicle v_i seeks to register with system, it must first submit its authentic ID_{v_i} to the TA. The v_i can only retrieve the key pair and certificate needed for platoon activities after it has completed the initial travel proof and the TA has verified ID_{v_i} . The whole process of vehicle registration is elaborated below:

1. The vehicle v_i requests the TA to start registration, and v_i first sends ID_{v_i} and a destination Loc_{dest} to the TA.
2. The TA needs review the identity information online and offline. Before starting to verify ID_{v_i} , the TA randomly selects two time stamps $\langle t_{start}^{reg}, t_{end}^{reg} \rangle$, having $t_{start}^{reg} < t_{end}^{reg}$, and also generates a list of RSUs' info $List_{RSU} = \{\langle Loc_{RSU_1}, pk_{RSU_1} \rangle, \langle Loc_{RSU_2}, pk_{RSU_2} \rangle, \dots, \langle Loc_{RSU_n}, pk_{RSU_n} \rangle\}$, along with a list of random number $List_r = \{r_1, r_2, \dots, r_n\}$ coordinated with RSUs list. If the reliability of the identity information is low, then the length of the initial travel proof will be higher. Then the TA sends $\langle t_{start}^{reg}, t_{end}^{reg}, List_{RSU}, List_r \rangle$ to v_i via a secured channel.

3. When v_i receives $< t_{start}^{reg}, t_{end}^{reg}, List_{RSU}, List_r >$ from the TA, it first extracts t_{start}^{reg} and t_{end}^{reg} as the valid proof time window, in which v_i needs to travel to different RSUs according to the designated order provided by $List_{RSU}$.
4. While v_i continuously reaches j -th RSU_j at time t_j , v_i calculates $Resp_j = PUF(t_j || r_j)$ as $sk_{v_i}^j$ and generates j -th public key $pk_{v_i}^j = Resp_j \cdot G$, then t_j and $pk_{v_i}^j$ are signed and submitted to the persistent chain by RSU_j . The v_i proceeds to RSU_{j+1} and repeats the above process until the final RSU_n around its destination Loc_{dest} . Once v_i finished the designated travel path, it gets a list of key pairs $\{< pk_{v_i}^1, Resp_1 >, < pk_{v_i}^2, Resp_2 >, \dots, < pk_{v_i}^n, Resp_n >\}$. All $pk_{v_i}^j$ are recorded on persistent chain and $sk_{v_i}^j = Resp_j$ are stored by v_i locally.
5. The TA generates a $cert_{v_i}$ including a timestamp t_{cert} , a PID_{v_i} and a $pk_{v_i}^j$ randomly selected from the initial travel proof of v_i . Then, the TA stores $< ID_{v_i}, List_r, pk_{v_i} >$, signs $cert_{v_i}$ with $\sigma_{TA}(cert_{v_i})$, and publishes $< pk_{v_i}, cert_{v_i}, \sigma_{TA}(pk_{v_i}) >$ to persistent chain.
6. Once got the $\sigma_{TA}(pk_{v_i})$ and $cert_{v_i}$ on the persistent chain through the help of RSUs, v_i stores $< pk_{v_i}, \sigma_{TA}(pk_{v_i}), cert_{v_i} >$ in its TPM, and is able to use sk_{v_i} access the system for platoon services.

4.3 Platoon Formation

Whenever a platoon is formed, a travel chain is built to collect and process all information (e.g. platoon operation and environment information) during the journey of platoon. The proposed LBVP consensus algorithm is employed on travel chain.

1. First, v_i submits platoon formation request $Req_{v_i}^{form}$ including unique requirements, e.g. current location $Loc_{v_i}^{cur}$, destination $Loc_{v_i}^{dest}$, desired arrival time $t_{arrival}$, expected time to join platoon t_{join} , and the signature of requirements. After received $< Req_{v_i}^{form}, \sigma_{v_i}(Req_{v_i}^{form}) >$, the platoon formation contract running on the persistent chain is triggered.
2. The platoon formation contract selects a LV and groups MV into one platoon l based on the requirements specified in the submitted $Req_{v_i}^{form}$. Then RSUs send the result $< List_{MV}, \sigma_{CS}(List_{MV}) >$ to the LV , the $List_{MV}$ in which is a sequence of MV 's information $< PID_{MV}, pk_{MV}, Req_{MV}^{form} >$ (the method for forming the platoon based on interest vectors and the method for selecting the leader vehicle has been implemented in TPSQ [10] and RPPM [16], respectively).
3. Upon receiving the formation result, LV must send the result to all FV based on $List_l$. After FV has received and verified the information of LV , it replies $< t_{form}, \sigma_{v_i}(t_{form}) >$ to join l and follows the guidance of LV .
4. Once all FV have successfully joined, LV generates and records a unique genesis block BLK_{form} in the ledger \mathbb{L}_{LV} . The BLK_{form} is then broadcasted to all FV . Each FV writes BLK_{form} into its local ledger \mathbb{L}_{FV} to finish the synchronization of the travel chain state.

5. Finally, all MV submit $Hash(BLK_{form})$ along with $\sigma_{MV}(Hash(BLK_{form}))$ to persistent chain to mark the successful establishment of the travel chain. Upon receiving these submitted signatures, the RSUs will temporarily place $cert_{MV}$ into a traveling list, which indicates MV already joined a platoon cannot simultaneously join multiple platoon. The $cert_{MV}$ in traveling list will only be refreshed after MV has completed the platoon breakup process which is elaborated in Sect. 4.5.

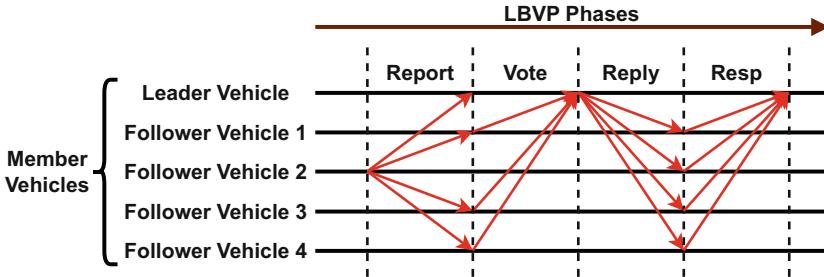


Fig. 3. Consensus pattern of the LBVP consensus algorithm with 1 leader vehicle and 4 follower vehicles in the same platoon.

4.4 Platoon Traveling

During platoon traveling, the MV exchange information (e.g. emergent event, traveling data recording and authentication) with the travel chain collaboratively maintained. Meanwhile, being inspired by [28, 32], we design Proof of Platoon Travel (PoPT), which makes MV need not to submit raw data or encrypted data to blockchain during traveling. Instead, MV only need to submit hash of BLK_{travel} . In other words, PoPT serves as a heartbeat mechanism between the persistent chain and the travel chain. It helps persistent chain updating the platoon travel status using the activity status of the travel chain. If no heartbeat for an extended period, it may indicate potential issues such as network disconnections, member dropouts, or malicious behavior. Additionally, the travel proofs from multiple MV can not only substantiate that MV have been traveling together for a specific period and enhance the verification correctness and of the travel chain ledgers.

Besides, we propose the LBVP consensus algorithm. The core idea of LBVP is to further simplify interaction complexity to achieve lower consensus latency. The consensus pattern of LBVP is illustrated in Fig. 3. During platoon traveling, a significant amount of M are collected, such as operational data related to platoon cruising and environmental data captured by sensors on each MV . Whenever a MV collects higher-priority or sufficient M , it can trigger a new round of consensus. The specific workflow of the LBVP algorithm is illustrated in Algorithm 1, and is delineated as follows.

1. Initially, in first line, MV is being verified whether its $cert_{MV}$ is validated and belongs to the platoon. And at the every beginning of a new consensus round, every member in platoon l is in the report stage.
2. As shown from lines 3 to 6, if MV is trying to report a new message M , it generates a report containing M , its signature of $\sigma_{MV}(Report)$, and a timestamp t_{Report} . MV will broadcast this $Report$ to whole platoon and enter reply stage to wait for $Reply$ from LV .
3. Lines 7 to 14 show the MV who received $Report$ first had to verify it. If verification passed, MV appends the report to received reports list. Then if MV is FV , the MV should send a $Vote$ containing the M , $Hash(M)$, its *Opinion* about the authenticity of M , and its own signature $\sigma_{MV}(Vote)$ to LV , then enter reply stage and wait for $Reply$ from LV . On the contrary, if MV is LV , it simply enters vote stage. But if verification is not passed, MV simply drops the $Report$.
4. During the vote stage, as shown in lines 16 to 27, LV verifies the received $Vote$ and appends them to its received vote list if the sender's signature is valid. Then LV checks whether the count of $Vote$ with agreement reaches the threshold $\frac{2N}{3}$ (N is the size of l). If the conditions are met, set C is set to true, and a $Reply$ is generated and broadcasted to FV . If the threshold is not met, LV waits for the remaining votes.
5. In the reply stage, as shown in lines 29 to 33, each FV verifies the $Reply$ received from LV . If the $Reply$ is valid and the C is true, BLK_{travel} is appended to the ledger \mathbb{L}_{FV} ; otherwise, it is discarded and FV return to report stage.
6. Finally in the response stage, from lines 35 to 38, each FV sends a $Response$ back to LV indicating that the $Reply$ has been successfully received, and submit $Hash(BLK_{travel})$ to the persistent chain as part of their PoPT.

4.5 Platoon Breakup

When a platoon finishes traveling, it follows the two-phase protocol shown below. In the first phase, upon the platoon's arrival at its destination, it enters the pre-breakup phase, which is essential for preparing the breakup operation.

A. Pre-breakup. Before all platoon members return to a free-driving state, each member needs to complete the following tasks:

1. Each FV generates $Req_{FV}^{breakup} = < Hash(\mathbb{L}_{FV}), Com_{FV}(f^l), t_{FV}^{upload} >$. The t_{FV}^{upload} is the time when FV should upload the f^l and the ledger \mathbb{L}_{FV} to the persistent chain. And $Com_{FV}(f^l)$ is the commitment of f^l generated by FV to assure others that FV will provide a f^l to the persistent chain in the future. Then FV should submit the request along with its signature $< Req_{FV}^{breakup}, \sigma_{FV}(Req_{FV}^{breakup}) >$ to the LV .

Algorithm 1. LBVP Consensus Algorithm.

Input : Member vehicle MV
Output: Consensus result C

```

1: if  $cert_{MV}$  is not validated or  $MV \notin l$  then return false // Report Stage
2: if  $MV$  is the vehicle to report  $M$  then
3:    $\sigma_{MV}(Report) = Sign(sk_{MV}, Hash(M)||t_{Report});$ 
4:    $Report \leftarrow < M, Hash(M), t_{Report}, \sigma_{MV}(Report) >;$ 
5:   Broadcast( $l$ ,  $Report$ );
6: else if  $MV$  is the vehicle that to receive  $Report$  then
7:   if Verify( $pk_{sender}$ ,  $\sigma_{MV}(M)$ ) then
8:     Append( $List_{Report}$ ,  $Report$ );
9:     if  $MV$  is  $FV$  then
10:        $\sigma_{MV}(Vote) = Sign(sk_{MV}, Hash(M)||Hash(Opinion)||t_{vote});$ 
11:       Vote
12:        $\leftarrow < M, Hash(M), Opinion, Hash(Opinion), t_{vote}, \sigma_{MV}(Vote) >;$ 
13:       Send( $LV$ ,  $Vote$ );
14:     else Drop( $Report$ )
15:   // Vote Stage
16:   if  $MV$  is  $LV$  and Verify( $pk_{sender}$ ,  $\sigma_{MV}(Vote)$ ) then
17:     Append( $List_{Vote}$ ,  $Vote$ );
18:     if Count( $List_{Vote}.Opinion == 1$ )  $\leq 2N/3$  then
19:       Wait for remaining  $Vote$ 
20:     else
21:        $C \leftarrow true;$ 
22:        $BLK_{travel} = < C, M, Hash(M), List_{Vote}, Hash(List_{Vote}), t_{Reply} >;$ 
23:        $\sigma_{TA}(Reply) = Sign(sk_{TA}, C||Hash(M)||Hash(List_{Vote})||t_{reply});$ 
24:       Reply  $\leftarrow < BLK_{travel}, \sigma_{TA}(Reply) >;$ 
25:       Broadcast( $FV$ ,  $Reply$ );
26:     end
27:   else Drop( $Vote$ ) // Reply Stage
28:   foreach  $MV \in FV$  do
29:     if Verify( $pk_{LV}$ ,  $\sigma_{TA}(Vote)$ ) and  $C == true$  then
30:       Append( $\mathbb{L}_{MV}$ ,  $BLK_{travel}$ )
31:     else Drop( $Reply$ )
32:   end
33:   // Response Stage
34:   foreach  $MV \in FV$  do
35:     Send( $LV$ ,  $Response$ );
36:     Send( $RSU$ ,  $Hash(BLK_{travel})$ );
37:   end

```

2. Upon receiving $Req_{FV}^{breakup}$ from all FV , LV packages these $Req_{FV}^{breakup}$ and produces a breakup block $BLK_{breakup}$, which is then broadcasted back to all FV .
3. When FV receives $BLK_{breakup}$ and writes it into \mathbb{L}_{FV} , it is permitted to leave the platoon. Before really leaving the platoon, FV are required to submit

one last $Req_{FV}^{leave} = < t_{FV}^{leave}, \sigma_{FV}(Hash(BLK_{breakup}) || t_{FV}^{leave}) >$ for LV to generate a leave confirmation block BLK_{leave} , which is linked to $BLK_{breakup}$ or latest BLK_{leave} to enhance security. At the same time that BLK_{leave} is written into the travel chain, the vehicle will lose its authorization to interact with the travel chain.

At the end of the pre-breakup phase, FV is able to leave the platoon and revert to free-driving state. The LV can only return to free-driving mode when all FV have departed. However, from the perspective of the persistent chain, the platoon remains in a breakup pending status, as the breakup state of the platoon has not yet been synchronized from the travel chain to the persistent chain, even though MV have already disconnected. Therefore, all MV need to complete the following post-breakup phase operations to fully conclude the breakup of the platoon and refresh their own $cert_{MV}$, thereby preparing them to participate in next platoon journeys.

B. Post-breakup. In order to truly finish the breakup of the platoon in the persistent chain, the following tasks must be done by every MV :

1. Each MV packages and submits $< f^l, \mathbb{L}_{MV} >$ to persistent chain via a RSUs.
2. From the moment the first MV submits ledger data, the RSUs will initialize a block counter based on the length of the PoPT of the remaining MV . This counter will decrease as new blocks are added to the persistent chain. If the counter reaches zero and the remaining MV of corresponding platoon l has not submitted their data, they will be penalized.
3. The RSUs complete the verification by comparing the block hash values in the PoPT with the actual block hash values calculated from \mathbb{L}_{MV} . Verification is passed only when the PoPT matches the actual block data.
4. Once the data is verified, the $cert_{MV}$ are retrieved from traveling list and renewed. If MV wants to form a platoon, it first needs to retrieve the newest $cert_{MV}$. In other words, MV cannot attend platoon anymore, unless MV submit $< f^l, \mathbb{L}_{MV} >$ in time.

5 Security Analysis

The LBVP scheme can resist several attacks, such as the impersonation attack, the selfish mining attack, the sybil attack, the modification attack and the long-range attack. Below, we explain why these attacks are not feasible within the LBVP scheme.

5.1 Impersonation Attack

An impersonation attack involves the unauthorized use of another vehicle's identity information for fraudulent activities. Combining PUF and TPM effectively defends against this threat. If one attacker attempts to steal other vehicle's

identity, the first approach is to obtain all the contents stored in the TPM. The other approach is to perform a modeling attack on PUF and acquire the random number sequence provided by the TA during the registration phase for the corresponding vehicle. These tasks are extremely difficult for a malicious attacker to accomplish. Therefore, communication using a fake or duplicate identity is not possible; only legitimate and authenticated vehicles are allowed to interact with the network.

5.2 Selfish Mining Attack

A selfish mining attack occurs if an attacker generates a legitimate block with specific information in advance but withholds it from the blockchain network. When the attacker wants to overwrite another legitimate block, it immediately releases the pre-generated block. However, as Algorithm 1 illustrates, blocks are written into the blockchain only when the majority of vehicles concur.¹ Once consensus is achieved, the block is immediately written, hence no vehicle can withhold the block without publishing it. If the attacker plays the role of the follower vehicle, it cannot generate a legitimate block by itself, since follower vehicle can only vote for new message. Furthermore, if the attacker is the leader vehicle, it cannot generate a legitimate block without enough agreement from follower vehicles in the platoon.

5.3 Sybil Attack

A sybil attack occurs when an attacker creates multiple identities or nodes to gain a disproportionate influence over a network, undermining its integrity, and potentially manipulating consensus algorithms. However, in the design of the LBVP scheme, a vehicle's legitimate identity must first be authorized by the TA and then vehicle must complete the initial travel proof designated by the TA. For honest vehicles that require system services, registration is straightforward. For attackers, however, creating multiple identities to influence network activities is challenging, as the cost increases with the number of identities created due to the implementation of initial travel proof.

5.4 Modification Attack

A modification attack involves an unauthorized alteration of data or messages being transmitted between parties. This type of attack can be used to change the information being exchanged, such as modifying the details of a transaction in a block. We have proposed PoPT based on the tamper-resistant characteristics of blockchain to defend against this attack. By recording and comparing the hash value of blocks in ledger from different vehicles and performing cross-validation, it can ensure the integrity and credibility of the data of travel chain from its creation to its merging onto the persistent chain.

5.5 Long-Range Attack

A long-range attack occurs when an attacker retrospectively constructs an alternative fork starting from a historical block, aiming to overwrite the current transaction history and compromise the ledger's integrity. Due to the use of the PoPT in the LBSV scheme, the number of viable historical fork points that an attacker can create becomes very limited, as they must start the attack from blocks for which the corresponding PoPT has not yet been generated.

6 Simulation Evaluation

In this section, we first introduce the experiment setting. Then we perform a quantitative evaluation about the computation overheads of five schemes, namely the EPoW scheme [4] scheme, the PBFT [2] scheme, the LBSV [31] scheme, the R-PBFT [12] scheme, and the LBVP scheme.

6.1 Experiment Setting

The performance evaluation is carried out on two experimental environment including an amd64 server and an arm development boards. A resource-rich server that equipped with an AMD Ryzen5 3600 CPU featuring 6 cores, 16GB of RAM, was used to ensure that the performance of the consensus algorithms was not limited by hardware constraints. In contrast, a resource-constrained environment featuring a 4 cores Arm Cortex-A57 chip, 4GB of RAM, was utilized to evaluate the performance in restricted environment.

6.2 Latency Comparison

The experimental results are presented in Fig. 4, illustrating the consensus latency between the LBVP scheme and other existing schemes. clearly demonstrate the advantages of LBVP over other schemes. Across varying numbers of vehicles, the LBVP algorithm consistently exhibits lower latency, indicating superior performance and efficiency. Especially when there are more than 10 vehicles, LBVP shows a significantly reduced latency compared to EPoW and PBFT, while also outperforming LBSV and R-PBFT. The consensus latency of the LBVP scheme remains consistent as the number of vehicles increases.

6.3 Memory Consumption Comparison

As illustrated in Fig. 5, the LBVP scheme significantly reduces memory consumption compared to PBFT, LBSV, and R-PBFT. Under resource-rich conditions, LBVP consistently shows lower memory usage as the number of vehicles increases. Similarly, in resource-constrained environments, LBVP maintains its memory efficiency advantage across varying vehicle numbers. Although EPoW shows the lowest memory consumption overall, LBVP strikes a better balance between memory efficiency and consensus latency, making it more suitable for scenarios with a substantial number of vehicles and stringent memory constraints.

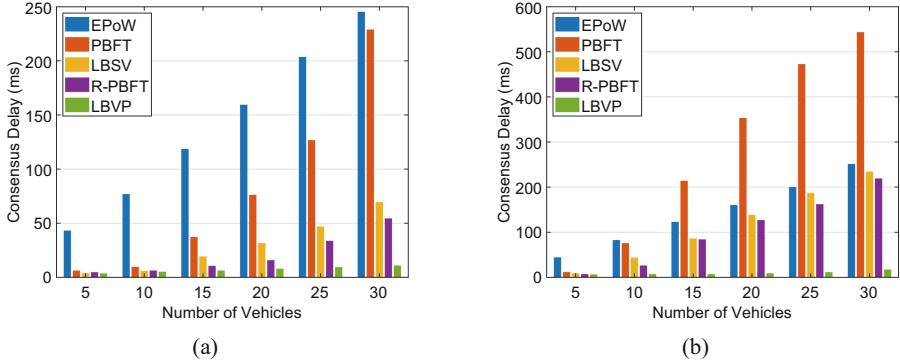


Fig. 4. Consensus latency comparison in EPoW, PBFT, LBSV, R-PBFT, and LBVP schemes in (a) resource-rich and (b) resource-constrained environment with the number of vehicles increases from 5 to 30 in increments of 5.

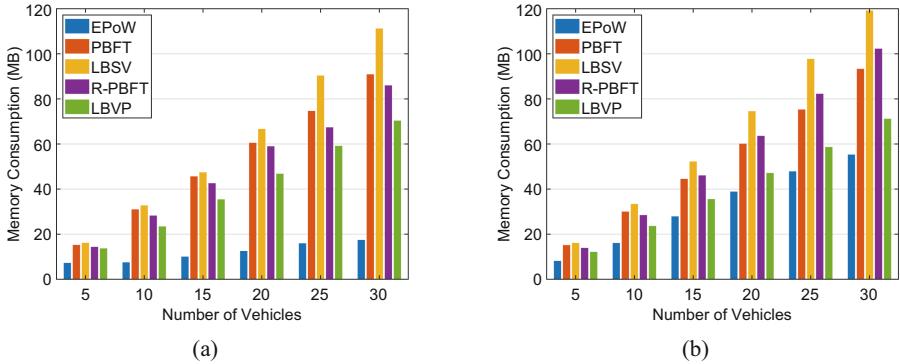


Fig. 5. Memory consumption comparison in EPoW, PBFT, LBSV, R-PBFT, and LBVP schemes in (a) resource-rich and (b) resource-constrained environment with the number of vehicles increases from 5 to 30 in increments of 5.

6.4 CPU Utilization Comparison

The experimental results comparing the CPU utilization are illustrated in Fig. 6. The results highlight the superior efficiency of the LBVP scheme. In both scenarios, LBVP consistently exhibits the lowest CPU utilization, significantly outperforming other schemes such as EPoW, PBFT, LBSV, and R-PBFT. Specifically, while EPoW and LBSV demonstrate the highest CPU utilization, indicating substantial resource consumption, LBVP's markedly lower CPU utilization underscores its effectiveness in optimizing resource usage. This consistent performance across different resource conditions underscores the robustness and efficiency of the LBVP scheme, making it a highly advantageous approach for systems aiming to enhance resource efficiency and overall performance.

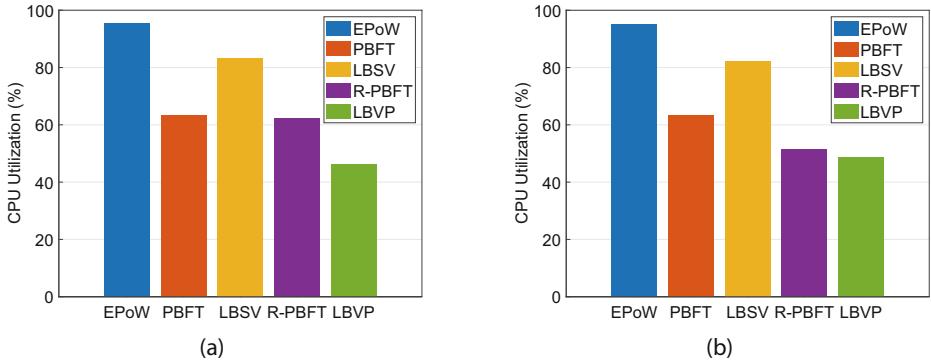


Fig. 6. Memory consumption comparison in EPoW, PBFT, LBSV, R-PBFT, and LBVP schemes in (a) resource-rich and (b) resource-constrained environment.

7 Conclusion

In this paper, we proposed the LBVP scheme to address the computational overheads and integration challenges of blockchain technology in vehicular networks. By introducing a multi-chain architecture with a persistent chain and multiple travel chains, and utilizing the PoPT protocol, we significantly reduced the interaction burden on edge devices while maintaining up-to-date status of travel train on the persistent chain. This approach enhances network performance, scalability, and security. Additionally, the lightweight consensus algorithm optimized interaction phases, resulting in notable reductions in consensus latency, memory consumption, and CPU utilization. Our comprehensive analysis and experimental results demonstrate that the LBVP scheme not only effectively preserves vehicle and platoon security but also resists various common attacks. Compared to existing schemes, the LBVP scheme achieved an average of 63% lower latency, 21% less memory consumption, and 33% lower computational cost. In future work, we will enhance the LBVP scheme by developing blockchain-based reputation and incentive mechanisms to improve security and privacy, ensuring the timely removal of malicious vehicles and achieving a higher level of integration.

Acknowledgments. We would like to express our sincere gratitude to the editor and anonymous reviewers for valuable feedback and constructive suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant 62032025, Grant 61932010, Grant 62272195, and Grant 62272203; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012776; and in part by the Guangxi Key Laboratory of Trusted Software under Grant KX202303.

References

1. Alladi, T., Chamola, V., Sahu, N., Venkatesh, V., Goyal, A., Guizani, M.: A comprehensive survey on the applications of blockchain for securing vehicular networks. *IEEE Commun. Surv. Tutor.* **24**(2), 1212–1239 (2022)
2. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation, OSDI 1999, pp. 173–186. USENIX Association, USA (1999)
3. Dibaei, M., et al.: Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: a survey. *IEEE Trans. Intell. Transp. Syst.* **23**(2), 683–700 (2022)
4. Du, G., et al.: A blockchain-based trust-value management approach for secure information sharing in internet of vehicles. *IEEE Internet Things J.* **1** (2023)
5. Dwivedi, S.K., Amin, R., Das, A.K., Leung, M.T., Choo, K.K.R., Vollala, S.: Blockchain-based vehicular ad-hoc networks: a comprehensive survey. *Ad Hoc Netw.* **137**, 102980 (2022)
6. Ghosal, A., et al.: Truck platoon security: state-of-the-art and road ahead. *Comput. Netw.* **185**, 107658 (2021)
7. Guette, G., Bryce, C.: Using TPMs to secure vehicular ad-hoc networks (VANETs). In: Onieva, J.A., Sauveron, D., Chaumette, S., Gollmann, D., Markantonakis, K. (eds.) WISTP 2008. LNCS, vol. 5019, pp. 106–116. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79966-5_8
8. Hbaieb, A., Ayed, S., Chaari, L.: A survey of trust management in the internet of vehicles. *Comput. Netw.* **203**, 108558 (2022)
9. Hou, B., Zhu, H., Xin, Y., Wang, J., Yang, Y.: MPoR: a modified consensus for blockchain-based internet of vehicles. *Wirel. Commun. Mob. Comput.* **2022**, e1644851 (2022)
10. Hu, H., Lu, R., Zhang, Z.: TPSQ: trust-based platoon service query via vehicular communications. *Peer-to-Peer Netw. Appl.* **10**(1), 262–277 (2017)
11. Kara, M., Laouid, A., Hammoudeh, M., AlShaikh, M., Bounceur, A.: Proof of chance: a lightweight consensus algorithm for the internet of things. *IEEE Trans. Industr. Inf.* **18**(11), 8336–8345 (2022)
12. Kumar, A., Vishwakarma, L., Das, D.: R-PBFT: a secure and intelligent consensus algorithm for Internet of vehicles. *Veh. Commun.* **41**, 100609 (2023)
13. Lao, L., Dai, X., Xiao, B., Guo, S.: G-PBFT: a location-based and scalable consensus protocol for IoT-blockchain applications. In: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 664–673 (2020)
14. Lesch, V., Breitbach, M., Segata, M., Becker, C., Kounev, S., Krupitzer, C.: An overview on approaches for coordination of platoons. *IEEE Trans. Intell. Transp. Syst.* **23**(8), 10049–10065 (2022)
15. Li, Q., Chen, Z., Li, X.: A review of connected and automated vehicle platoon merging and splitting operations. *IEEE Trans. Intell. Transp. Syst.* **23**(12), 22790–22806 (2022)
16. Li, R., et al.: RPMP: a reputation-based and privacy-preserving platoon management scheme in vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **1**–14 (2023)
17. Li, X., Jing, T., Li, R., Li, H., Wang, X., Shen, D.: BDRA: blockchain and decentralized identifiers assisted secure registration and authentication for VANETs. *IEEE Internet Things J.* **10**(14), 12140–12155 (2023)
18. Lin, C., He, D., Huang, X., Kumar, N., Choo, K.K.R.: BCPPA: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **22**(12), 7408–7420 (2021)

19. Lin, C., Huang, X., He, D.: EBCPA: efficient blockchain-based conditional privacy-preserving authentication for VANETs. *IEEE Trans. Dependable Secure Comput.* **20**(3), 1818–1832 (2023)
20. Liu, H., Han, D., Li, D.: Behavior analysis and blockchain based trust management in VANETs. *J. Parallel Distrib. Comput.* **151**, 61–69 (2021)
21. Liu, Z., et al.: PPRU: a privacy-preserving reputation updating scheme for cloud-assisted vehicular networks. *IEEE Trans. Veh. Technol.* 1–16 (2023)
22. Liu, Z., et al.: PPTM: a privacy-preserving trust management scheme for emergency message dissemination in space-air-ground integrated vehicular networks. *IEEE Internet Things J.* 1 (2021)
23. Liu, Z., et al.: TCEMD: a trust cascading-based emergency message dissemination model in VANETs. *IEEE Internet Things J.* **7**(5), 4028–4048 (2020)
24. Maes, R., Verbauwhede, I.: Physically unclonable functions: a study on the state of the art and future research directions. In: Sadeghi, A.R., Naccache, D. (eds.) *Towards Hardware-Intrinsic Security: Foundations and Practice*, pp. 3–37. Springer, Heidelberg (2010)
25. Maiti, S., Winter, S., Kulik, L., Sarkar, S.: Ad-hoc platoon formation and dissolution strategies for multi-lane highways. *J. Intell. Transp. Syst.* **27**(2), 161–173 (2021)
26. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
27. Pi, D., et al.: Automotive platoon energy-saving: a review. *Renew. Sustain. Energy Rev.* **179**, 113268 (2023)
28. Suo, D., Mo, B., Zhao, J., Sarma, S.E.: Proof of travel for trust-based data validation in V2I communication. *IEEE Internet Things J.* **10**(11), 9565–9584 (2023)
29. Taiyaba, Ms., Akbar, M.A., Qureshi, B., Shafiq, M., Hamza, M., Riaz, T.: Secure V2X environment using blockchain technology. In: Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering, EASE 2020, pp. 469–474. Association for Computing Machinery, New York (2020)
30. Tandon, R., Verma, A., Gupta, P.K.: D-BLAC: a dual blockchain-based decentralized architecture for authentication and communication in VANET. *Expert Syst. Appl.* **237**, 121461 (2024)
31. Vishwakarma, L., Nahar, A., Das, D.: LBSV: lightweight blockchain security protocol for secure storage and communication in SDN-enabled IoV. *IEEE Trans. Veh. Technol.* **71**(6), 5983–5994 (2022)
32. Wan, Z., Zhou, Y., Ren, K.: Zk-AuthFeed: protecting data feed to smart contracts with authenticated zero knowledge proof. *IEEE Trans. Dependable Secure Comput.* **20**(2), 1335–1347 (2023)
33. Wang, E.K., Sun, R., Chen, C.M., Liang, Z., Kumari, S., Khurram Khan, M.: Proof of X-repute blockchain consensus protocol for IoT systems. *Comput. Secur.* **95**, 101871 (2020)
34. Xu, G., Bai, H., Xing, J., Luo, T., Xiong, N.N., Cheng, X., Liu, S., Zheng, X.: SG-PBFT: a secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **164**, 1–11 (2022)
35. Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.C.M.: Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet Things J.* **6**(2), 1495–1505 (2019)
36. Zhang, C., et al.: TPPR: a trust-based and privacy-preserving platoon recommendation scheme in VANET. *IEEE Trans. Serv. Comput.* **15**(2), 806–818 (2022)

37. Zhang, D., Shi, W., St-Hilaire, M., Yang, R.: Multiaccess edge integrated networking for internet of vehicles: a blockchain-based deep compressed cooperative learning approach. *IEEE Trans. Intell. Transp. Syst.* **23**(11), 21593–21607 (2022)
38. Zhang, J., Jiang, Y., Cui, J., He, D., Bolodurina, I., Zhong, H.: DBCPA: dual blockchain-assisted conditional privacy-preserving authentication framework and protocol for vehicular ad hoc networks. *IEEE Trans. Mob. Comput.* 1–15 (2022)
39. Zhu, S., Cai, Z., Hu, H., Li, Y., Li, W.: zkCrowd: a hybrid blockchain-based crowdsourcing platform. *IEEE Trans. Industr. Inf.* **16**(6), 4196–4205 (2020)



Low-Carbon Geographically Distributed Cloud-Edge Task Scheduling

Yingjie Zhu^{1(✉)}, Ji Qi², Zehao Wang¹, Shengjie Wei¹, Yan Chen¹, Tuo Cao¹, Gangyi Luo², and Zhuzhong Qian^{1(✉)}

¹ State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, China

{yingjiezhu,zehawang,mg20330069,dz21330047,tuocao}@smail.nju.edu.cn,
qzz@nju.edu.cn

² China Mobile (Suzhou) Software Technology Co., Ltd., Suzhou, Jiangsu, China
{qiji,luogangyi2}@cmss.chinamobile.com

Abstract. Edge computing is a rapidly developing research area known for its ability to reduce latency and improve energy efficiency, and it also has a potential for green computing. Many geographically distributed edge servers are powered by renewable energy sources, due to the difficulties of using traditional power supplies or because of advancements in energy harvesting technologies. These green edge servers can cut down carbon emissions by processing tasks locally, but the inherent limitations of their computing capacity result in some tasks having to be uploaded to a data center to meet service-level agreement (SLA) requirements. To further reduce carbon emissions in cloud-edge systems, scheduling tasks to those low-carbon data centers while meeting latency constraints is highly beneficial. In this paper, we propose a low-carbon cloud-edge scheduling algorithm that utilizes Lyapunov optimization techniques and Markov approximation to address the long-term optimization problem of carbon emissions. Our algorithm guarantees provable performance, and simulation results demonstrate its effectiveness in striking a balance between carbon emissions and task latency.

Keywords: Low-carbon · Geo-distributed data centers · Edge-Cloud collaboration · Task scheduling

1 Introduction

Edge computing entails a decentralized computing architecture that strategically redistributes data storage and processing capabilities from centralized data centers to the network edge. By enabling on-premise execution of tasks on the edge server, it delivers low-latency and high-bandwidth services that facilitate real-time data analysis and processing. Furthermore, there are increasing scenarios

This work is funded by Nanjing University-China Mobile Communications Group Co., Ltd. Joint Institute.

where edge servers are powered entirely by clean energy, which can be leveraged to achieve sustainable and environmentally friendly computing practices. For one thing, in many remote regions, it is difficult for edge servers to directly use wired power supplies due to the inconvenience of grid construction or the high cost of access to power supply [1]. For another, the geographically distributed nature of edge nodes allows for the utilization of on-site green energy with the aid of energy harvesting technologies [2,3]. Additionally, edge service providers are increasingly committed to building green edge servers in order to minimize their carbon footprint. A notable example is Akamai, which has pledged to power their edge servers with 100% renewable energy by 2030 [4]. These green edge servers emit significantly lower levels of carbon, thereby contributing to a reduction in carbon emissions when performing tasks on them.

However, the limited computing capabilities of edge servers pose a challenge when dealing with a large influx of tasks, especially those requiring significant computational resources. In such cases, the edge server may struggle to meet the service-level agreement (SLA) requirements due to prolonged processing times. Consequently, certain tasks have to be uploaded to cloud data centers equipped with ample computing resources, albeit at the expense of increased carbon emissions and the introduction of transmission latency.

Clean energy has been extensively integrated into the power infrastructure of data centers, and has been proven effective in reducing carbon emissions per unit of electricity consumption [5]. Nonetheless, the temporal and spatial distribution uncertainty of clean energy sources causes significant disparities in carbon emission rates among different data centers. Existing research on reducing carbon emissions of data centers mainly focuses on energy consumption reduction strategies, such as redirecting workloads to more energy-efficient data centers [6], dynamically turning off idle servers [7], scaling server speed during low workload periods [8], and using virtual machine placement (VMP) approaches [9]. However, these energy-saving measures do not necessarily result in carbon emission reduction for geographically distributed data centers, as some highly energy-efficient data centers may still exhibit high carbon emission rates due to the carbon intensity of the local energy supply. [10]. In this context, scheduling computational tasks to low-carbon data centers can effectively reduce total carbon emissions, but this strategy may potentially introduce additional transmission latency, as low-carbon data centers are often located farther away. Consequently, the scheduling of computational tasks presents a complex trade-off between the imperative to minimize carbon emissions and the requirement to maintain acceptable levels of latency.

This study focuses on a practical scenario that edge servers are interconnected with multiple geo-distributed data centers. We introduce a task scheduling framework designed to harness the advantages of these distributed data centers and green edge servers, with the dual objectives of reducing overall carbon emissions and adhering to latency constraints. As illustrated in Fig. 1, computational tasks from users are first submitted to the edge servers, which then determine whether the tasks should be executed locally or uploaded to a specific data center for processing. We mathematically formulate the scheduling prob-

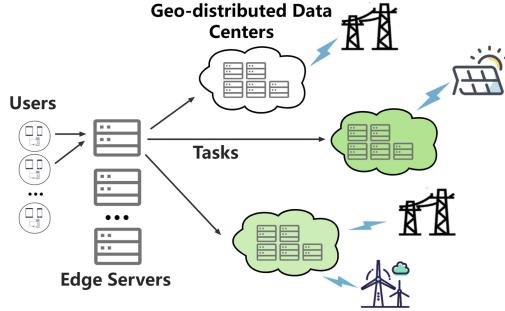


Fig. 1. An exemplification of the low-carbon geo-distributed cloud-edge task scheduling framework.

lem as a long-term optimization problem, aiming to minimize the cumulative carbon emissions, and the constraint is that the average processing time for each task remains within a predefined threshold, which reflects the system's SLA. To tackle this, we employ a Lyapunov-based online algorithm that decomposes the long-term problem into a series of single-time-slot sub-problems. Subsequently, we apply a Markov approximation-based algorithm to efficiently address each sub-problem within its respective time slot. Our proposed algorithm has theoretical performance guarantees, and we demonstrate the efficacy of our proposed algorithm through comprehensive simulations, highlighting its superior performance in reducing carbon emissions and task delay compared to alternative methodologies.

Our contributions in this work can be summarized as follows:

- We present a practical scheduling model of computational tasks in a scenario where edge servers are connected to geo-distributed data centers. By prioritizing carbon emission reduction as the primary goal rather than a constraint of the scheduling model, we significantly reduce the system's carbon emissions to the greatest extent possible. Instead of optimizing individual time slots, we mathematically formulate the scheduling problem as a long-term optimization objective to maximize the optimization of carbon emissions while ensuring the long-term stable operation of the system.
- We employ Lyapunov optimization techniques to derive a near-optimal policy that is independently of future information, and we design an efficient scheduling algorithm based on Markov approximation, which strikes a balance between acceptable time complexity and the preservation of solution effectiveness. Our methodologies are anchored by a robust theoretical framework, providing our proposed algorithm with a solid guarantee of theoretical soundness.
- We evaluate the performance of our algorithm through extensive simulations, and compare it with several benchmarks. These results suggest that our proposed algorithm has an advantage over other methods, and it is robust, adapt-

able, and effective in minimizing carbon emissions under a wide range of conditions.

The rest of this paper is structured as follows: Sect. 2 provides an overview of the related work. Section 3 presents the system model used in this study. Section 4 details the proposed algorithm. Section 5 showcases the results obtained from simulation experiments. Lastly, Sect. 6 concludes the paper.

2 Related Work

Some research has been conducted on reducing energy of edge computing system, but few studies have taken carbon emissions as the theme, and most of the efforts have focused on optimizing the cost at the edge by offloading strategies, ignoring the impact of data centers. Chen *et al.* [11] introduced an online peer offloading strategy that aims to optimize performance efficiency while considering long-term energy consumption constraints of edge servers. However, it is important to note that their peer offloading approach is specifically focused on interactions between edge servers. Xu *et al.* [1] used reinforcement learning to learn online offloading strategies, reducing long-term latency and operational costs in renewable energy-powered mobile edge systems, but they did not consider the impact of carbon emissions. In [12], Gu *et al.* proposed a virtual machine task migration strategy based on deep reinforcement learning to reduce carbon emissions in edge computing systems, but it only focuses on service tasks that can be migrated and deployed among edge servers, rather than computationally intensive tasks that consume more energy. The authors of [13] presented a resource-constrained randomized dependent rounding algorithm to enable offloading of machine learning tasks while operating within the constraints of limited carbon emission rights.

Geographical load balancing has also been explored to reduce system costs. However, most research efforts have focused on reducing energy consumption or electricity price. Liu *et al.* [14] distributed tasks to data centers with lower latency and electricity prices to reduce energy cost while ensuring latency requirements. Zhou *et al.* [10] experimentally demonstrated that saving energy cost due to spatial variations in carbon emission rates is not equivalent to reducing carbon emissions. They proposed a carbon emission-aware online control algorithm, COCA, based on Lyapunov optimization to minimize energy cost while satisfying carbon emission constraints. Lin *et al.* [15] further incorporated the uncertainty of on-site clean energy supply, but these works did not specifically model tasks and mainly considered request-type tasks with low computation and carbon emissions. Khosravi *et al.* [16] considered using a virtual machine placement strategy to reduce carbon emissions in multiple cloud data centers. However, it does not achieve flexible scheduling at the task level and primarily focuses on carbon pricing and carbon taxes, overlooking the temporal fluctuations in carbon emission rates of the power supply. The authors of [17] developed a renewable energy-aware multi-index job classification and scheduling methodology, which involves assigning workloads to data centers that possess an ample supply of renewable energy for efficient processing. However, these works only consider data center

task scheduling in the traditional cloud computing paradigm and are not applicable to the cloud-edge scenario addressed in this study.

Several prior studies have employed Lyapunov optimization algorithms to reduce system costs, such as energy and latency, in mobile edge systems. DREAM [18] used Lyapunov optimization algorithm to investigate the energy-delay trade-off of mobile clouds that encompass diverse types of computation tasks. Mao *et al.* [19] integrated energy harvesting techniques into mobile-edge computing and proposed a Lyapunov-based algorithm to reduce the service cost in edge servers. Zhang *et al.* [20] used Lyapunov optimization to achieve the optimal arrangement of CPU-cycle frequencies for mobile devices and power levels for data transmission.

3 System Model

Within this section, we present an overview of our proposed system model of computational tasks in a geo-distributed cloud-edge system. Our model is based on the following components:

3.1 Cloud-Edge Scheduling Model

In the context of a network consisting of N interconnected edge servers and M data centers, we define various parameters to characterize the system. The computation capacity of the i -th edge server, denoted as $MIPS_i^E$, quantifies its ability to execute instructions per unit time. Similarly, the computation capacity of single server in the j -th data center is expressed as $MIPS_j^D$, and its power rate is P_j^D . The carbon emission rate of the j -th data center at time slot t is presented as $C_j^D(t)$, and its power usage effectiveness is denoted as PUE_j^D . Furthermore, the network encompasses transmission connections between each edge server and data center, necessitating consideration of parameters such as the physical hop count R_{ij} (corresponding to the distance), the power consumption per hop router for transmitting unit data P_{ij}^{net} , and the overall carbon emission rate of the link denoted as C_{ij}^{net} .

In each time slot t , the arrival rate of tasks in i -th edge server is $a_i(t)$, which means there will be $A_i(t)$ computational tasks, each characterized by distinct computation and data requirements. The computation amount of a task is quantified by the number of clock cycles it necessitates, denoted as $I_{(i,k)}(t)$. The data amount $D_{(i,k)}(t)$ represents the number of bits needed for transmitting the task. Binary variables $X_{(i,k)j}(t)$ are used to indicate whether the k -th computing task on the i -th edge server is uploaded to the j -th data center for processing during time slot t . If $X_{(i,k)j}(t)$ is set to 0 for all data centers j , it signifies that the task is scheduled to be executed locally on the edge server. Hence, we can

formulate the following constraints to govern the task scheduling:

$$\sum_{j=1}^M X_{(i,k)j}(t) \leq 1, \quad (1)$$

$$X_{(i,k)j}(t) \in \{0, 1\}. \quad (2)$$

If the k -th task on the i -th edge server needs to be uploaded to the j -th data center, its bandwidth consumption can be denoted as $b_{(i,k)j}(t)$. All tasks uploaded from the i -th edge to the j -th data center share a common link with a total bandwidth limit denoted as $B_{ij}(t)$. To guarantee that the aggregate data volume of all transmitted tasks on this link remains within the available bandwidth, the following bandwidth constraint is formulated:

$$\sum_{k=1}^{A_i(t)} X_{(i,k)j}(t) \cdot b_{(i,k)j}(t) \leq B_{ij}(t). \quad (3)$$

3.2 Latency Model

The delay of the tasks on the i -th edge server, when processed locally, is predominantly influenced by the local computation time, which is represented as $T_i^E(t)$. As multiple tasks can be processed locally, they must share the computation capacity of the edge server. Consequently, the total delay of all tasks on the edge server can be derived using the following expression:

$$T_i^E(t) = \frac{\sum_{k=1}^{A_i(t)} (1 - \sum_{j=1}^M X_{(i,k)j}(t)) \times I_{(i,k)}(t)}{MIPS_i^E}. \quad (4)$$

When the k -th task on the i -th edge server is decided to be uploaded to the j -th data center, its delay can be denoted as $T_{(i,k)}^D(t)$ and the delay consists of two parts, transmission time and computing time:

$$T_{(i,k)}^D(t) = \sum_{j=1}^N X_{(i,k)j}(t) \times (TCal_{(i,k)j}(t) + TTrans_{(i,k)j}(t)). \quad (5)$$

The transmission time from the i -th edge node to the j -th data center depends on the data amount $D_{(i,k)}(t)$, physical hop count R_{ij} , and transmission bandwidth $b_{(i,k)j}(t)$, while the time for computing the task at the data center can be calculated by dividing computation amount by the server's computation capacity. Thus we can get two equations as follows:

$$TTrans_{(i,k)j}(t) = X_{(i,k)j}(t) \times R_{ij} \times \frac{D_{(i,k)}(t)}{b_{(i,k)j}(t)}, \quad (6)$$

$$TCal_{(i,k)j}(t) = X_{(i,k)j}(t) \times \frac{I_{(i,k)}(t)}{MIPS_j^D}. \quad (7)$$

Then we can get the average latency of all tasks in one slot:

$$T_A(t) = \frac{\sum_{i=1}^N (T_i^E(t) + \sum_{k=1}^{A_i(t)} T_{(i,k)j}^D(t))}{\sum_{i=1}^N A_i(t)}. \quad (8)$$

In order to meet the SLA requirements and sufficiently reduce long-term carbon emissions, we can tolerate a slightly larger average latency in a specific time slot, and simply require that the average delay across all time slots does not exceed the predefined value T_{avg} . Ultimately, the latency constraint can be formulated as follows:

$$\frac{1}{T} \sum_{t=1}^T T_A(t) \leq T_{avg}. \quad (9)$$

3.3 Carbon Emission Model

When the k -th task on the i -th edge server is selected to be uploaded to the j -th data center for processing, its carbon emissions comprise two main components: transmission carbon emissions $CETrans_{(i,k)j}(t)$ and computation carbon emissions $CECal_{(i,k)j}(t)$. Firstly, let's consider the computation carbon emissions generated by the data center's server when performing operations. The power consumption can be derived by multiplying the server's power rate P_j^D with computation time $TCal_{(i,k)j}(t) = \frac{I_{(i,k)}(t)}{MIPS_j^D}$. To obtain carbon emissions, we also need to multiply the energy utilization rate PUE_j^D and the carbon emission rate $C_j^D(t)$ of the data center, then we can get the equation as follows:

$$CECal_{(i,k)j}(t) = P_j^D \times \frac{I_{(i,k)}(t)}{MIPS_j^D} \times PUE_j \times C_j^D(t). \quad (10)$$

The transmission carbon emissions are determined by the data amount $D_{(i,k)}(t)$ and the number of physical hops R_{ij} that the transmission goes through. Similarly, we need to multiply the power consumption of each hop router to transmit unit data P_{ij}^{net} and the carbon emission rate C_{ij}^{net} . The formula is as follows:

$$CETrans_{(i,k)j}(t) = R_{ij} \times D_{(i,k)}(t) \times P_{ij}^{net} \times C_{ij}^{net}. \quad (11)$$

The sum of these two parts gives the carbon emissions generated by the k -th task when it is decided to be uploaded to the j -th data center for processing, so the total carbon emissions can be obtained by summing the carbon emissions of all tasks as follows:

$$SumCE(t) = \sum_{i=1}^N \sum_{k=1}^{A_i(t)} \left[\sum_{j=1}^M X_{(i,k)j}(t) \times (CECal_{(i,k)j}(t) + CETrans_{(i,k)j}(t)) \right]. \quad (12)$$

3.4 Long-Term Optimization Problem

Our goal is to minimize the carbon emissions generated by the system while ensuring delay and bandwidth requirement, therefore the optimization problem can be formulated as:

$$\begin{aligned} P_1 : \min & \quad \lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{t=1}^T \text{SumCE}(t) \\ \text{s.t.} & \quad (1) - (3), (9). \end{aligned} \quad (13)$$

4 Algorithm Design

Due to the two major challenges in problem P_1 : the long-term latency constraint and the NP-hardness of instantaneous decision-making, we employ an optimization algorithm design based on Lyapunov optimization [21] and Markov approximation. Firstly, we utilize Lyapunov functions to transform the long-term constraints and optimization objectives of the original problem into per-time-slot sub-problems. Then, we approximately solve the sub-problem in one time slot using the Markov approximation process.

4.1 Lyapunov-Based Online Algorithm

The optimization problem contains long-term optimization objectives and constraint inequalities, while the actual system does not know the situation of the subsequent time slots, and can only make decisions in the current time slot, which will lead to the potential violation of long-term constraints, or can not make full use of the elasticity space in a single time slot brought about by the long-term constraints, so it is necessary to firstly transform the original problem into per-time-slot decision sub-problems, which are then further solved.

To transform the original long-term optimization problem into per-time-slot sub-problems, we first define the Lyapunov virtual queue, denoted as $q(t)$, to represent the cumulative deviation between the delay at time slot t and the target constraint, with an initial value of 0 and a recursive equation of:

$$q(t+1) = \max(0, T_A(t) - T_{avg} + q(t)). \quad (14)$$

A larger value of the queue $q(t)$ indicates a greater cumulative deviation of the delay from the original constraint up to time slot t . Therefore, our objective is to control the size of this queue. For this purpose we further define a Lyapunov function, $L(q(t)) = \frac{1}{2}(q(t))^2$, which measures the degree of congestion of the queue, and in order to make this function evolve to smaller states, we introduce a Lyapunov drift function within a single time slot to represent the growth of the queues' backlog from time slot t to time slot $(t+1)$:

$$\Delta(q(t)) = E[L(q(t+1)) - L(q(t)) \mid q(t)]. \quad (15)$$

Since $\Delta(q(t))$ represents the growth of the Lyapunov queue within a single time slot, we can ultimately restrict the queue size by minimizing $\Delta(q(t))$. This ensures compliance with the constraints of the original problem. To simultaneously minimize the objective while satisfying the constraints as much as possible, we can minimize both the objective and $\Delta(q(t))$ by introducing a parameter V to control the optimization weight. As a result, we obtain the Lyapunov drift-plus-penalty term as $\Delta(q(t)) + V \cdot \text{SumCE}(t)$.

However, $\Delta(q(t))$ itself has a high computational complexity, and for this reason, we employ the Minimum Drift Plus Penalty algorithm from Lyapunov optimization theory, targeting the minimization of the upper bound on the drift plus penalty term, rather than directly minimizing the term itself. We first give the following lemma:

Lemma 1.

$$\Delta(q(t)) + V \cdot \text{SumCE}(t) \leq B + q(t) \cdot (T_A(t) - T_{avg}) + V \cdot \text{SumCE}(t), \quad (16)$$

where $B = \frac{1}{2}(T_{max} - T_{avg})^2$, $T_{max} = \max_{t \in T} T_A(t)$ represents the maximum worst-case delay within each time slot.

Proof. From Eq. (15), we have:

$$\begin{aligned} \Delta(q(t)) &= \frac{1}{2}E[q(t+1)^2 - q(t)^2 \mid q(t)] \\ &\leq \frac{1}{2}E[(q(t) + T_A(t) - T_{avg})^2 - q(t)^2 \mid q(t)] \\ &= \frac{1}{2}(T_A(t) - T_{avg})^2 + q(t)E[(T_A(t) - T_{avg}) \mid q(t)] \\ &\leq \frac{1}{2}(\max_{t \in T} T_A(t) - T_{avg})^2 + q(t) \cdot (T_A(t) - T_{avg}). \end{aligned} \quad (17)$$

Remember $T_{max} = \max_{t \in T} T_A(t)$ and $B = \frac{1}{2}(T_{max} - T_{avg})^2$, then we have:

$$\Delta(q(t)) + V \cdot \text{SumCE}(t) \leq B + q(t) \cdot (T_A(t) - T_{avg}) + V \cdot \text{SumCE}(t). \quad (18)$$

□

Therefore, we only need to minimize the expression $B + q(t) \cdot (T_A(t) - T_{avg}) + V \cdot \text{SumCE}(t)$ to obtain an approximate solution. Since B is a constant, the transformed instantaneous decision problem is as follows:

$$\begin{aligned} P_2 : \min \quad & q(t) \cdot T_A(t) + V \cdot \text{SumCE}(t) \\ \text{s.t.} \quad & (1) - (3). \end{aligned} \quad (19)$$

Finally, we obtain an approximate sub-problem P_2 of the original problem that can be solved instantaneously, and we can obtain an approximate solution to the original problem by solving this problem. Additionally, we can control the trade-off between latency and the overall carbon emissions by adjusting the parameter V . Algorithm 1 presents our Lyapunov-based online algorithm framework, and we will solve problem P_2 in the next subsection.

Algorithm 1. Lyapunov-based Online algorithm

Input: T_{avg}, T ;

- 1: $q(1) = 0$;
- 2: **for** $t = 1$ to T **do**
- 3: Update system parameters;
- 4: Use Algorithm to obtain scheduling strategy by solving P_2 ;
- 5: $q(t+1) = \max(0, T_A(t) - T_{avg} + q(t))$;
- 6: **end for**

4.2 Markov Approximation Based One-Slot Algorithm

For ease of expression, we denote $U(\alpha, \beta)$ as the objective function of P_2 where $\alpha = \{X_{(i,k)j}(t) \mid k \in \{1, \dots, A_i(t)\}, j \in \{1, \dots, M\}, i \in \{1, \dots, N\}\}$ represents a task scheduling strategy, and $\beta = \{b_{(i,k)j}(t) \mid k \in \{1, \dots, A_i(t)\}, j \in \{1, \dots, M\}, i \in \{1, \dots, N\}\}$ is the allocation of bandwidth. Notably, the constraint involving β is only reflected in the time constraint, and once α has been determined, each edge server can independently decide on the bandwidth allocation for its uploaded tasks. We can obtain β directly through the following theorem:

Theorem 1. *When α has been determined, the optimal choice for β to satisfy the bandwidth constraint is as follows:*

$$b_{(i,k)j}(t) = \frac{B_{ij} \cdot \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}}{\sum_{k=1}^{A_i(t)} \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}}. \quad (20)$$

Proof. It is straightforward to see that such a bandwidth allocation satisfies the bandwidth constraint, because:

$$\sum_{k=1}^{A_i(t)} X_{(i,k)j}(t) \cdot b_{(i,k)j}(t) = \frac{\sum_{k=1}^{A_i(t)} B_{ij} \cdot \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}}{\sum_{k=1}^{A_i(t)} \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}} = B_{ij}(t). \quad (21)$$

Secondly we prove its optimality. As shown in Sect. 3.2 on delay calculation, the bandwidth choice β only affects the transmission delay $TTrans_{(i,k)j}(t)$ of each task:

$$\begin{aligned} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{A_i(t)} TTrans_{(i,k)j}(t) &= \sum_{i=1}^N \sum_{j=1}^M R_{ij} \times \sum_{k=1}^{A_i(t)} \frac{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}{b_{(i,k)j}(t)} \\ &\geq \sum_{i=1}^N \sum_{j=1}^M R_{ij} \times \frac{(\sum_{k=1}^{A_i(t)} \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)})^2}{\sum_{k=1}^{A_i(t)} b_{(i,k)j}(t)} \\ &\quad (\text{By using the Cauchy-Schwarz inequality}) \\ &\geq \sum_{i=1}^N \sum_{j=1}^M R_{ij} \times \frac{(\sum_{k=1}^{A_i(t)} \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)})^2}{B_{ij}(t)}. \end{aligned} \quad (22)$$

In order to achieve its optimal value, β need to fulfill the condition of taking equality of two inequalities, so it can be calculated as:

$$b_{(i,k)j}(t) = \frac{B_{ij} \cdot \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}}{\sum_{k=1}^{A_i(t)} \sqrt{X_{(i,k)j}(t) \cdot D_{(i,k)}(t)}}. \quad (23)$$

And this is the value of β . Therefore, β is both feasible and optimal. \square

Thus, we are able to solve the optimization problem with single decision variable α . We will use the Markov approximation method to solve P_2 because it is still NP-hard for its combinatorial nature [22].

Algorithm 2. Markov Approximation Based One-slot Optimization Algorithm

Input: R_{max} ;

- 1: $\alpha = \{X_{(i,k)j}(t) = 0 \mid k \in \{1, \dots, A_i(t)\}, j \in \{1, \dots, M\}, i \in \{1, \dots, N\}\}$;
- 2: Calculate β using Eq.(20);
- 3: $count \leftarrow 0$
- 4: **repeat**
- 5: $count \leftarrow count + 1$
- 6: Randomly change α to α' ;
- 7: **if** α' is feasible **then**
- 8: Calculate β' using Eq.(20);
- 9: Calculate $U(\alpha, \beta)$, $U(\alpha', \beta')$;
- 10: $\eta \leftarrow \frac{1}{1+e^{(U(\alpha', \beta')-U(\alpha, \beta))/r}}$;
- 11: with probability η replace α , β with α' , β' ;
- 12: **end if**
- 13: **until** $count = R_{max}$ or there is no significant improvement for more than 10 iterations;

Output: α, β ;

The online algorithm (described in Algorithm 2) applies Markov chain approximation [23] to continuously update the task scheduling policy α and obtain a near-optimal solution. The algorithm works in the following way: firstly, at the beginning of each time slot, we simply put all incoming computation task on edge servers by using assignment $\alpha = \{X_{(i,k)j}(t) = 0 \mid k \in \{1, \dots, A_i(t)\}, j \in \{1, \dots, M\}, i \in \{1, \dots, N\}\}$. Secondly, we repeat the process as follows: we randomly change α to α' by selecting a task and changing its deployment (i.e., if it is scheduled to be uploaded to a data center, we can reschedule it on a different data center or reschedule it on the original edge server), then we can use Eq. (20) to obtain β' , and further calculate the new objective value $U(\alpha', \beta')$ in problem P_2 . calculate the probability of a state transition, denoted as η , based on the difference between objective values. In the current iteration, the new strategy replace the original one with probability η . Hence, the likelihood of changing the strategy is higher if the new strategy leads to a lower objective value. The parameter $r \geq 0$ (Line 10) is used to balance exploration and exploitation, a

bigger r will lead to a bigger probability η and thus encouraging accepting a new deployment, and when $r \rightarrow 0$, $\eta \rightarrow 0$, the algorithm will hardly accept new deployment. The iterative process is sustained until the criterion of R_{max} iterations is met or there is an absence of noteworthy enhancement in the objective value,(i.e., $|U(\alpha, \beta) - U(\alpha', \beta')| < 0.01$) for more than 10 iterations. As shown in [24], with the selection of appropriate parameters, the Markov approximation-based Algorithm is capable of achieving super-linear convergence.

4.3 Theoretic Analysis

Sine our algorithm uses a Lyapunov-based optimization framework and a Markov approximation based one-slot optimization algorithm, respectively, it has two main theoretical guarantees:

Theorem 2. *Our Lyapunov-based online algorithm implements the following performance constraints in terms of both the optimization objective of carbon emission rate and the constraint objective of queue stability:*

$$\lim_{T \rightarrow +\infty} \frac{\sum_{t=1}^T E[SumCE(t) | q(t)]}{T} \leq ce_{opt} + \frac{B}{V}, \quad (24)$$

$$\lim_{T \rightarrow +\infty} \frac{\sum_{t=1}^T E[q(t)]}{T} \leq \frac{B + V(ce_{max} - ce_{opt})}{\xi}. \quad (25)$$

where ce_{opt} is the theoretically optimal carbon emission value of the original problem P_1 , ce_{max} is the highest carbon emission value among all feasible solutions of P_1 , and $\xi > 0$ is a positive constant which will be used in the proof, specifically to indicate that there must exist a certain scheduling policy under the problem whose latency can always be less than the upper bound on latency in every time slot, and the cumulative delay constraint difference is ξ .

Proof. Due to space constraints, we give an abbreviated proof, and the detailed proof can be found in Theorem 4.8 of the [21]. Since the original problem P_1 is guaranteed to have a feasible solution, it also possesses an optimal solution and a vector of optimal scheduling strategy denoted as ce_{opt} and α_{opt} , respectively [25]. Therefore, we can state that:

$$\begin{aligned} & E[\Delta(q(t)) + V \cdot SumCE(t)|q(t)] \\ & \leq B + E[q(t) \cdot (T_A(\alpha_{opt}(t)) - T_{avg})] + V \cdot E[SumCE(\alpha_{opt}(t))] \\ & \leq B + V \cdot ce_{opt}. \end{aligned} \quad (26)$$

By summing up the derived conclusion over time slots, we obtain:

$$\begin{aligned}
 (B + V \cdot ce_{opt})T &\geq \sum_{t=1}^T E[\Delta(q(t)) + V \cdot SumCE(t)|q(t)] \\
 &= E[L(q(T))] + V \cdot \sum_{t=1}^T E[SumCE(t)|q(t)] \\
 &\geq V \cdot \sum_{t=1}^T E[SumCE(t)|q(t)].
 \end{aligned} \tag{27}$$

Thus, we can get our conclusion as:

$$\frac{1}{T} \sum_{t=1}^T E[SumCE(t)|q(t)] \leq ce_{opt} + \frac{B}{V}. \tag{28}$$

This establishes the first conclusion of Theorem 2. Next, we will prove the second conclusion. Assuming there is a vector of placement strategy α^* satisfies:

$$\exists \xi > 0, E[T_A(\alpha^*) - T_{avg}] \leq -\xi. \tag{29}$$

Since the carbon emissions objective in the original problem is bounded, with a lower bound of the optimal value ce_{opt} and an upper bound of ce_{max} , we can conclude that:

$$\Delta(q(t)) + V \cdot ce_{opt} \leq B + q(t) \cdot (T_A(\alpha^*) - T_{avg}) + V \cdot ce_{max}. \tag{30}$$

Taking the expectation of both sides of the above equation, we obtain:

$$\begin{aligned}
 E[\Delta(q(t))|q(t)] + V \cdot ce_{opt} &\leq B + E[q(t)]E[\cdot(T_A(\alpha^*) - T_{avg})] + V \cdot ce_{max} \\
 &\leq B - \xi \cdot E[q(t)] + V \cdot ce_{max}.
 \end{aligned} \tag{31}$$

Thus, we can get:

$$E[L(q(t+1)) - L(q(t))] \leq B + V \cdot (ce_{max} - ce_{opt}) - \xi \cdot E[q(t)]. \tag{32}$$

By summing up the derived conclusion over time slots, we obtain:

$$E[L(q(T)) - L(q(1))] \leq T[B + V \cdot (ce_{max} - ce_{opt})] - \xi \sum_{t=0}^T E[q(t)]. \tag{33}$$

Thus, we can get:

$$\frac{1}{T} \sum_{t=1}^T E[q(t)] \leq \frac{B + V \cdot (ce_{max} - ce_{opt})}{\xi}. \tag{34}$$

□

From these two equivalences we can infer that there exists a trade-off between latency and carbon emissions is within $[O(1/V), O(V)]$, which means by changing the value of V we can control the performance of our algorithm. When we choose a larger control parameter V , optimization targeting carbon emissions yields better results, but the latency will also increase.

Theorem 3. *Our Markov approximation based one-slot optimization algorithm has the following performance guarantees:*

$$E[s_{local}(t)] \leq s_{opt}(t) + r \cdot \ln |\Omega| \quad (35)$$

where $s_{opt}(t)$ is the theoretical optimal solution of sub-problem P_2 , and the online algorithm based on Markov approximation process proposed in this paper converges to the solution as $s_{local}(t)$, r is the system parameter regulating the state transfer probability in Algorithm, and Ω is the set of feasible solutions of P_2 .

Proof. the detailed proof can be found in [26].

5 Performance Evaluation

In this section, we demonstrate the effectiveness of our algorithm by comparing it with several benchmarks in a simulation experiment based on real data. Besides, we verify the derived theoretical results mentioned above and discuss the impact of different parameters.

Table 1. Settings of experiment parameters

	parameters	value
Network device	transmission power(kW/Gbps)	1.28
	bandwidth(Mbps)	100
Data Center Server	calculate power rate(kW)	0.4
	main frequency(GHz)	8
	PUE	1.5
Edge Server	main frequency(GHz)	1.5
Carbon emission	thermal power(kgCO2/kW.h)	0.842

A. Simulation Setup

To ensure the reliability of our simulation results, we incorporate the clean energy ratio of data centers in different regions of China, which was provided by Greenpeace [27]. Our deployed data centers are strategically located in representative provinces such as Sichuan, Guizhou, Jiangsu, Guangdong and Shanghai, where

there are large differences in the carbon emission rates of data center power and transmission distances in these regions. Our simulation spans 576 time slots, each lasting 5 min. At the beginning of each time slot, the carbon emission rate for each data center is adjusted within a plausible range, following a normal distribution that is centered on the median. Concurrently, other system parameters remain constant and are detailed in Table 1. During each time slot, the number of newly arrived computing tasks of each edge node is uniformly distributed across different scales, and the task information, including computation and data amounts, is obtained from Intel Netbatch [28].

B. Performance Benchmark

In this section, we abbreviate our proposed algorithm as the Lyapunov Markov(LM) for convenience, as it is based on Lyapunov optimization and Markov approximation. We will compare its performance against the following benchmarks:

- **All-Cloud algorithm(ACloud)**: This algorithm adheres to the conventional cloud computing paradigm by disregarding the processing capabilities of edge servers and uploading all tasks from the edge to the nearest data center to achieve minimal latency.
- **Random Load Balance(RLB)**: This algorithm, while ensuring latency requirements are met, initially selects a subset of tasks for execution at the edge, and then uploads a portion of the remaining tasks to randomly selected distant low-carbon data centers, rather than uploading all to the nearest data center. This approach not only maintains a stable load across various data centers but also effectively reduces carbon emissions.
- **Delay-optimal Markov(DoM)**: This single-time-slot algorithm simply optimizes for latency as the primary objective. By implementing a Markov approximation process algorithm with the same parameters as in the LM algorithm, it achieves a cloud-edge scheduling algorithm with minimal latency within a single time slot, without significantly deviating from the carbon emission constraints.
- **Carbon-optimal Markov(CoM)**: This single-time-slot algorithm simply optimizes for carbon emissions as the primary objective. By implementing a Markov approximation process algorithm with the same parameters as in the LM algorithm, it achieves a cloud-edge scheduling algorithm with minimal carbon emissions within a single time slot, without significantly deviating from the latency constraints.

In summary, ACloud simulates a traditional cloud computing scheduling algorithm that prioritizes latency, while RLB emulates a heuristic geo-distributed cloud-edge optimization strategy. DoM and CoM, which both employ a singular optimization goal within a Markov approximation framework with identical parameters, are specifically designed to underscore the comparative advantages of the Lyapunov optimization approach in achieving a balance between latency and carbon emissions.

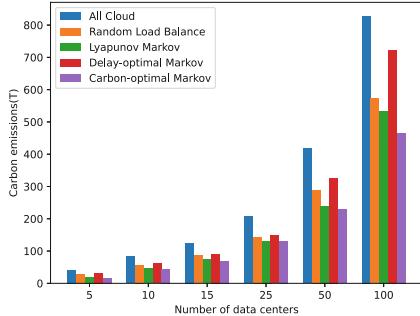


Fig. 2. Carbon emissions of five algorithms under different system configurations.

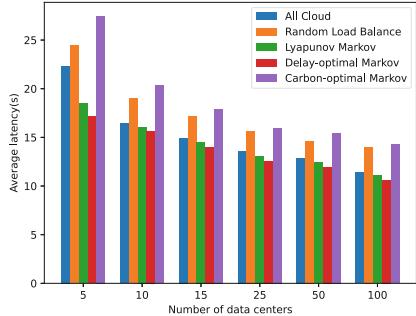


Fig. 3. Average system delay of five algorithms under different system configurations.

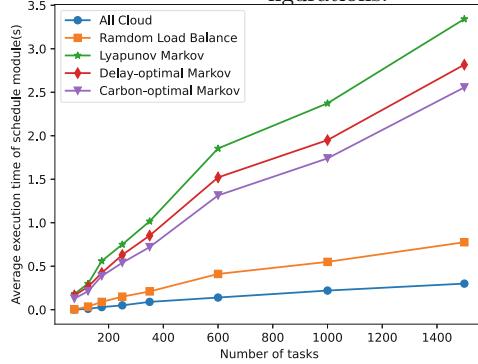


Fig. 4. Average execution time of five algorithms under different loading conditions.

C. Algorithm Comparison

Figure 2 and Fig. 3 depict the carbon emissions and average latency of five algorithms under different system configurations. For each configuration with a fixed number of data centers, we conducted tests across various task scales and integrated the results through a weighted ensemble approach, ultimately deriving a comprehensive assessment of system carbon emissions and latency for different configurations of data center counts.

From the figures, we observe that focusing exclusively on latency with greedy scheduling and failing to utilize the edge capabilities, ACloud results in the worst carbon emissions and also exhibits a lower degree of optimization in terms of latency. RLB reduces carbon emissions through load balancing between edge nodes and geo-distributed data centers, but at the cost of higher latency due to its random scheduling approach. DoM and CoM represent two extremes in the trade-off between carbon emissions and latency. Although DoM achieves the lowest average latency, it incurs significant carbon emissions, deviating from our original design objective. Similarly, CoM minimizes carbon emissions but causes intolerable latency. Our algorithm, on the contrary, achieves a large optimization

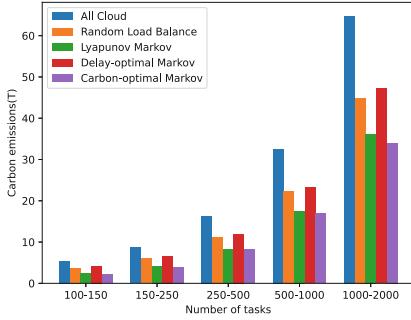


Fig. 5. Carbon emissions of five algorithms under different loading conditions.

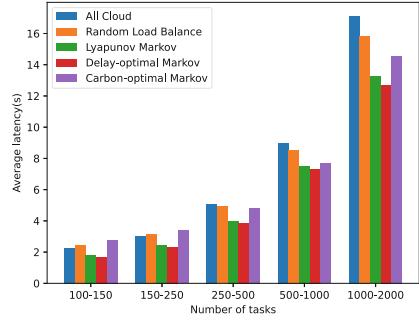


Fig. 6. Average system delay of five algorithms under different loading conditions.

in both the delay and carbon emission dimensions, both of which are close to the optimal value, and it demonstrates stability even in large-scale scenarios.

Figure 4 illustrates the average execution time of five algorithms for each time slot under different loading conditions. Although algorithms based on Markov approximation have relatively slower execution times compared to ACloud and RLB, considering that a time slot is 5 min long, the execution time of few seconds is still acceptable. In practice, system administrator can easily reduce the execution time of the scheduling module by appropriately decreasing the values of the system parameters r and R_{max} , which will be further discussed below.

Figures 5 and 6 showcase the carbon emissions and average latency of five algorithms under varying loading conditions when $M = 5$. As the workload intensifies, the number of tasks that can be efficiently executed on the edge servers tends to reach a stable state. Consequently, both the overall carbon emissions and latency experience an upward trend. Nevertheless, our algorithm maintains its superiority throughout this process, primarily attributed to its advanced cross-regional data center scheduling capability.

D. The Impact of Different Parameters

- 1) **Convergence:** Figure 7 shows the process of converging to the minimum value of the optimization objective with the increase of the number of iterations, where the parameters of the different curves are the system parameter r set in the Algorithm, which affects the probability η of changing the state in the Markov approximation algorithm. The larger r is, the larger the probability is, which indicates that the state of the system is more easily to be changed, and the algorithm will have a larger space for exploring, but it will be equally more difficult to converge to a fixed value, and sometimes even miss the minimum value. However, if r is too small, the state of the system is difficult to be changed, and it may be difficult to find the optimal solution. From Fig. 7, it can be found that the algorithm works best when $r = 0.1$, which indicates that the selection should be moderate, and not too large or too small.

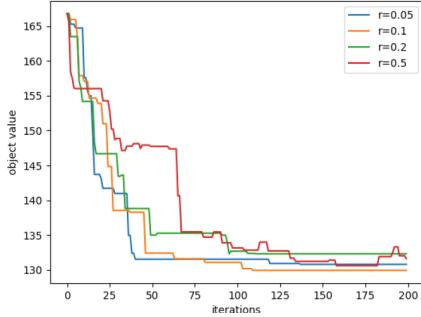


Fig. 7. The impact of different values of r in Algorithm.

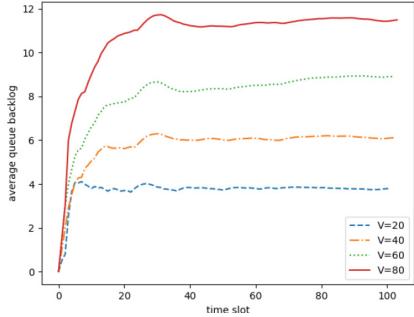


Fig. 8. The impact of different values of V .

- 2) **Weight-parameter:** The parameter V controls the trade-off between carbon emissions and latency in the optimization objective of the sub-problem within a single time slot, and as V increases this trade-off is biased in favour of carbon emissions and leads to larger and larger virtual queue backlog controlling the delay constraints, which can be seen in Fig. 8. We observe that the average queue size gradually stabilises as the time slot represented by the x -axis increases, suggesting that the Lyapunov optimization does indeed provide an effective constraint on the task latency, while the positive correlation between the queue size and the value of V suggests a controlling effect of the parameter V .
- 3) **Iteration-parameter:** In Fig. 9 we can find that the system parameter T_{avg} also directly controls the trade-off between latency and carbon emissions, when T_{avg} is small it represents a tighter constraint on delay and it is difficult to optimize the carbon emissions, while a larger T_{avg} indicates that the constraints on delay are looser and the carbon emissions can be optimized to a greater extent. While R_{max} is the iteration number parameter, it is obvious that the approximate solution obtained by the system will gradually converge to the theoretical optimal solution when the number of iterations is higher, but it will lead to more algorithm execution time. The degree of closeness of the optimal solution and the algorithm execution time are also a pair of trade-offs.

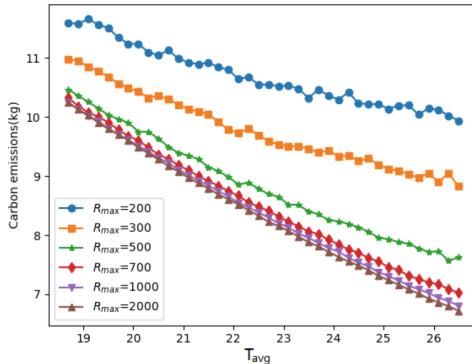


Fig. 9. The impact of different values of T_{avg} and R_{max} .

6 Conclusion

In this paper we study the overall carbon emissions and latency of geographically distributed cloud-edge systems. We propose an online scheduling algorithm based on Lyapunov optimization techniques and Markov approximation to decouple the long-term optimization problem into individual time slots and optimize the trade-off between carbon emissions and latency. We prove the performance guarantees of our proposed algorithm on a theoretical level and compare it with other algorithms through simulation experiments to demonstrate the superiority of our proposed algorithm.

References

1. Xu, J., Ren, S.: Online learning for offloading and autoscaling in renewable-powered mobile edge computing. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2016). <https://doi.org/10.1109/GLOCOM.2016.7842069>
2. Han, T., Ansari, N.: A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources. IEEE/ACM Trans. Networking **24**(2), 1038–1051 (2016). <https://doi.org/10.1109/TNET.2015.2404576>
3. Li, W., et al.: On enabling sustainable edge computing with renewable energy resources. IEEE Commun. Mag. **56**(5), 94–101 (2018). <https://doi.org/10.1109/MCOM.2018.1700888>
4. Akamai: Akamai sustainability report 2021 (2021). <https://www.akamai.com/zh/resources/research-paper/akamai-sustainability-report-2021>
5. Lemay, M., Nguyen, K.K., St. Arnaud, B., Cheriet, M.: Toward a zero-carbon network: converging cloud computing and network virtualization. IEEE Internet Comput. **16**(6), 51–59 (2012). <https://doi.org/10.1109/MIC.2011.128>
6. Xu, H., Feng, C., Li, B.: Temperature aware workload management in geodistributed data centers. IEEE Trans. Parallel Distrib. Syst. **26**(6), 1743–1753 (2015). <https://doi.org/10.1109/TPDS.2014.2325836>

7. Lin, M., Wierman, A., Andrew, L.L.H., Thereska, E.: Dynamic right-sizing for power-proportional data centers. In: 2011 Proceedings IEEE INFOCOM, pp. 1098–1106 (2011). <https://doi.org/10.1109/INFCOM.2011.5934885>
8. Wierman, A., Andrew, L.L.H., Tang, A.: Power-aware speed scaling in processor sharing systems. In: IEEE INFOCOM 2009, pp. 2007–2015 (2009). <https://doi.org/10.1109/INFCOM.2009.5062123>
9. Ibrahim, A., Noshy, M., Ali, H.A., Badawy, M.: Papso: a power-aware VM placement technique based on particle swarm optimization. *IEEE Access* **8**, 81747–81764 (2020). <https://doi.org/10.1109/ACCESS.2020.2990828>
10. Zhou, Z., Liu, F., Zou, R., Liu, J., Xu, H., Jin, H.: Carbon-aware online control of geo-distributed cloud services. *IEEE Trans. Parallel Distrib. Syst.* **27**(9), 2506–2519 (2016). <https://doi.org/10.1109/TPDS.2015.2504978>
11. Chen, L., Zhou, S., Xu, J.: Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. *IEEE/ACM Trans. Networking* **26**(4), 1619–1632 (2018). <https://doi.org/10.1109/TNET.2018.2841758>
12. Gu, L., Zhang, W., Wang, Z., Zeng, D., Jin, H.: Service management and energy scheduling toward low-carbon edge computing. *IEEE Trans. Sustain. Comput.* **8**(1), 109–119 (2023). <https://doi.org/10.1109/TSUSC.2022.3210564>
13. Ma, H., Zhou, Z., Zhang, X., Chen, X.: Toward carbon-neutral edge computing: greening edge AI by harnessing spot and future carbon markets. *IEEE Internet Things J.* **10**(18), 16637–16649 (2023). <https://doi.org/10.1109/JIOT.2023.3268339>
14. Liu, Z., Lin, M., Wierman, A., Low, S., Andrew, L.L.H.: Greening geographical load balancing. *IEEE/ACM Trans. Networking* **23**(2), 657–671 (2015). <https://doi.org/10.1109/TNET.2014.2308295>
15. Lin, W.T., Chen, G., Li, H.: Carbon-aware load balance control of data centers with renewable generations. *IEEE Trans. Cloud Comput.* **11**(2), 1111–1121 (2023). <https://doi.org/10.1109/TCC.2022.3150391>
16. Khosravi, A., Andrew, L.L.H., Buyya, R.: Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers. *IEEE Trans. Sustain. Comput.* **2**(2), 183–196 (2017). <https://doi.org/10.1109/TSUSC.2017.2709980>
17. Kumar, N., Aujla, G.S., Garg, S., Kaur, K., Ranjan, R., Garg, S.K.: Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers. *IEEE Trans. Industr. Inf.* **15**(5), 2947–2957 (2019). <https://doi.org/10.1109/TII.2018.2800693>
18. Kwak, J., Kim, Y., Lee, J., Chong, S.: Dream: dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE J. Sel. Areas Commun.* **33**(12), 2510–2523 (2015). <https://doi.org/10.1109/JSAC.2015.2478718>
19. Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Sel. Areas Commun.* **34**(12), 3590–3605 (2016). <https://doi.org/10.1109/JSAC.2016.2611964>
20. Zhang, G., Zhang, W., Cao, Y., Li, D., Wang, L.: Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices. *IEEE Trans. Industr. Inf.* **14**(10), 4642–4655 (2018). <https://doi.org/10.1109/TII.2018.2843365>
21. Neely, M.: Stochastic network optimization with application to communication and queueing systems. Springer (2010)
22. Xu, J., Chen, L., Zhou, P.: Joint service caching and task offloading for mobile edge computing in dense networks. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp. 1–9 (2018). <https://doi.org/10.1109/INFOCOM.2018.8480510>

- ence on Computer Communications, pp. 207–215 (2018). <https://doi.org/10.1109/INFOCOM.2018.8485977>
- 23. Liu, F., Shu, P., Lui, J.C.: Appatp: an energy conserving adaptive mobile-cloud transmission protocol. *IEEE Trans. Comput.* **64**(11), 3051–3063 (2015). <https://doi.org/10.1109/TC.2015.2401032>
 - 24. Chen, M., Liew, S.C., Shao, Z., Kai, C.: Markov approximation for combinatorial network optimization. *IEEE Trans. Inf. Theory* **59**(10), 6301–6327 (2013). <https://doi.org/10.1109/TIT.2013.2268923>
 - 25. Ouyang, T., Zhou, Z., Chen, X.: Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing. *IEEE J. Sel. Areas Commun.* **36**(10), 2333–2345 (2018). <https://doi.org/10.1109/JSAC.2018.2869954>
 - 26. Cao, T., Qian, Z., Wu, K., Zhou, M., Jin, Y.: Service placement and bandwidth allocation for MEC-enabled mobile cloud gaming. In: 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 179–188 (2021). <https://doi.org/10.1109/WoWMoM51794.2021.00031>
 - 27. Greenpeace: Lighting up the green cloud: A study of data center energy consumption and renewable energy potential in China (2019). <https://www.greenpeace.org.cn/wp-content/uploads/2019/09/>
 - 28. Shai, O., Shmueli, E., Feitelson, D.G.: Heuristics for resource matching in intel’s compute farm. In: Desai, N., Cirne, W. (eds.) *Job Scheduling Strategies for Parallel Processing*, pp. 116–135. Springer, Heidelberg (2014)



Integrating Blockchain, Smart Contracts, NFTs, and IPFS for Enhanced Transparency and Ethical Sourcing in Coffee and Cocoa Supply Chains

V. H. Khanh, N. M. Triet, L. K. Bang^(✉), P. D. Trinh, N. N. Hung, N. H. Bang, P. T. Nghiem, and T. D. Khoa

FPT University, Can Tho city, Vietnam
KhanhVH@fe.edu.vn, banglkce160155@fpt.edu.vn

Abstract. This paper explores the integration of blockchain technology, smart contracts, Non-Fungible Tokens (NFTs), and the InterPlanetary File System (IPFS) to enhance transparency, efficiency, and ethical sourcing in the coffee and cocoa supply chains. By developing a comprehensive supply chain management system tailored to these sectors, we aim to address the challenges of traceability and data integrity from the point of origin to the end consumer. Our approach evaluates the system's implementation across four Ethereum Virtual Machine (EVM)-compatible platforms—Binance Smart Chain, Polygon, Fantom, and Celo—to assess operational efficacy and economic viability. Through this exploration, we demonstrate how the synergistic use of blockchain, smart contracts, NFTs, and IPFS can revolutionize supply chain management practices, ensuring data integrity, enhancing transparency, and fostering a more equitable system for all stakeholders involved.

Keywords: Supply Chain Transparency · Fair Trade · Blockchain Technology · Smart Contracts · NFTs · IPFS · Distributed Ledger

1 Introduction

The coffee and cocoa industries face significant challenges in achieving transparency, ethical sourcing, and sustainability in their global supply chains. Traditional methods often lack the ability to provide full traceability and transparency, which are crucial for maintaining ethical standards and consumer confidence. The conventional supply chain model, from cultivation to consumption, presents various hurdles at each stage, highlighting the need for a system that ensures integrity and transparency throughout [9, 10]. The advent of blockchain technology, smart contracts, Non-Fungible Tokens (NFTs), and the InterPlanetary File System (IPFS) offers promising solutions to these challenges. Our strategy utilizes these innovations to develop an integrated supply chain management system [15, 16] tailored for the coffee and cocoa sectors, aiming to improve transparency and data integrity from origin to end-user.

The integration of blockchain technology into sustainable supply chain management has been a focal point of recent research, offering solutions to enhance transparency and trust within the sector [11, 25]. Saberi et al. and Yoo et al. have demonstrated how blockchain can facilitate sustainable practices and transparent pricing, essential for fair trade and building trust across the supply chain [23, 28]. Dietrich et al. and Shahid et al. further this narrative by showcasing smart contracts' role in risk mitigation and traceability, particularly in the agri-food sector [7, 24]. The concept of 'virtual operations', as introduced by Dolgui et al., and the application of blockchain and IoT for real-time monitoring by Hasan et al., represent significant strides towards a more proactive and transparent approach to supply chain management [8, 12]. These studies collectively underscore the potential of blockchain and smart contracts to revolutionize supply chain operations, advocating for their broader adoption to improve sustainability and efficiency.

Our methodology addresses the unique requirements of the coffee and cocoa sectors by incorporating smart contracts, NFTs, and IPFS into a unified supply chain management system. This setup not only traces the product journey from farm to consumer but also safeguards the data's integrity and transparency throughout [4]. We tailor our solution to meet the specific challenges of these sectors, emphasizing traceability and ethical sourcing. Utilizing IPFS alongside blockchain, we secure the permanence and accuracy of supply chain data, tackling issues of scalability and integrity as noted by Hung et al. [14]. The system is underpinned by distributed ledger technology, creating a reliable source for all transactions and activities within the supply chain. This facilitates real-time, uniform, and precise information access for all network participants, leading to a supply chain that is transparent, adaptable, and equitable for everyone involved.

Our system's adaptability and efficiency were tested across four EVM-compatible platforms: Binance Smart Chain, Polygon, Fantom, and Celo. This analysis highlights the system's flexibility and its ability to utilize each platform's unique capabilities for supply chain optimization. We evaluated key operations like data entry, NFT generation, and NFT transferability, providing insights into the system's performance in terms of speed, efficiency, and reliability. Additionally, a comparison of transaction fees across these platforms reveals our system's economic benefits, suggesting potential for cost savings and improved operational efficiency.

In summary, our paper presents a novel and comprehensive model that leverages blockchain technology, smart contracts, NFTs, and IPFS to revolutionize supply chain management in the coffee and cocoa sectors. This model not only addresses the challenges of transparency, efficiency, and ethical sourcing but also sets a new standard for supply chain management practices. Through detailed implementation, rigorous evaluation, and a focus on economic viability, we demonstrate the transformative potential of these technologies in creating a more equitable, transparent, and trustable supply chain ecosystem.

2 Related Work

2.1 Sustainable Supply Chain and Transparency

The domain of sustainable supply chain management has benefited from research underscoring the role of blockchain technology in enhancing transparency. Saberi et al. investigate blockchain's potential to improve trust and verify sustainable practices within supply chains [23]. Complementing this, Yoo et al. provide insights into transparent price tracing mechanisms, crucial for fair trade verification in supply chain networks [28].

Dietrich et al. offer a perspective on using smart contracts to diminish risks in supply chains, suggesting that blockchain's reliability can contribute to the sustainability of these systems [7]. Similarly, Shahid et al. discuss a comprehensive blockchain solution for the agri-food sector, ensuring traceability from production to consumption [24].

Alvarado et al. recognize blockchain as a transformative tool for supply chain transparency, with implications for improved sustainability [2], while Bai et al. propose a framework for evaluating blockchain's effectiveness in fostering transparent and sustainable supply chains [3].

2.2 Blockchain and Smart Contracts for Supply Chain Management

In the field of supply chain management, blockchain technology and smart contracts are being rigorously explored to address longstanding logistical challenges. The work by Dolgui et al. offers an innovative perspective by incorporating the concept of 'virtual operations' into smart contracts, creating a bridge between physical logistics processes and their digital execution [8]. This approach presents a dynamic model aligning with the complex scheduling demands of contemporary supply chains.

Enhancing the tracking and management of shipments, Hasan et al. propose a blockchain solution that leverages smart contracts and IoT technology for real-time monitoring [12]. This integration exemplifies how blockchain can facilitate immediate and automated responses to changes in shipment conditions, marking a shift towards more proactive supply chain management.

The study by Agrawal et al. extends the application of blockchain for collaborative resource sharing in business networks, utilizing smart contracts to validate quality and data authenticity [1]. Similarly, Li et al. tackle the challenges of information asymmetry and collaboration efficiency by designing a system that standardizes information exchange through smart contracts, thus enhancing the security and reliability of supply chain data [17].

On a more interactive front, Putri et al. employ blockchain smart contracts within a serious game, simulating agricultural supply chain transactions and offering a novel educational tool for understanding blockchain applications in this sector [21]. Meanwhile, Chang advocates for blockchain's transformative potential in supply chain re-engineering, with smart contracts at the heart of automating and securing transactions [6].

2.3 Advancements in Blockchain for Agricultural and Food Supply Chain Management

Blockchain technology is increasingly recognized for its transformative impact on the agricultural and food supply chains, enhancing transparency and security. V.M. et al. developed a blockchain system utilizing smart contracts to track and trace crops without intermediaries, demonstrating blockchain's capability to ensure the security and integrity of agricultural operations [18]. Similarly, 'FarmersChain' by Reddy et al. leverages blockchain and IoT to improve data integrity and traceability, addressing key industry challenges like fair pricing and market access for farmers [22]. These studies highlight blockchain's role in creating more transparent and equitable supply chains in agriculture.

Turning to the food industry, Hawashin et al. discuss the utilization of composable NFTs for managing and trading expensive packaged food products [13]. Their proposed solution employs blockchain and NFTs to verify the traceability and authenticity of food products, thereby reinforcing trust among stakeholders and consumers. By integrating IPFS for data storage, they address the challenge of storing large data files while maintaining their permanency and immutability. Similarly, Pawar et al. propose an Ethereum and IPFS-based decentralized supply chain management system that addresses scalability and data integrity issues [20]. Their framework automates procedures and data exchanges, demonstrating the potential of blockchain to enhance the efficiency of supply chain management.

Further contributions include Battah et al.'s exploration of blockchain and NFTs for managing the ownership and trading of AI models, drawing parallels with the need for transparent provenance in supply chains [5]. Tahmasbzadeh et al. present a blockchain-based data storage model for the drug supply chain, aiming to ensure efficient information queries and reduce the load pressure on the chain [26]. Lastly, Majdalawieh et al. propose a blockchain and IoT-based solution to improve the safety and quality of food products in the processed poultry food supply chain [19], while Waikar emphasizes blockchain's role in enhancing trust and transparency in supply chain management [27].

3 Approach

3.1 The Classical Architecture of Transparency and Ethical Sourcing in Coffee and Cocoa Supply Chain Management

The depicted flowchart delineates the conventional stages in the supply chain for coffee and cocoa products, from the cultivation by coffee farmers to the ultimate consumption. At the outset of this supply chain are the farmers, who are entrusted with the cultivation of coffee plants, the harvesting of beans, and their preliminary processing. This foundation of the supply chain is not just about agricultural production but also about the assurance of quality and the application of farming practices that are expected to meet the standards of fair trade.

The harvested coffee beans are then transported to processors and manufacturers. This transition marks the stage where the raw coffee undergoes a transformation into a finished product ready for consumption or into a semi-finished product awaiting further refinement. The role of the processing and manufacturing entities is pivotal; they are responsible for not only preserving the quality of the coffee beans but also for the adherence to ethical processing standards. The outcome of this stage greatly influences the final product's quality, and it is here that the value addition process is most tangible. Further down the supply chain are the distributors and retailers who assume the responsibility of moving the finished products closer to the end consumers. Their role encompasses the intricate tasks of logistics, including transportation, warehousing, and ensuring the products' presence in the market. This segment of the supply chain is characterized by a complex network of entities working cohesively to maintain the product's condition and uphold the fair trade certifications the products might carry (Fig. 1).

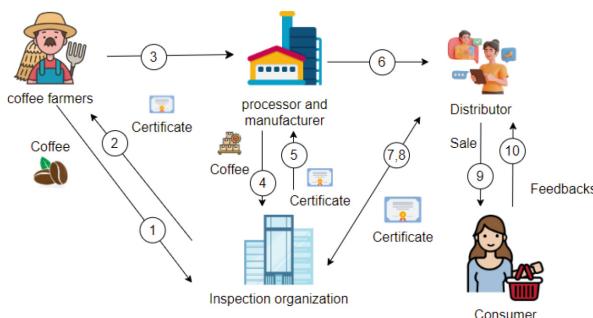


Fig. 1. Traditional Flowchart of the Conventional Supply Chain Mechanism for Coffee and Cocoa: Pathways from Farm to Consumer

Independent certification organizations are pivotal actors in the traditional supply chain of coffee and cocoa, tasked with the verification of compliance with ethical, quality, and sustainability standards. Acting as neutral arbiters, they audit processes from cultivation to retail, ensuring that each step adheres to fair trade principles. By issuing certificates to entities that meet these standards, they not only bolster fair labor and environmental practices but also enhance consumer trust. Their certification signifies that a product has been ethically sourced, with its journey through the supply chain transparent and traceable. This role is crucial in maintaining the integrity of fair trade claims and in reinforcing the consumers' confidence in the products they choose to purchase.

At the terminal point of this model is the consumer, whose choices and preferences exert a significant influence over the entire supply chain. Consumers' decisions to purchase fair trade products act as a catalyst for maintaining ethical standards in the supply chain. Additionally, the feedback from consumers post-

purchase serves as a gauge for the effectiveness of the supply chain practices and propels continuous improvement.

This traditional supply chain model, while systematic, does face challenges in ensuring complete transparency and traceability. Ensuring that ethical practices are consistently applied at each stage and that the benefits of fair trade are equitably distributed throughout the chain, especially to the initial producers - the farmers, remains a persistent concern. This model has served as a foundational structure for the industry and continues to facilitate the journey of coffee and cocoa from their origins to the global consumer market.

3.2 Implementing Blockchain, Smart Contracts, NFTs, and IPFS for Supply Chain Integrity in the Coffee and Cocoa Sectors

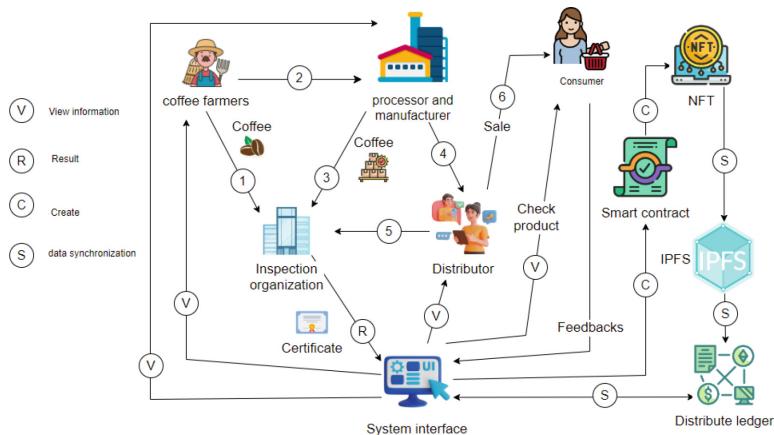


Fig. 2. Integration of Blockchain, Smart Contracts, NFTs, and IPFS for Enhanced Transparency in Coffee and Cocoa Supply Chains

The model depicted outlines a supply chain management system for coffee and cocoa that leverages blockchain technology, smart contracts, non-fungible tokens (NFTs), and the InterPlanetary File System (IPFS) to foster transparency. This framework is designed to provide a comprehensive view of the product journey, from the coffee farmers to the end consumer, while ensuring data integrity and transparency. At the commencement of this chain, coffee farmers record the production data on the blockchain, creating an immutable ledger of the coffee's origin, quality, and quantity. This information, once inscribed onto the blockchain, is accessible for viewing by subsequent participants in the chain, ensuring a transparent inception of the product's journey (Fig. 2).

As the coffee beans progress to processors and manufacturers, smart contracts automatically execute agreements related to the processing based on the quality and quantity of the coffee received, documented in the blockchain. In this supply

chain, they are employed to facilitate and verify transactions without third-party intermediaries. For example, when coffee beans are delivered to processors and manufacturers, smart contracts are triggered, ensuring that payments are released only when the agreed-upon conditions concerning the coffee's quality and quantity are met. This system reduces the margin for human error and enhances the efficiency of the trade process. This automation ensures that the terms agreed upon at the onset are fulfilled without manual intervention, thereby minimizing errors and maintaining consistent standards.

The inspection organization occupies a central role within the supply chain, acting as the authoritative entity that ensures compliance with the industry's standards for quality and ethical production. It scrutinizes the coffee at various production stages, from the farming methods employed to the processing practices. Once the inspection is complete, the organization records its findings on the blockchain. This not only includes the outcomes of the inspections but also, when necessary, issues digital certificates which are tokenized as NFTs. These NFTs serve as immutable proof of the coffee's quality and ethical pedigree, offering an additional layer of trust. These NFTs are immutable once recorded on the blockchain, providing a digital signature that cannot be altered or duplicated, thus preventing fraud. As the coffee moves through the supply chain, these tokenized certificates provide stakeholders with the ability to verify, at any point, that the products have been produced according to the highest standards, without the possibility of tampering or falsification of the compliance records.

Distributors check the product against the blockchain records, ensuring that the coffee they receive aligns with the information logged at the source. Retail sales are then conducted with the confidence that the product's history is transparent and verifiable by all parties. In this model, consumers can scan a product to view its full history, from farm to store, including the NFT certificates, thus receiving assurance of the product's ethical sourcing.

At the consumer end, the final purchaser can review the product's history and verify its certifications through the blockchain records. The NFTs provide an additional layer of security and authenticity, ensuring that the certificates cannot be forged or misrepresented. Feedback from consumers and other supply chain participants can be recorded on the blockchain via IPFS, providing a decentralized storage solution that ensures the permanence and security of the data. This feedback becomes part of the product's digital footprint, contributing to a dynamic and responsive supply chain where information flows continuously and transparently.

In the system, IPFS plays a vital role in storing and accessing the data across a distributed network, ensuring that information such as product origins, transaction history, and consumer feedback is permanently recorded and tamper-proof. This decentralized storage method not only secures the data but also facilitates the efficient dissemination of information to all stakeholders, who can access the comprehensive history of the products. Additionally, the distributed ledger serves as the backbone of the system, providing a single source of truth for all transactions and interactions within the supply chain. It ensures

data synchronization across various nodes, meaning that every participant in the network has access to real-time, consistent, and accurate information. This transparency is crucial for maintaining the credibility of the supply chain and for supporting the ethical claims of the products.

4 Implementation

In the ensuing section, we detail the operational mechanisms underpinning the issuance and verification of certificates within a blockchain-enabled supply chain, specifically tailored for the coffee and cocoa industries. This segment delineates the integrated use of smart contracts, non-fungible tokens (NFTs), and the Inter-Planetary File System (IPFS) to establish and maintain a transparent record of product quality and ethical compliance from origin to end consumer. We will unpack the system's interfaces and processes, which allow for the creation and renewal of certificates, and how these are inextricably linked to the product and its journey through the supply chain. The focus herein is to articulate the functionality of each component within the system, demonstrating how they collectively contribute to an immutable and synchronized ledger that undergirds the integrity of the supply chain network.

4.1 Transaction/Data Creation

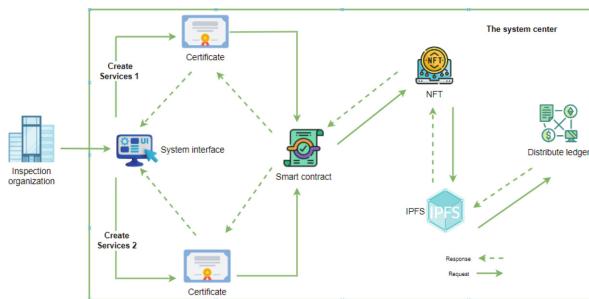


Fig. 3. Creation Framework for Enhanced Certificate Management in the Blockchain-Enabled Supply Chain for Coffee and Cocoa

The diagram outlines a blockchain-based framework for enhancing the supply chain of coffee and cocoa, focusing on product tracking, verification, and recording from origin to consumer (Fig. 3). Central to this setup is the system center, where smart contracts facilitate key supply chain activities, automating transactions based on specific criteria like quality checks. These contracts play a crucial role in ensuring compliance and consistency across the supply chain, minimizing errors and standardizing trade practices.

Inspection entities within this system are tasked with certifying products and processes, their findings and certificates uploaded to the blockchain for transparency and integrity. These digital certificates, often tokenized as NFTs, serve a dual purpose: they confirm product quality and ethical sourcing, and they enable stakeholders to authenticate product information through the supply chain.

NFTs in this architecture are essential for maintaining a verifiable record of product compliance and quality, securely anchored to a distributed ledger that records every transaction and movement. This ensures an unalterable trail from production to purchase, enhancing trust and transparency. Moreover, the InterPlanetary File System (IPFS) is utilized for decentralized data storage, safeguarding information like product origins and transaction histories against tampering. This approach ensures data security and accessibility, allowing stakeholders easy verification of product provenance and integrity, marking a significant step forward in supply chain management for the coffee and cocoa industries.

4.2 Data Update

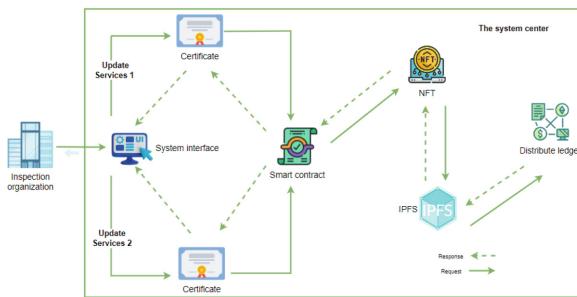


Fig. 4. Updated Framework for Enhanced Certificate Management in the Blockchain-Enabled Supply Chain for Coffee and Cocoa

The updated architecture for the coffee and cocoa supply chain capitalizes on blockchain technology to refine certificate management and adapt to the evolving demands of the industry (Fig. 4). Central to this system is the system center, which serves as a digital hub for orchestrating interactions across the supply chain. Here, smart contracts manage the issuance and renewal of certificates, directly responding to the dynamic nature of agricultural production and trade. This enables the system to swiftly adapt to changes, ensuring that certificate data on the blockchain remains current and reflective of the latest product statuses.

Inspection organizations play a crucial role in this architecture, validating each production stage and updating the blockchain ledger through the system interface. This process not only secures a real-time reflection of certification statuses but also introduces a level of flexibility previously unattainable in traditional supply chain models. The updated services facilitate the issuance of new

certificates and the modification of existing ones, accommodating the inherent variability in agricultural products and ethical sourcing standards.

Non-Fungible Tokens (NFTs) are integral to the system, serving as digital certificates that verify a product's history and compliance with quality standards. By anchoring these NFTs to a distributed ledger, the architecture guarantees the immutability of the supply chain's data, ensuring that each product's journey from farm to consumer is transparent and verifiable. This aspect is crucial for building trust among stakeholders and providing consumers with clear, unalterable records of product provenance and quality.

Finally, the InterPlanetary File System (IPFS) is employed to store data securely and decentralize storage solutions, ensuring that all supply chain information, from transaction histories to feedback, is immutable and permanently accessible. This commitment to data integrity and security underscores the system's capacity to foster trust and ensure the reliability of the supply chain's digital records, making it a robust framework for modern agricultural trade.

4.3 Data Query Process

The architecture we've developed for the coffee and cocoa supply chain employs blockchain technology to offer comprehensive query services, enhancing data and certification status accessibility across the network (Fig. 5). At the heart of this system is the system center where smart contracts are deployed to automate and manage transactions based on predefined supply chain operations triggers. These smart contracts ensure the accuracy and uniformity of business practices by responding to specific events, such as the receipt of goods or quality control checks. Stakeholders utilize query services to access and verify information related to product origins, quality, and compliance with ethical standards. This system allows for inquiries to be converted into blockchain transactions, enabling the retrieval of data from the distributed ledger efficiently. Inspection organizations play a crucial role by using these services to register and update blockchain certificates, which are tokenized as NFTs to signify quality and trust.

Furthermore, the integration of the InterPlanetary File System (IPFS) bolsters the system's commitment to data security and immutability. By storing query results and transaction data on IPFS, the architecture safeguards against unauthorized changes, ensuring that all supply chain information remains accurate and tamper-proof. This framework not only enhances the transparency of the supply chain but also strengthens the verification processes, making it easier for all participants to trust the integrity of the data shared within the network.

5 Evaluation

In the realm of blockchain technology, several platforms have emerged, each offering compatibility with the Ethereum Virtual Machine (EVM). For a comprehensive evaluation of our system, we intend to implement our smart contracts on four notable EVM-compatible platforms: Binance Smart Chain (BSC), Polygon,

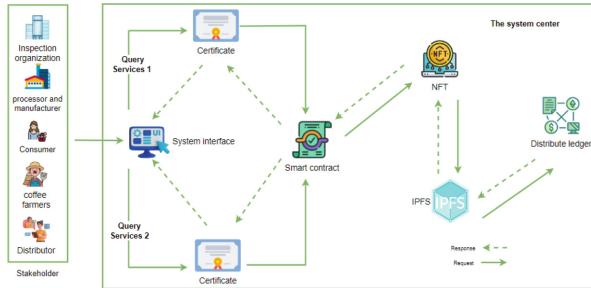


Fig. 5. Query Framework for Enhanced Certificate Management in the Blockchain-Enabled Supply Chain for Coffee and Cocoa

Fantom, and Celo. These platforms are selected based on their unique features and performance attributes, which we will examine to determine their respective efficacies in managing supply chain processes. Concurrently, acknowledging the shift towards decentralized data storage solutions and the increasing utilization of NFTs, we plan to incorporate the storage of certification and supply chain data on the IPFS. For this purpose, the Pinata platform will serve as the interface to this distributed storage network, facilitating the management of supply chain information in a decentralized manner.

5.1 Environment Simulation

The environment outlined herein is a model of a blockchain network tailored for evaluating the integration of NFTs and IPFS within a supply chain context. This network is constituted by nodes, each secured with a distinct public-private key pair, a standard prerequisite for maintaining the integrity of blockchain operations. These nodes represent the various participants in the supply chain, such as inspection organizations, manufacturers, and distributors, with their public keys acting as transactional addresses and their private keys providing necessary authentication (Fig. 6).

Within this framework, nodes are tasked with roles pertinent to supply chain management, including the issuance of NFTs and interaction with smart contracts. The test network is provisioned with an ample supply of ether, allowing for unhindered operation and the elimination of constraints typically imposed by finite resources. The objective of this simulated environment is to closely examine the practicalities of NFT utilization and IPFS integration for the management and verification of supply chain documentation. By emulating a real-world supply chain scenario, the test provides valuable data on the system's performance in a controlled setting, circumventing the potential risks of direct implementation on a live network.

```

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cfffb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbcd5efcae784d7bf4f2ff80

Account #1: 0x70997970C51812d3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdD6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79b6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bcC2695C58ba16FB37d819B0A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976EA74026E726554dB657fA54763abd0C3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfc3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dC79964da2C08b23698B3D3cc7Ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f22181b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81E3f5cdF7f54C3d65f7FBc0aBf5B21E8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22feeaa921c0cf5d620ea67b97

Account #9: 0xa0Ee7A142d267C1f36714E4a8F75612F20a79720 (10000 ETH)
Private Key: 0xa2871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

```

Fig. 6. Configuration of Nodes within a Simulated Blockchain Environment for NFT and IPFS Integration

5.2 Implementing IPFS for Supply Chain Management Systems

In the process of generating a Non-Fungible Token (NFT) for the purpose of supply chain transparency, the initial step involves establishing a structured data representation of the item to be tracked. The Fig. 8 demonstrates the creation of a JSON object, which encapsulates the relevant details of a supply chain item, such as type, origin, processing, manufacturing, and certification information. This JSON object serves as the foundational metadata for the NFT, providing a comprehensive overview of the item's journey from farm to consumer. Following the data structuring, the next step is to test and confirm the functionality of the associated smart contract. The Fig. 9 illustrates a successful test result, indicating that the smart contract's deployment and operations, such as setting conditions and managing time-locked actions, have been verified. This confirmation is crucial as it ensures that the smart contract will perform as expected, automating and enforcing the supply chain processes encoded within.

Once the smart contract is validated, the third step in the sequence is to upload the NFT metadata to a decentralized storage solution, ensuring its permanence and accessibility. The Fig. 10 showcases the interface of Pinata, a service that facilitates the addition of files to the InterPlanetary File System (IPFS). Here, the metadata JSON file is uploaded, ensuring that the supply chain item's digital representation is securely stored and can be retrieved using a unique IPFS hash.

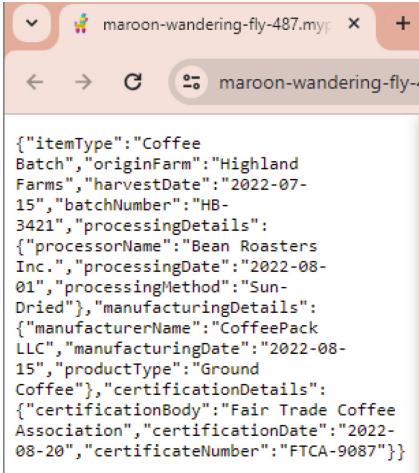


Fig. 7. Generated IPFS Hash for Stored NFT Metadata

```

const body = {
  itemType: "Coffee Batch",
  originFarm: "Highland Farms",
  harvestDate: "2022-07-15",
  batchNumber: "HB-3421",
  processingDetails: {
    processorName: "Bean Roasters Inc.",
    processingDate: "2022-08-01",
    processingMethod: "Sun-Dried"
  },
  manufacturingDetails: {
    manufacturerName: "CoffeePack LLC",
    manufacturingDate: "2022-08-15",
    productType: "Ground Coffee"
  },
  certificationDetails: {
    certificationBody: "Fair Trade Coffee Association",
    certificationDate: "2022-08-20",
    certificateNumber: "FTCA-9087"
  }
};

const options = {
  pinataMetadata: {
    name: "metadata.json",
  }
},

```

Fig. 8. Defining the Data Structure for Supply Chain Item NFT

The Fig. 7 in the sequence represents the generated IPFS hash corresponding to the uploaded metadata. This hash acts as a unique identifier for the NFT's metadata stored on IPFS, ensuring that anyone with the hash can access the metadata in an unaltered state. This step is critical as it links the NFT to a secure, immutable record of the supply chain item's details, completing the NFT generation process.

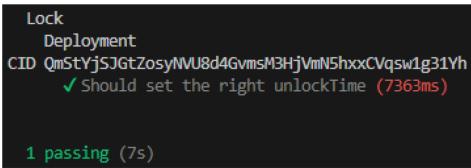


Fig. 9. Test Confirmation of Smart Contract Functionality



Fig. 10. Pinata Cloud Interface for IPFS Metadata Storage

5.3 Testing on EVM-Supported Platforms

Our study leverages the Ethereum Virtual Machine (EVM) as the foundation for deploying smart contracts across several EVM-compatible platforms: Binance Smart Chain, Polygon, Fantom, and Celo. These platforms were chosen to assess

their effectiveness in a transparent supply chain management system. We focused on key operations: data entry for recording supply chain information, the creation of Non-Fungible Tokens (NFTs) to ensure product traceability, and the transfer of NFTs to facilitate secure information flow within the supply chain. Our evaluation criteria included transaction speed, resource efficiency, and user interface accessibility, essential for a system regularly used by supply chain professionals. This analysis aims to demonstrate each platform's capacity to support a transparent and efficient supply chain through secure and reliable NFT generation and transfer.

The table provided offers a comparative view of transaction fees across different blockchain platforms within the context of supply chain management. These fees are a vital aspect of the blockchain's operational costs and are instrumental in the overall assessment of the system's economic viability. Furthermore, we provide a snapshot of the token valuations for the platforms under scrutiny, as observed on January 27, 2024, at 7:00 AM UTC. This summary is aimed at offering a concise overview of the economic environment pertaining to these blockchain networks.

For the Binance Smart Chain (BSC), the transaction fee for creating a blockchain entry—a process analogous to logging a new event or item within the supply chain—is listed as 0.0273134 BNB, approximately valued at \$8.27. This fee reflects the cost of recording transaction data, such as the receipt of a new batch of coffee beans at a processing facility. The creation of an NFT, representing the digital counterpart of a physical supply chain item or certificate, incurs a fee of 0.00109162 BNB or \$0.33 on BSC. This fee is for minting a unique token that encapsulates the attributes of the product, such as origin and certification status. The transfer of an NFT on BSC, which may correspond to the change of custody or ownership within the supply chain, has a fee of 0.00057003 BNB, equating to \$0.17, highlighting the cost-efficiency of asset transfers on this platform.

On the Fantom network, the fee structure is notably low, with transaction creation, NFT creation, and NFT transfer costs enumerated as 0.00957754 FTM, 0.000405167 FTM, and 0.0002380105 FTM, respectively, all translating to negligible dollar amounts. This implies a cost-effective platform for managing transactions within the supply chain, such as updating the status of a shipment or transferring documentation between parties (Table 1).

Table 1. Transaction fee

	Transaction Creation	Create NFT	Transfer NFT
BNB Smart Chain	0.0273134 BNB (\$8.27)	0.00109162 BNB (\$0.33)	0.00057003 BNB (\$0.17)
Fantom	0.00957754 FTM (\$0.00)	0.000405167 FTM (\$0.00)	0.0002380105 FTM (\$0.00)
Polygon	0.006840710032835408 MATIC (\$0.01)	0.000289405001852192 MATIC (\$0.00)	0.000170007501088048 MATIC (\$0.00)
Celo	0.007097844 CELO (\$0.005)	0.0002840812 CELO (\$0.000)	0.0001554878 CELO (\$0.000)

Polygon showcases similar affordability with transaction fees for entry creation, NFT minting, and NFT transfers listed as 0.006840710032835408 MATIC, 0.000289405001852192 MATIC, and 0.000170007501088048 MATIC, respectively. These minimal costs, amounting to just a few cents, suggest a platform conducive to frequent and numerous supply chain transactions without imposing significant financial burdens.

Lastly, the Celo platform maintains a transaction fee structure that supports low-cost operations, with the creation of new entries costing 0.007097844 CELO, and the minting and transferring of NFTs costing 0.0002840812 CELO and 0.0001554878 CELO, respectively, all amounting to less than a cent. This demonstrates the platform's potential for handling high-volume, low-cost transactions typical in supply chain activities.

In conclusion, the table offers an illustrative comparison of the transaction costs associated with supply chain operations on four EVM-compatible blockchain platforms. These costs are crucial for businesses to consider when selecting a blockchain platform for supply chain management, as they directly impact the overall efficiency and cost-effectiveness of the system. The data suggests a favorable economic scenario for implementing blockchain technology within the supply chain, with the potential for significant cost savings compared to traditional systems.

6 Conclusion

The exploration and implementation of blockchain technology, smart contracts, NFTs, and IPFS within the coffee and cocoa supply chains have underscored their potential to significantly improve transparency, traceability, and ethical sourcing practices. Our system's architecture, emphasizing the integration of these technologies, provides a robust framework for managing supply chain data with unparalleled integrity and accessibility. The evaluation across multiple EVM-compatible platforms has revealed the system's adaptability and efficiency, showcasing its potential for broader application within the supply chain domain. The comparative analysis of transaction fees and the operational efficacy of our system highlight its economic viability and potential to offer substantial cost savings over traditional supply chain management practices. Ultimately, our work contributes to the ongoing discourse on the application of blockchain technology in supply chain management, offering a practical and scalable model that enhances transparency, efficiency, and trust across the supply chain network. By leveraging the unique capabilities of blockchain, smart contracts, NFTs, and IPFS, we pave the way for a new era in supply chain management that prioritizes ethical sourcing, data integrity, and stakeholder equity.

References

1. Agrawal, T.K., et al.: Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration. *Int. J. Prod. Res.* **61**, 1497–1516 (2022)
2. Alvarado, J., et al.: New era in the supply chain management with blockchain. In: Research Anthology on Blockchain Technology in Business, Healthcare, Education, and Government (2021)
3. Bai, C., et al.: A supply chain transparency and sustainability technology appraisal model for blockchain technology. *Int. J. Prod. Res.* **58**, 2142–2162 (2019)
4. Bang, N., et al.: Blockchain-enhanced ioht: a patient-centric internet of healthcare things platform with smart contract-driven data management. In: International Conference on Advances in Mobile Computing and Multimedia Intelligence, pp. 50–56. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-48348-6_4
5. Battah, A., et al.: Blockchain and nfts for trusted ownership, trading, and access of ai models. *IEEE Access* **10**, 112230–112249 (2022)
6. Chang, S.E., et al.: Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process. *Technol. Forecast. Social Change* (2019)
7. Dietrich, F., Turgut, A., Palm, D., Louw, L.: Smart contract-based blockchain solution to reduce supply chain risks. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) APMS 2020. IAICT, vol. 592, pp. 165–173. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57997-5_20
8. Dolgui, A., et al.: Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. *Int. J. Prod. Res.* **58**, 2184–2199 (2020)
9. Duong-Trung, N., et al.: Multi-sessions mechanism for decentralized cash on delivery system. *Int. J. Adv. Comput. Sci. Appl* **10**(9) (2019)
10. Ha, X.S., Le, H.T., Metoui, N., Duong-Trung, N.: Dem-cod: novel access-control-based cash on delivery mechanism for decentralized marketplace. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 71–78. IEEE (2020)
11. Ha, X.S., et al.: Scrutinizing trust and transparency in cash on delivery systems. In: Wang, G., Chen, B., Li, W., Di Pietro, R., Yan, X., Han, H. (eds.) SpaCCS 2020. LNCS, vol. 12382, pp. 214–227. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68851-6_15
12. Hasan, H., et al.: Smart contract-based approach for efficient shipment management. *Comput. Ind. Eng.* **136**, 149–159 (2019)
13. Hawashin, D., et al.: Using composable nfts for trading and managing expensive packaged products in the food industry. *IEEE Access* **11**, 10587–10603 (2023)
14. Hung, N., et al.: Revolutionizing real estate: a blockchain, nft, and ipfs multi-platform approach. In: International Conference on Information Integration and Web Intelligence, pp. 68–73. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-48316-5_10
15. Le, H.T., et al.: Introducing multi shippers mechanism for decentralized cash on delivery system. *Int. J. Adv. Comput. Sci. Appl.* **10**(6) (2019)
16. Le, N.T.T., et al.: Assuring non-fraudulent transactions in cash on delivery by introducing double smart contracts. *Int. J. Adv. Comput. Sci. Appl.* **10**(5), 677–684 (2019)
17. Li, J., et al.: Design of supply chain system based on blockchain technology. *Appl. Sci.* **11**, 9744 (2021)

18. Vanditha, M., et al.: Agricultural supply chain management system using blockchain. In: 2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC), pp. 1–4. IEEE (2023)
19. Majdalawieh, M., et al.: Blockchain-based solution for secure and transparent food supply chain network. Peer-to-Peer Network. Appl. **14**, 3831–3850 (2021)
20. Pawar, M.K., et al.: Secure and scalable decentralized supply chain management using ethereum and ipfs platform, pp. 1–5 (2021)
21. Putri, A.N., et al.: Supply chain management serious game using blockchain smart contract. IEEE Access **11**, 131089–131113 (2023)
22. Reddy, G.J., et al.: Farmerschain: a decentralized farmer centric supply chain management system using blockchain and iot. In: 2021 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), pp. 444–449. IEEE (2021)
23. Saberi, S., et al.: Blockchain technology and its relationships to sustainable supply chain management. Int. J. Prod. Res. **57**, 2117–2135 (2018)
24. Shahid, A., et al.: Blockchain-based agri-food supply chain: a complete solution. IEEE Access **8**, 69230–69243 (2020)
25. Son, H.X., et al.: Towards a mechanism for protecting seller's interest of cash on delivery by using smart contract in hyperledger. Int. J. Adv. Comput. Sci. Appl. **10**(4) (2019)
26. Tahmasbzadeh, A., et al.: A blockchain-based approach for data storage in drug supply chain. In: 2023 9th International Conference on Web Research (ICWR), pp. 335–341. IEEE (2023)
27. Waikar, A., et al.: Blockchain and supply chain management: the future of trust and transparency. Int. J. Res. Appl. Sci. Eng. Technol. (2022)
28. Yoo, M., et al.: A study on the transparent price tracing system in supply chain management based on blockchain. Sustainability (2018)



The Role of Artificial Intelligence Technologies in Sustainable Urban Development: A Systematic Survey

Maria Rosaria Sessa^(✉) , Ornella Malandrino , and Antonio Cesarano

Department of Management and Innovation Systems, University of Salerno,
Via Giovanni Paolo II, 132, 84084 Fisciano, SA, Italy
masessa@unisa.it

Abstract. In recent years, Artificial Intelligence (AI) technologies have been increasingly used in various fields of application. From the medical field to environmental protection, transport, law enforcement and security, and many other sectors. Also in the field of Sustainable Urban Development (SUD), Artificial Intelligence technologies can be considered to address the social and economic environmental sustainability challenges of cities. However, the use of AI for Sustainable Urban Development is still an under-explored area of research. In fact, few studies have carried out a systematic assessment of the state of the art on the topic. Therefore, this study aims to present a Systematic Survey on how AI technologies can support Sustainable Urban Development, in order to understand which technologies are most widely used, the potential benefits and any critical issues arising from the use of Artificial Intelligence for sustainable urban regeneration. The Systematic Survey revealed that the use of AI is also developing in this field. Yet, the use of Artificial Intelligence technologies requires meeting important challenges in the near future. These include transparency and traceability in the development of Artificial Intelligence, sustainability and ethics in the use of AI technologies concerning all Stakeholders and the relevant community, and the implementation of policies and regulations regarding the adoption of Artificial Intelligence tools in the context of Sustainable Urban Development.

Keywords: Deep Learning · Green Cities · Artificial Intelligence · Sustainable Development · Neural Network

1 Introduction

Currently, according to the World Bank, about 56% of the world's population lives in cities [1]. This trend will continue to grow in the coming years. It is estimated that the urban population will double its current size by 2050.

Thus, the speed and scale of urbanisation present challenges for cities in terms of economic, environmental and social sustainability. In particular, the expansion of urban spaces to meet growing housing demand leads to pressures on land and natural resources resulting in environmental degradation, food, energy and water insecurity,

and the negative externalities of climate change. It also affects the economic, social and governance aspects of cities. In response, the concept of sustainable development has been pushed to the forefront of urban policies [2]. Sustainable development can be defined as development that ‘meets the needs of the present without compromising the ability of future generations to meet their own needs’ [3].

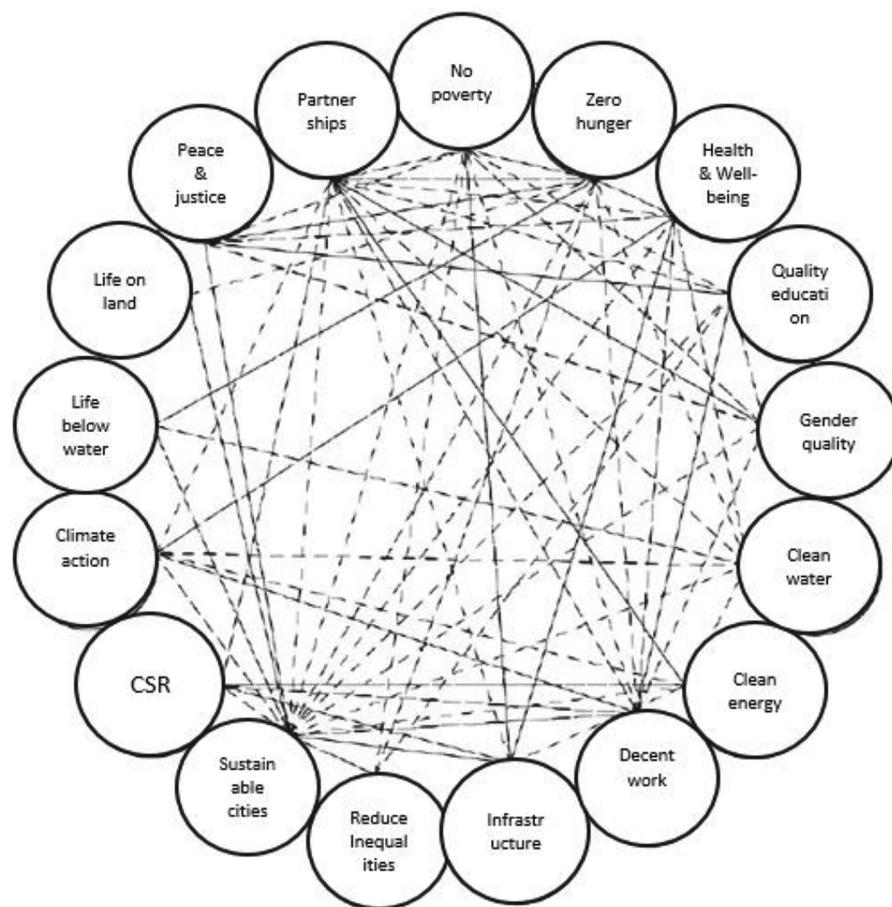


Fig. 1. Relations system between Sustainable Development Goals [3].

In this regard, at the international level, the United Nations (UN) adopted the 2030 Agenda for Sustainable Development provides a shared blueprint for peace and prosperity for people and the planet, now and into the future. At its heart are the 17 Sustainable Development Goals (SDGs), which are an urgent call for action by all countries - developed and developing - in a global partnership. Among these SDGs, Goal 11 refers to making cities and human settlements inclusive, safe, resilient, and sustainable.

Particularly, in the Sustainable Urban Development sphere, meeting SDGs more efficiently, especially those goals that are more directly linked with urban contexts (see Fig. 1) such as SDGs 3, 6, 7, 9, 15 [4, 5], is becoming more achievable with the rapid advances used in technologies [6].

Thus, the possibility of achieving Sustainable Urban Development implies the need to combine the technocentric approach [7], which uses advanced technologies in urban planning, with the Green City Approach [8]. The latter is an integrated, multi-sectoral approach to the well-being, social inclusion, and sustainable development of cities, based on high environmental quality, resource efficiency and circularity, and climate change mitigation and adaptation.

The Green City Approach, promoted by the European Union as part of the Green City Accord (2020) [9] is aimed at defining strategies, tools, and policies ranging from urban and architectural quality to green infrastructure, from urban regeneration to building redevelopment, from mobility to the circular economy, from climate measures to energy measures. The Accord suggests action in five areas of environmental management of cities: air quality, water, nature and biodiversity, waste and circular economy, and noise. The appropriate use of advanced technologies can support the regeneration of urban spaces in smart cities but, at the same time, can favour Sustainable Urban Development.

Thus, this paper aims to present a Systematic Survey of Sustainable Urban Development in which AI technologies are contemplated or applied, in order to understand which technologies are most widely used, the potential benefits and any critical issues arising from the use of Artificial Intelligence for urban spaces regeneration into green cities.

Regarding the methodological approach, this Systematic Survey has followed the PRISMA model and the co-occurrence analysis.

Following this introduction, the remainder of the article is structured as follows. Section 2 refers to the materials and research methods. The results answering the Research Questions (RQs) are presented in Sect. 3. Section 4 discusses the results and the conclusions in Sect. 5.

2 Materials and Methods

This Section describes the procedures and methodologies by which the research was conducted. The choice of them was weighted, first of all, by taking into account the fact that the subject matter, apart from being highly topical, is rather recent. The first applications of AI in urban planning [10], in fact, date back less than a decade, and only for the past few years AI technologies have been used to support Sustainable Urban Development [7]. This explains why there is little literature on the topic, although it is developing. In particular, few contributions [7] have presented a systematic study of the

state of the art of the topic under consideration. The article attempts to fill this gap by conducting a literature review in two stages: the first follows the PRISMA model for conducting bibliometric analysis; the second stage consists of a co-occurrence analysis whose purpose is to bring out the keywords that recur most in the selected articles through the PRISMA model.

2.1 Research Methodology

Following [11], this paper has been developed through the five-stage process (see Fig. 2) theorised by Denyer and Tranfield [12]. Different from a narrative literature review characterised by a subjective and narrative approach to interpreting previous contributions, a Systematic Survey needs to follow a rigorous methodology to avoid the weaknesses of a narrative style and to provide a valid and structured contribution.

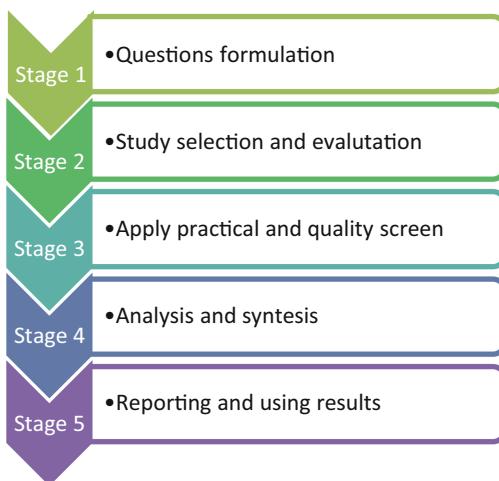


Fig. 2. Steps of Systematic Survey. *Source:* adapted by [11].

According to the research purpose, the following RQs were formulated:

RQ1: How can AI technologies support Sustainable Urban Development?

RQ2: Which AI technologies support Sustainable Urban Development?

The database Scopus was viewed to retrieve the relevant articles on the topic of how and which used AI technologies in Sustainable Urban Development, without any restrictions. The search within the title, abstract, and keywords was conducted in late March 2024. The choice of this platform is linked to its reputation and also because it incorporates more contributions, as well as having a faster citation than its competitors [13].

A well-structured search strategy was developed for retrieving the best articles according to these keywords: ‘urban regeneration’, ‘sustainable urban development’, ‘green cities’, ‘Artificial Intelligence’ and ‘AI’. The search string was built with the Boolean logic and therefore the word bindings ‘AND’ and ‘OR’ must be used.

The PRISMA model was used to exclude irrelevant datasets in a meaningful and traceable manner [14]. In refining the library results, eligibility criteria were used. More precisely, these criteria serve as a tool for screening out studies that are not relevant to the research.

The exclusion criteria (EC) were as follows:

- EC1 (Limitation to Data Range). The time range under analysis is from 2014 to 2024, covering the entire period in which articles on the topic were published. Furthermore, it should be noted that the database did not return any results in the years 2014 and 2015. Therefore, the time range is from 2016 to 2024. The limitation, in fact, did not reduce the cluster.
- EC2 (Limitation to Document Type). Only research articles were considered because of the relevance of this type of document to other ones. The query, after this exclusion, dropped to 52 results.
- EC3 (Limitation to Language). Only English-language articles, which are appropriate for presentation to an international audience, were included. When using bibliometric analysis, moreover, better results are obtained if all units in the query are written in a single language [15]. The cluster after the inclusion of this criterion was reduced by two units to 50.
- EC4 (Limitation to Access Type). The criterion of free accessibility was included in order to obtain results consistent with the need for analytical and qualitative control of the query outputs, useful for the deeper exploration of the topic and a better definition of the state of the art in the knowledge of the literature on AI application methods to support sustainable urban regeneration.

The search string with the exclusion criteria is as follows:

(TITLE-ABS-KEY (“urban regeneration”) OR TITLE-ABS-KEY (“sustainable urban development”) OR TITLE-ABS-KEY (“green cities”) AND TITLE-ABS-KEY (“Artificial Intelligence”) OR TITLE-ABS-KEY (“AI”)) AND (LIMIT-TO (DOCTYPE, “ar”)) AND (LIMIT-TO (LANGUAGE, “English”)) AND (LIMIT-TO (OA, “all”)).

After identifying 30 documents that met the search strings and the eligibility criteria, the content was screened specifically by titles and abstracts to apply the Quality Assessment (QA).

The QA was permitted to select the studies that in a Systematic Survey were consistent with the defined purpose (see Fig. 3). This assessment was needed to optimise the potential results.

Following Rieder et al. (2023) [16], it was used the present checklist of reasons:

1. Were the authors, abstract, or keywords explicitly provided?
2. Were the aims or objectives of the study clear?
3. Was the research method of the study explained?
4. Was the presentation of the study findings clear?
5. Was the subject of the research?

A total of 5 documents were not included in the selection of extracted studies as they did not correlate thematically with the topic of the research work.

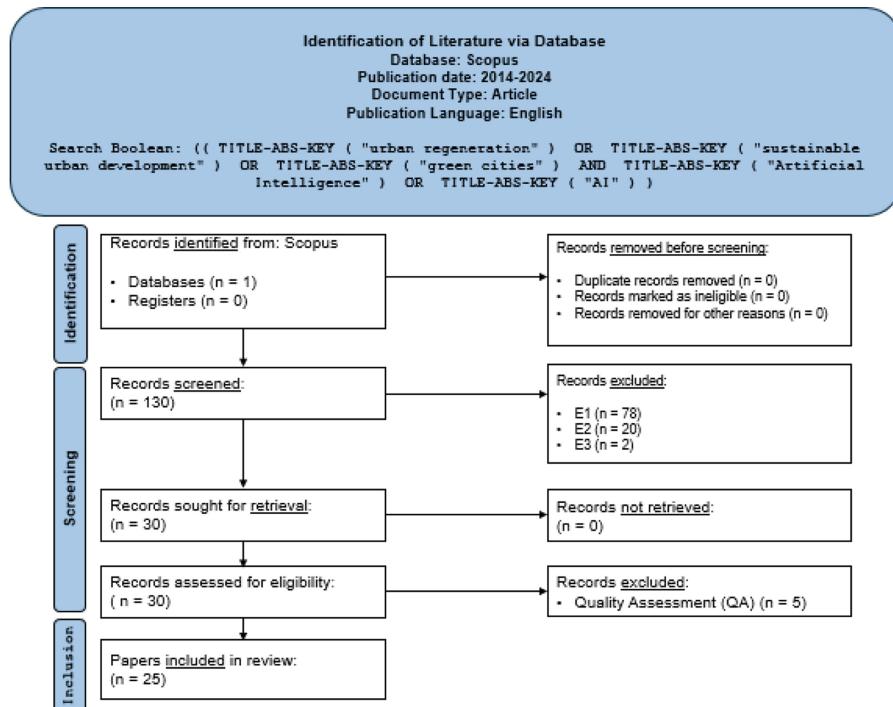


Fig. 3. PRISMA flowchart for literature search and selection. Source: adapted by [7].

3 Results

Before presenting the answers to research questions, based on the results obtained of the methodology approach implemented, in the following study characteristics and an overview of the interest over the years in the chosen topic and the distribution of publications in different subject areas and countries.

From the initial search string, 130 documents were extracted. Following the selection criteria presented above, 25 documents were extracted after the Quality Assessment.

The authors with the highest number of citations (153) for the work ‘Assessment of the Added-Value of Sentinel-2 for Detecting Built-up Areas’ are Pesaresi et al. (2016). The study focuses on the applicability of the Symbolic Machine Learning image classification method on Sentinel-2¹ images to support the enhancement of images of urban and green areas extracted from satellites [17]. Yigitcanlar et al. (2020) authors obtained 121 citations with the work ‘Can Building “Artificially Intelligent Cities” Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar’s Perspective’. The article aims to analyse the positive and negative impacts of smart cities, with a focus on the ability of AI-monitored cities to safeguard inhabitants from natural disasters [18]. The topic is of particular interest as the irrational urbanisation of

¹ Sentinel-2 is a mission developed by ESA as part of the Copernicus Programme to monitor the planet’s green areas and provide support in dealing with natural disasters.

recent years has had an impact on the ecological safety of cities [19]. This has prompted us to reflect on the need to make urban forestry more efficient and to act through intelligent monitoring of its costs and risks, especially in densely populated areas [20]. In the ‘Understanding Sensor Cities: Insights from Technology Giant Company Driven Smart Urbanism Practices’ article, D’Amico et al. authors (2020) recorded 50 citations until March 2024. In this paper, the authors investigate the effects of the sensor city approach on the sustainable development of cities, highlighting the challenges that new technologies face [21]. The only author to have contributed more than one article to the research query is Yigitcanlar T., with two articles, among others among the most cited, exploring Sustainable Urban Development through Artificial Intelligence of Things (AIoT) and Artificial Intelligence Sensory Techniques.

Taking a look at the number of documents by year (see Fig. 4), it can be seen that the highest number of contributions published on these topics was reached in 2023, with an upward trend in 2024.

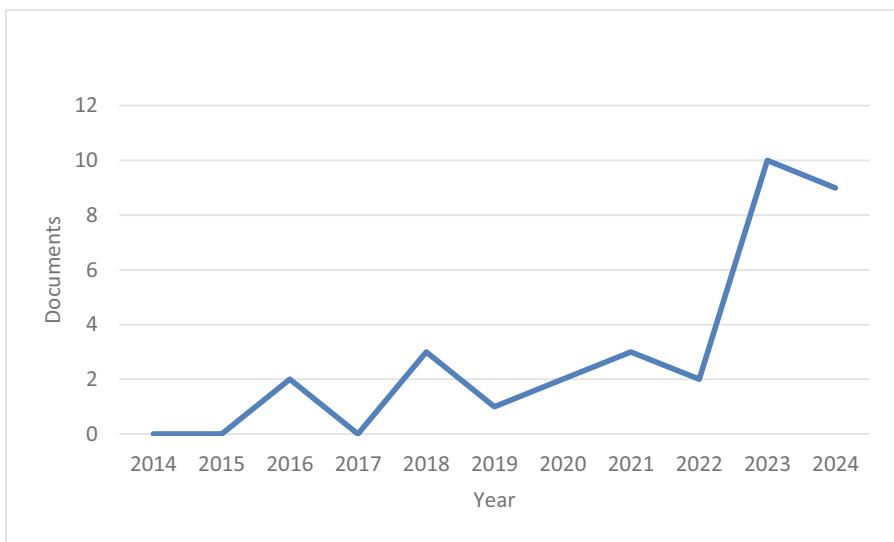


Fig. 4. Documents by year.

The country with the most articles is China, where the eco-sustainable city model is spreading, led by the capital Shanghai [22]. This is followed by Spain. While Italy counts are 2 articles concerning the selected search query.

The subject areas (see Fig. 5) in which there has been a greater production are Computer Science and Social Sciences. But the topic has been discussed in several disciplines, reflecting the interdisciplinary and horizontal interest that characterises the phenomenon.

A co-occurrence analysis (see Fig. 6) was carried out using VosViewer with the dataset derived from the search query used in Scopus, which made it possible to analyse the repeated association of key elements linking the various articles in the cluster.

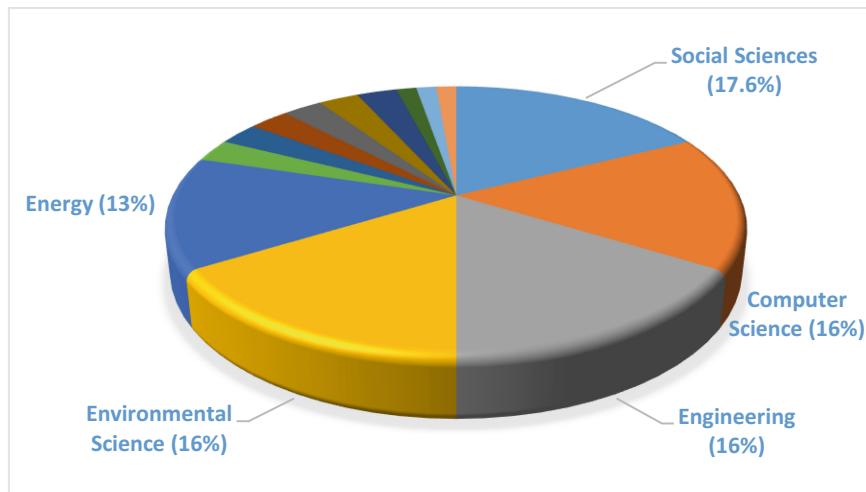


Fig. 5. Documents by subject area.

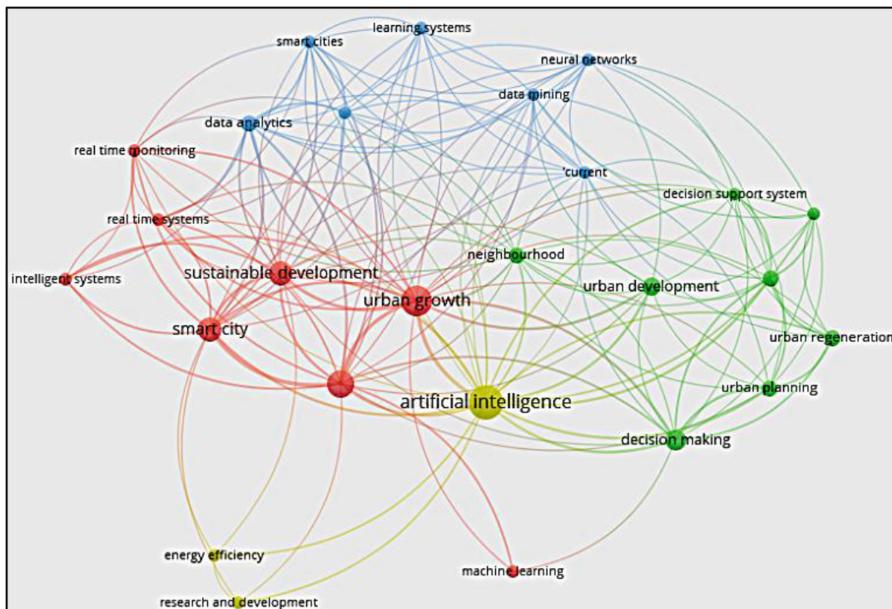


Fig. 6. Co-occurrence analysis of the most frequent keywords.

The analysis produced four clusters.

Red Cluster: 'Sustainable Urban Development' which is the first macro topic to be analysed. The phenomenon of Sustainable Urban Development can be divided into four dimensions: environmental protection; economic development; social justice and equity; culture and governance [15].

Yellow Cluster: ‘Artificial Intelligence’ is the second macro topic of this article. Artificial Intelligence is considered a tool to support sustainable urban regeneration. An interesting study conducted from 2006 to 2019 on 282 prefecture-level cities shows the evolution of their energy efficiency levels, identifying AI as the cause of their increase [23]. Note how in some cases the relationship between AI and SUD is not unidirectional, as demonstrated in the case of autonomous driving of cars that can be a valuable tool to reduce emissions and accidents in cities [24].

Blue cluster: ‘Data Mining and Analysis’. The analysis of co-occurrences highlights in this cluster the most used techniques in the field of Artificial Intelligence at the service of sustainable urban planning, including new theorisations.

In particular, the theorisation of a new algorithm to meet new urban planning challenges including those of the 2030 Agenda. The name of this algorithm is UrbanGenoGAN [25]. It combines the generative power of Generative Adversarial Networks (GANs), the optimisation capabilities of genetic optimisation algorithms and the spatial analysis capabilities of GIS to generate optimised urban plans. Also of interest is the recent theorisation of an integrated approach for predicting air quality data. This is an EEMD-CEEMDAN-GCN model, enhanced by the intelligence of things (AIoT), whose effectiveness has already been successfully tested in four provinces in China [26]. This combined approach allows, through the first two ensemble models (EEMD and CEEMDAN), to decompose complex input signals into simpler components called Intrinsic Mode Functions (IMF), and to make prospective predictions on the data thanks to the Graph Convolutional Network (GCN), a particular type of neural network used for processing data with a specific structure. Beyond the new theorisations and specific case studies, a recurring pattern has emerged in how AI manages the vast amount of data that circulates through the network in cities, to produce research insights useful for guiding urban regeneration plans. Thus, the use of AI technologies in this field implies some choices: data sources (e.g. public Wi-Fi infrastructures in the cities [27]; mobile phones [28], satellites [29], public databases...); data mining and analysis processes that, starting from the raw data, collect useful information and present it in a standardised way; the creation of analytical models that allow the machine, starting from standardised input data, to make and/or suggest decisions and/or perform actions in an automatic or semi-automatic manner. The last phase can be realised through Deep Learning and Computer Vision techniques. Some examples of Deep Learning implementations in sustainable urban development are: recurrent neural networks, useful for traffic forecasting, efficient management of public transport and the electricity grid; multi-objective optimisation systems, to calibrate choices in the presence of conflicting objectives, such as saving energy and maintaining comfort; multi-agent systems for simulating human behaviour; GANs for managing simulated scenarios for urban planning; and ensemble approaches where multiple AI models are used in combination to improve overall performance compared to using a single model. While, an interesting case study showing how Computer Vision can be used as an output of useful information to monitor and increase social welfare was conducted in the historic centre of Macau, a UNESCO World Heritage Site, in which an eye-tracking system was identified as a tool to support design that takes into account the visual perceptions of those living in the city [30]. The same

type of perceptual analysis was conducted to study the urban regeneration of certain neighbourhoods in Lisbon [31].

Green Cluster: ‘Policy Decision Support’. The keywords mapped in this cluster suggest that the purpose of the use of AI technologies in the context of the SUD is to support the formulation of strategies adopted by policy-makers [32], i.e. the institutional actors in charge of guiding the sustainable regeneration of urban spaces. In particular, AI can be taken into account through digital twin techniques [33] that provide a virtual replica of the urban environment, favouring a simpler and more intuitive management of the monitoring, simulation and real-time analysis of urban systems, as well as favouring consensus building [34]. It should be noted that the aspect of policy decision support is most emphasised in articles that focus their studies on single policy areas [35, 36, 37]. The specification of the area of intervention probably allows for more targeted and direct results aimed at supporting the decisions of policy-makers.

4 Discussion

The Systematic Survey presented in this paper represents an initial contribution to the state of the art (from 2014 to 2024) about how the use of Artificial Intelligence technologies to support Sustainable Urban Development. At this time, there are still few contributions on the subject in the literature, although the trend is increasing.

Therefore, an attempt was made with this study to provide an overview of the research objective and answer the research questions related to the objective presented in Sect. 2.1.

From the review work, it emerged that AI can support sustainable urban development through multiple channels, such as: optimising energy efficiency, in transport, public areas and private buildings where research is being invested in ZEB buildings [38]; reducing pollution and environmental monitoring; simulated urban planning to know upstream the possible effects of an urban project; improving waste, traffic and water management; increasing citizens’ involvement in urban planning choices; and improving public safety. AI techniques in this field are innumerable and mainly concern intelligent learning systems such as machine learning, deep learning and computer vision. Section 3 presents the implementation of some of these techniques in the field of sustainable urban development such as neural networks, ensemble approaches, reinforcement learning and segmentation [39, 40].

However, many of the AI applications in SUD are still at an early stage, with applications mostly at an experimental or small-scale level. Several factors, in addition to residual research limitations, are holding back the utilisation of AI techniques within SUD.

In fact, the literature review presented in this paper suggests the presence of some critical issues related to the use of AI in the context of SUD. In particular, the need to use large amounts of personal data clashes with a delicate issue, that of the use of sensitive data, concerning which international regulations are moving towards a restrictive policy. Furthermore, the use of AI technologies in the context of urban regeneration can have considerable impacts on people’s lives, so its use and scope must be understood by all users. Finally, while such technologies can support the sustainable development of cities, the question arises as to how sustainable the use of AI technologies can be. In fact,

according to a study carried out by the University of Massachusetts, it is estimated that the development of an Artificial Intelligence model entails the emission of 284 tonnes of carbon dioxide, i.e. the equivalent of five times the impact that a car has in its entire life cycle [41]. Therefore, the question arises whether the use of AI technologies can support the regeneration of urban spaces according to the sustainability paradigm.

Sustainability and ethical responsibility in the use of AI technologies, as well as the definition of rules and laws related to the use of AI may represent new challenges in this field.

5 Conclusions

The Systematic Survey conducted provided an overview of the topic. From the results presented, it was possible to answer the research questions posed, confirming that AI technologies can support SUD as they can analyse the large amounts of data present in urban spaces, as well as improve the planning and management of resources (such as energy and water) and reduce waste production. Yet, it is necessary to present some limitations of the research. The first limitation refers to the use of only one database (Scopus) for the collection of contributions. This made it impossible to view further contributions in other databases such as IEEE Xplore, PubMed, Google Scholar, and Web of Science. The choice of these keywords in the search string may have led to the omission of important contributions in the literature. Thus, the choice of a single database and certain keywords resulted in a small number of contributions being analysed to answer the search questions. Another limitation concerns the choice to take into consideration only research papers published in open-access journals. In fact, all other types of contributions (book chapters, conference proceedings and so on) were excluded. In fact, all other types of contributions (book chapters, conference proceedings and so on) were excluded. This could have generated a bias in the Systematic Survey.

Furthermore, Sustainable Urban Development is a very broad field of study, so this study may not be suitable to explain all areas of the use of AI technologies in the promotion of SUD.

Therefore, the Systematic Survey conducted, without any claim to exhaustiveness, represents an embryonal contribution to the definition of the state of the art regarding the use of AI technologies in Sustainable Urban Development. It is necessary to develop future lines of research to respond to the further challenges identified on the topic and consider real applications to support that the introduction of AI in the urban context leads to real benefits in economic, social and environmental terms for cities.

References

1. The World Bank. Urban Development. <https://www.worldbank.org/en/topic/urbandevelopment/overview#1>. Accessed 25 Apr 2024
2. Yigitcanlar, T., Teriman, S.: Rethinking sustainable urban development: towards an integrated planning and development process. *Int. J. Environ. Sci. Technol.* **12**(1), 341–352 (2015). <https://doi.org/10.1007/s13762-013-0491-x>
3. EUR-Lex. Sustainable Development. <https://eur-lex.europa.eu/IT/legal-content/glossary/sustainable-development.html>. Accessed 25 Apr 2024

4. Nesticò, A., Guarini, M.R., Morano, P., Sica, F.: An economic analysis algorithm for urban forestry projects. *Sustainability* **11**, 314 (2019). <https://doi.org/10.3390/su11020314>
5. Zhou, M., Ma, Y., Tu, J., Wang, M.: SDG-oriented multi-scenario sustainable land-use simulation under the background of urban expansion. *Environ. Sci. Pollut. Res.* **29**(48), 72797–72818 (2022). <https://doi.org/10.1007/s11356-022-20904-9>
6. Sanchez, T., Shumway, H., Gordner, T., Lim, T.: The prospects of artificial intelligence in urban planning. *Int. J. Urban Sci.* (2022). <https://doi.org/10.1080/12265934.2022.2102538>
7. Son, T.H., Weedon, Z., Yigitcanlar, T., Sanchez, T., Corchado, J.C., Mehmood, R.: Algorithmic urban planning for smart and sustainable development: systematic review of the literature. *Sustain. Cities Soc.* **94** (2023). <https://doi.org/10.1016/j.scs.2023.104562>
8. Green City Network. <https://www.greencitynetwork.it/green-city-approach/>. Accessed 25 Apr 2024
9. European Commission. https://environment.ec.europa.eu/topics/urban-environment/green-city-accord_en. Accessed 25 Apr 2024
10. Crawford, K.: The Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence. Yale University Press, London (2021)
11. Esposito, B., Sessa, M.R., Sica, D., Malandrino, O.: Towards circular economy in the agri-food sector. A systematic literature review. *Sustainability* **12**(18), 7401 (2020). <https://doi.org/10.3390/su12187401>
12. Denyer, D., Tranfield, D.: Using qualitative research synthesis to build an actionable knowledge base. *Manag. Decis.* **44**, 213–227 (2008)
13. Falagas, M.E., Pitsouni, E.I., Malietzis, G.A., Pappas, G.: Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses. *FASEB J.* **22**(2), 338–42 (2008). <https://doi.org/10.1096/fj.07-9492LSF>
14. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G.: The preferred reporting items for systematic reviews and meta-analyses: the PRISMA Statement. *PLoS Med.* **6**(7), e1000097 (2009). <https://doi.org/10.1371/journal.pmed.1000097>
15. Dharmani, P., Das, S., Prashar, S.: A bibliometric analysis of creative industries: current trends and future directions. *J. Bus. Res.* **135**, 252–267 (2021)
16. Rieder, E., Schmuck, M., Tugui, A.: A scientific perspective on using artificial intelligence in sustainable urban development. *Big Data Cogn. Comput.* **7**(3), (2023). <https://doi.org/10.3390/bdcc7010003>
17. Pesaresi, M., Corbane, C., Julea, A., Florczyk, A.J., Syrris, V., Soille, P.: Assessment of the added-value of sentinel-2 for detecting built-up areas. *Remote Sens.* **8**, 299 (2016). <https://doi.org/10.3390/rs8040299>
18. Yigitcanlar, T., Butler, L., Emily Windle, E., Desouza, K.C., Mehmood, R., Corchado, J.M.: Can building “artificially intelligent cities” safeguard humanity from natural disasters, pandemics, and other catastrophes? An urban scholar’s perspective. *Sensors* **20**, 2988 (2020). <https://doi.org/10.3390/s20102988>
19. Wei, S., Cheng, S.: An artificial intelligence approach for identifying efficient urban forest indicators on ecosystem service assessment. *Front. Environ. Sci.* **10** (2022)
20. Huang, J., Geng, H.: Investigating of spatiotemporal correlation between urban spatial form and urban ecological resilience: a case study of the city cluster in the Yangzi river midstream, China. *Buildings* **14**, 274 (2024). <https://doi.org/10.3390/buildings14010274>
21. D’Amico, G., L’Abbate, P., Liao, W., Yigitcanlar, T., Ioppolo, G.: Understanding sensor cities: insights from technology giant company driven smart urbanism practices. *Sensors* **20**, 4391 (2020). <https://doi.org/10.3390/s20164391>
22. Den Hartog, H.: Shanghai’s regenerated industrial waterfronts: urban lab for sustainability transitions? *Urban Plan.* **6**(3), 181–196 (2021)

23. Li, X., Wang, Q., Tang, Y.: The impact of artificial intelligence development on urban energy efficiency-based on the perspective of smart city policy. *Sustainability* **16**, 3200 (2024). <https://doi.org/10.3390/su16083200>
24. Kuru, K., Khan, W.: A framework for the synergistic integration of fully autonomous ground vehicles with smart city. *IEEE Access* **9** (2021). <https://doi.org/10.1109/ACCESS.2020.3046999>
25. Cheng W., Chu Y., Xia C., Zhang, B., Chen, J., Jia, M., Wang, W.: UrbanGenoGAN: pioneering urban spatial planning using the synergistic integration of GAN, GA, and GIS. *Front. Environ. Sci.* **11** (2023). <https://doi.org/10.3389/fenvs.2023.1287858>
26. Bhatti, M.A., Song, Z., Bhatti, U.A., Syam, M.S.: AIoT-driven multi-source sensor emission monitoring and forecasting using multi-source sensor integration with reduced noise series decomposition. *J. Cloud Comput.* **13**(65), (2024)
27. Salas, P., Ramos, V., Ruiz-Perez, M., Alorda-Ladaria, B.: Methodological proposal for the analysis of urban mobility using Wi-Fi data and artificial intelligence techniques: the case of palma. *Electronics* **12**(3), 504 (2023). <https://doi.org/10.3390/electronics12030504>
28. Dong, H., Chen, Y., Huang, X.: A new framework for analysis of the spatial patterns of 15-minute neighbourhood green space to enhance carbon sequestration performance: a case study in Nanjing, China. *Ecol. Indic.* **156**, 111196 (2023). <https://doi.org/10.1016/j.ecolind.2023.111196>
29. Das, M., Mandal, A., Das, A., Inacio, M., Pereira, P.: Urban dynamics and its impact on habitat and eco-environmental quality along urban-rural gradient in an urban agglomeration (India). *Environ. Chall.* **14**, 100824 (2024). <https://doi.org/10.1016/j.envc.2023.100824>
30. Wang, P., Song, W., Zhou, J., Tan, J., Wang, H.: AI-based environmental color system in achieving sustainable urban development. *Systems* **11**(3), 135 (2023). <https://doi.org/10.3390/systems11030135>
31. Serrano-Jiménez, A., Lima, M.L., Molina-Huelva, M., Barrios-Padura, A.: Promoting urban regeneration and aging in place: APRAM – an interdisciplinary method to support decision-making in building renovation. *Sustain. Cities Soc.* **47**, 101505 (2019). <https://doi.org/10.1016/j.scs.2019.101505>
32. Obaideen, K., Albasha, L., Iqbal, U., Mir, A.: Wireless power transfer: applications, challenges, barriers, and the role of AI in achieving sustainable development goals - a bibliometric analysis. *Energy Strateg. Rev.* **53**, 101376 (2024). <https://doi.org/10.1016/j.esr.2024.101376>
33. Gkонтзis, A.F., Sotiris, K.S., Feretzakis, G., Verykios, V.S.: Enhancing urban resilience: smart city data analyses, forecasts, and digital twin techniques at the neighborhood level. *Future Internet* **16**(2), 47 (2024). <https://doi.org/10.3390/fi16020047>
34. Mortaheb, R., Jankowski, P.: Smart city re-imagined: city planning and GeoAI in the age of big data. *J. Urban Manag.* **12**(1), 4–15 (2023). <https://doi.org/10.1016/j.jum.2022.08.001>
35. Mercader-Moyano, P., Camporeale, P., Serrano-Jiménez, A.: Integrated urban regeneration for high-rise multi-family buildings by providing a multidimensional assessment model and decision support system. *J. Build. Eng.* **76**, 107359 (2023). <https://doi.org/10.1016/j.jobe.2023.107359>
36. Abarca-Alvarez, F.J., Campos-Sanchez, F.S., Reinoso-Bellido, R.: Demographic and dwelling models by artificial intelligence: urban renewal opportunities in Spanish Coast. *Int. J. Sustain. Dev. Plan.* **13**(7), 941–953 (2018)
37. Panteleeva, M., Borozdina, S.: Sustainable urban development strategic initiatives. *Sustainability* **14**(1), 37 (2022). <https://doi.org/10.3390/su14010037>
38. Jin, B., Bae, Y.: Prospective research trend analysis on zero-energy building (ZEB): an artificial intelligence approach. *Sustainability* **15**(18), 13577 (2023)

39. Nosratabadi, S., Mosavi, A., Keivani, R., Ardabili, S., Aram, F.: State of the art survey of deep learning and machine learning models for smart cities and urban sustainability. In: Várkonyi-Kóczy, A. (eds.) INTER-ACADEMIA 2019. LNNS, vol. 101. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-36841-8_22
40. Marasinghe, R., Yigitcanlar, T., Mayere, S., Washington, T., Limb, M.: Computer vision applications for urban planning: a systematic review of opportunities and constraints. *Sustain. Cities Soc.* **100**, 105047 (2024). <https://doi.org/10.1016/j.scs.2023.105047>
41. Hao, K.: Training a single AI model can emit as much carbon as five cars in their lifetimes. *MIT Technol. Rev.* <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/>. Accessed 25 Apr 2024



Enhancing Privacy in Machine Unlearning: Posterior Perturbation Against Membership Inference Attack

Chen Chen[✉], Hengzhu Liu^{ID}, Huanhuan Chi^{ID}, and Ping Xiong^(✉)^{ID}

School of Information Engineering, Zhongnan University of Economics and Law,
Wuhan, China
pingxiong@zuel.edu.cn

Abstract. Machine unlearning aims to safeguard data privacy by mitigating the data's impact on machine learning models. Nonetheless, machine unlearning practices can introduce new privacy vulnerabilities, leaving models susceptible to various forms of attack, such as confidence attack and label-only attacks. Existing defense methods encounter challenges in striking a balance between defending against attacks and sustaining model performance. In this paper, we propose a posterior perturbation method to defend against membership inference attacks by randomizing the model's outputs without requiring adjustments. To prevent confidence attacks, we employ optimization algorithms to generate adversarial noise that disrupts the model outputs' confidence scores, thereby obscuring the output differences between the original and unlearned models. We also propose a label perturbation method to defend against label-only attacks by randomizing the model's output labels through high-dimensional sphere sampling. The experimental results demonstrate that our proposed method effectively defends against membership inference attacks while maintaining the model's performance.

Keywords: Privacy preservation · Machine unlearning · Label-only attack · Membership inference attack

1 Introduction

Data has become the cornerstone of the current digital age. As a foundational resource, data plays an indispensable role in various machine learning and artificial intelligence applications, such as medical diagnosis [20], autonomous driving [21], and video generation (e.g., Sora) [1]. However, exploring data poses significant risks to individual privacy. Therefore, many countries have implemented corresponding regulations, such as the European Union's General Data Protection Regulation (GDPR) [15], the California Consumer Privacy Act (CCPA) in the United States [14], and the Personal Information Protection Law in China [7], to protect user data privacy and the *right to be forgotten*. As a result, data collectors must delete a user's data upon receiving a recall request, and remove

the data's contribution in the machine learning process from the trained model. This process is known as *machine unlearning* [6].

A straightforward approach to achieving machine unlearning is retraining the initial model using the training dataset after removing the requested data. However, this generates significant computational costs, especially for deep models with large training datasets. Thus, many machine unlearning methods have been proposed in recent years, which aim to obtain an unlearned model by updating the original without retraining from scratch. For example, Bourtoule et al. [6] proposed SISA(Sharded, Isolated, Sliced, and Aggregated training), a typical partition-based method, which splits the training set into several non-overlapping shards and trains submodels for the disjointed sets. This ensures that unlearning can be achieved by only retraining the corresponding submodel. Guo et al. [11] proposed a certified removal mechanism, which can offset the data's influence on the model by updating the parameters. Meanwhile, Chundawat et al. [9] presented a zero-shot unlearning algorithm that erases data without retraining operations and training sample access requirements.

Existing solutions have demonstrated excellent machine unlearning performance, ensuring that the unlearned model's distribution is precisely or approximately similar to that of the retrained model [27]. However, we assert that machine unlearning should not simply use the retrained model as the optimization target, as this may introduce additional risks to the unlearned data through exploiting the contrast between the initial and retrained models. Chen et al. [8] investigated the unintended information leakage caused by machine unlearning. They discovered that membership inference attacks (MIAs) can be successfully implemented by exploiting the prediction vectors of the original and unlearned models. Furthermore, Lu et al. [19] proposed label-only MIAs for machine unlearning. The study revealed that MIAs are can efficiently leverage the predicted labels without posterior dependence.

Therefore, we assert that retraining the model with the request date deleted can completely eliminate the data's influence, but it also exposes data privacy of to the greatest extent due to the differences between the initial and retrained models. For a given sample, the model's output is typically differs depending on whether the sample is a member of the training set. Consequently, this leaves the deleted machine unlearning data vulnerable to MIAs.

In this paper, we propose the posterior perturbation-based method to effectively defend against unlearned models against MIAs. This method randomizes the unlearned model's output instead of modifying the model, thereby mitigating the privacy leakage risks caused by machine unlearning. Specifically, we regard the target models (i.e., the original and unlearned models), as black boxes, and propose the adversarial noise (ANP) and label perturbation (LP) algorithms to defend against the *confidence* [8] and *label-only attacks* [19], respectively. Particularly, the ANP algorithm aims to blur the differences in the confidence scores output by the original and unlearned models by adding optimized noise. Meanwhile the LP algorithm defends against *abel-only attacks* by distorting the predicted labels produced by the models using a high-dimensional sphere sampling

approach. The experimental results demonstrate the proposed algorithms' effectiveness in defending against the attacks while maintaining high model accuracy.

Our contributions can be summarized as follows:

- We propose the ANP algorithm to effectively defend against *confidence attacks* during machine unlearning.
- We propose the LP algorithm to defend against *label-only attacks* with a low model accuracy cost.
- We conduct experiments on real-world datasets, demonstrating our algorithms' superiority compared to existing methods.

The remainder of this paper is organized as follows. We introduce the background and related work in Sect. 2. Then, we propose the methods for defending against confidence and label-only attacks in Sects. 3 and 4, respectively. Section 5 presents the experimental results, followed by the conclusion in Sect. 6.

2 Preliminaries

2.1 Definitions and Notations

Machine Unlearning. Machine unlearning involves selectively removing certain samples from the training dataset and updating the model parameters to mitigate their effects without completely discarding existing knowledge. This process can be formalized as follows:

$$A(D_o, D_f, M_o) \rightarrow M_u, \quad (1)$$

where $A(\cdot)$ is an unlearning algorithm, D_o denotes the original training dataset, M_o represents the original model trained on D_o , D_f is the sample set to be deleted, and M_u signifies the unlearned model. Suppose D_r is the retained training dataset with D_f deleted from D_o , and M_r is the model retrained on D_r , the M_u and M_r distributions are required to be the identical or approximate.

Membership Inference. Membership inference is a privacy disclosure attack that determines whether specific samples are present in a machine learning training set [24]. Generally, adversaries collect the target machine learning model's responses to specific inputs, and use them to establish an attack model, which infers whether the target sample belongs to the training dataset. Depending on the the attack scenario, membership inference is usually divided into two types: white- and black-box attacks. During a white-box attack, the adversary has full access to the target model, including its architectureand parameters. Meanwhile, during black-box attacks, the attacker has limited or no access to the model's internal structure and weights.

In addition, depending on the target model's outputs, MIAs can be further classified into confidence [8] and label-only attacks [19]. During confidence

attacks, the adversary submits queries and analyzes the confidence scores provided by the model to infer the target sample’s membership status. In contrast, during label-only attacks, the adversary can only query the model and analyze the returned labels.

Furthermore, during machine unlearning, adversaries can utilize the original and unlearned models’ outputs to analyze their differences, thereby increasing the attack’s success rate. In this paper, we focus on the black-box attack scenario and propose methods to defend against confidence and label-only attacks.

2.2 Problem Definition

Machine learning as a service (MLaaS) allows users to access trained models through APIs or web interfaces for data prediction tasks. Due to the users’ data deletion requests, let M_o be the original model trained on dataset D_o . The model owner can update the M_o using an unlearning algorithm $A(\cdot)$ for the unlearned model M_u .

Suppose \mathbf{x} is one of the samples deleted from D_o , \mathbf{x} is a member of D_o and not the retained dataset D_r . Thus, we can let \mathbf{s}_u be the models’ outputs for sample \mathbf{x} . These two outputs generally exhibit significant differences because the model’s responses to seen and unseen data typically differs significantly, especially in cases of overfitting. Existing research has shown that adversaries can effectively launch MIAs using \mathbf{s}_o , \mathbf{s}_u , and their differences [8].

Specifically, during confidence attacks, attackers can acquire the model’s predicted confidence scores for target samples. Consequently, creating an optimal noise vector \mathbf{n} that achieves the following objectives is necessary: (i) lowering the probability of successful confidence attacks by incorporating noise into the confidence score \mathbf{s}_u , and (ii) ensuring that the noise minimally impact the unlearned model’s accuracy.

Meanwhile, during a label-only attack, the adversary can only access the model’s predicted label l_u for the target sample \mathbf{x} . Typically, the adversary locates one the samples \mathbf{x}' nearest to the target sample \mathbf{x} near the decision boundary, where \mathbf{x}' has a different label l'_u from \mathbf{x} . Then, they evaluate \mathbf{x} ’s membership status by measuring the similarity between \mathbf{x} and \mathbf{x}' . A high similarity level suggests a likelihood of non-membership, and vice versa. The underlying concept posits that unseen samples tend to lie closer to the decision boundary. Therefore, introducing randomness to the output label is essential to achieving the following goals: (i) increasing the the model’s likelihood of outputting incorrect labels so that the target sample appears closer to the decision boundary than it is, and (ii) ensuring that the randomness in the model’s output does not significantly impact its accuracy.

2.3 Related Work

Privacy risks arising from machine unlearning have received significant attention in recent years.

Gupta et al. [13] proposed an attack against SISA, utilizing the model's interpretability to infer the data shard to which the target sample belongs. Likewise, Chen et al. [8] proposed a confidence MIA to infer whether a target sample belongs to the training dataset by leveraging the different outputs of the original and unlearned model. They also introduced several possible defenses against this attack, such as publishing the only top- k confidence values or the labels. However, they observed that publishing only the top k confidence values may not be adequately effective against privacy attacks [23]. Lu et al. [19] proposed a label-only MIA and demonstrated that publishing the label only does not effectively prevent membership inference.

To impede these attacks, several defense methods have been proposed. For example, Jia et al. [17] proposed MemGuard to defend against confidence MIAs by adding noise to the model's outputs. This method uses temperature scaling to reduce the impact of a single sample, thereby mitigating overfitting and guarding against confidence attacks [12]. Similarly, Graves et al. [10] proposed *amnesiac unlearning* to defend against model inversion attacks and MIAs by retraining small batches of distorted labels and reversing parameter updates.

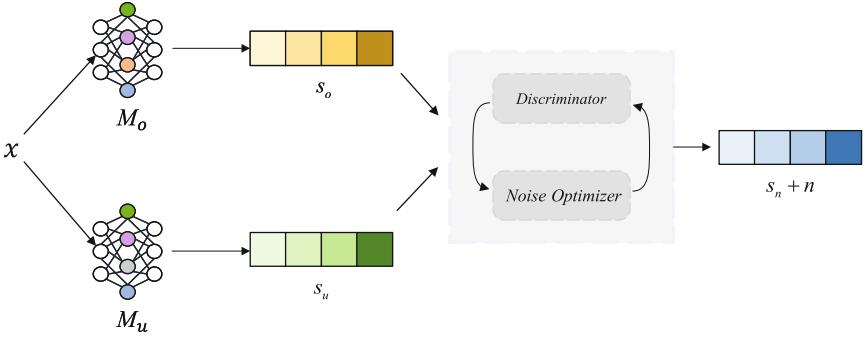
For *label-only attacks* [19], applying differential privacy [16, 18] can limit the impact of individual samples on outputs, and introduce randomness to the output labels. Tang et al. [25] proposed a self-distillation approach with an integrated architecture to diminish the implicit training set information contained in machine learning models. Meanwhile, Rajabi et al. [22] employed query sampling within a high-dimensional sphere centered on input and aggregate predictions to distort the boundary samples' true labels. Additionally, Ye et al. [28] introduced meta-reinforcement learning to differentiate malicious queries and strategically adapt model labels.

3 Defense Against Confidence Attacks

3.1 Adversarial Noise Perturbation Method Overview

In this section, we propose ANP to defend against *confidence attacks* by adding noise to the model. ANP comprises of two key components, a discriminator and a noise optimizer, as shown in Fig. 1. The discriminator acts as the membership inference attacker, while the noise optimizer adjusts the noise based on the discriminator's results. These components optimize the noise in iterative adversarial interactions, enabling the noise \mathbf{n} to resist MIAs when added to \mathbf{s}_u without significantly decreasing the model's performance.

Specifically, given a prediction query for sample \mathbf{x} , the target models compute the corresponding confidence scores \mathbf{s}_o and \mathbf{s}_u . Then, the score vectors are fed to the discriminator to infer whether \mathbf{x} is a member of the training set. According to the discriminator loss, the optimizer generates a noise \mathbf{n} to perturb the output \mathbf{s}_u using $\mathbf{s}_u + \mathbf{n}$, which is fed to the discriminator again. This iterative optimization process is repeated until the stopping criterion is met. Finally, the unlearned model releases $\mathbf{s}_u + \mathbf{n}$ to the requester.

**Fig. 1.** ANP workflow.

It's important to note that we assume the requester obtains \mathbf{s}_o in advance, which is the original model's output for \mathbf{x} .

3.2 Defense Module

This study aims to mitigate the differences in the target model's outputs that can lead to privacy leakage. To achieve this, ANP generates an optimized noise to perturb the unlearned model's outputs to: (i) mislead the discriminator's classification results, and (ii) minimize the performance loss.

We formalize these two requirements using the following constraints:

$$M^* = \underset{M}{\operatorname{argmin}} |E_M(g(\mathbf{s}_o, \mathbf{s}_u + \mathbf{n}) - 0.5)| \quad (2)$$

$$\arg \max \{s_u^j + n^j\} = \arg \max \{s_u^j\} \quad (3)$$

$$s_u^j + n^j \geq 0, \forall j \quad (4)$$

$$\sum_j s_u^j + n^j = 1 \quad (5)$$

where $g(\mathbf{s}_o, \mathbf{s}_u + \mathbf{n}_u)$ denotes the discriminator's confidence level in identifying an unlearned sample, and s_u^j and n^j represent the j th entries of \mathbf{s}_u and \mathbf{n} , respectively.

Equation 2 ensures that the discriminator's success probability is approximately 0.5. Equation 3 ensures that the noise's impact on model performance is minimal. Furthermore, a threshold can be set to balance the tradeoff between privacy protection and model performance. Finally, Eqs. 4 and 5 ensure that the model outputs after adding noise still conform to the probability distribution.

Discriminator. The discriminator is a binary classification model based on the MIA method. It identifies whether a sample has been unlearned from the original model.

Formally, we labeled a given sample \mathbf{x} with $\{in/out\}$ to indicate if it is an unlearned sample. Then, several shadow datasets with distributions similar to the original were constructed. These shadow datasets undergo the same training and unlearning processes as the target model, resulting in multiple shadow models. In addition, the confidence vectors \mathbf{s}_o and \mathbf{s}_u , obtained by inputting \mathbf{x} into the corresponding shadow models were used as original features. The $\{in/out\}$ label is the target label for constructing a new sample to train the discriminator. Finally, the scores output by the discriminator represent the confidence that \mathbf{x} is a member of the training set:

$$g : \mathbf{s} \rightarrow [0, 1]. \quad (6)$$

If the output of g is closer to 0 or 1, the target sample is more likely to be a member or non-member, while an output close to 0.5 suggests uncertainty.

Noise Optimizer. The noise optimizer generates a noise vector \mathbf{n} that satisfies the four constraints. Our approach involves performing a noise optimization process inspired by the adversarial sample.

As mentioned above, the noise generated by the optimizer must meet the four constraints in Eqs. 2–5. Therefore, we used the confidence output generated by the discriminator as a metric for evaluating the noise’s alignment with the constraints specified in Eq. 2, and formalize a loss function denoted as L_1 :

$$L_1 = |g(\mathbf{s}_o, \mathbf{s}_u + \mathbf{n}_u) - 0.5| \quad (7)$$

Furthermore, let s_m be the largest entry in \mathbf{s} . We aim for the largest entry in the noise added, transforming $\mathbf{s} + \mathbf{n}$ to $s_m + n_m$ and maintaining the unlearned model’s performance.

Let the m th entry, s_m , be the largest \mathbf{s}_u entry. We expect the largest entry in $\mathbf{s}_u + \mathbf{n}$ to also be the m th entry. This ensures that the unlearned model’s output remains consistent with that of the original, thereby maintaining performance. To achieve this, we defined the loss function L_2 according to Eq. 3 as:

$$L_2 = \max_j \{s_j + n_j\} - (s_m + n_m). \quad (8)$$

Based on the above two evaluation metrics, we constructed the noise optimizer’s loss function as:

$$\text{Loss} = L_1 + L_2. \quad (9)$$

In each optimization round, the optimizer calculates the loss value and adjusts the noise until the predefined convergence conditions are met. This process ultimately yields the adversarial noise \mathbf{n} . The detailed ANP process is described in Algorithm 1.

Algorithm 1. Adversarial noise perturbation

Input: original prediction \mathbf{s}_o , initial prediction from unlearned model \mathbf{s} , discriminator algorithm g , threshold of noise ϵ , max_iter , learning rate α

Output: Perturbed prediction \mathbf{s}'

```

1: if  $|g(\mathbf{s}_o, \mathbf{s}) - 0.5| < 1e - 4$  then
2:    $\mathbf{s}' = \mathbf{s}$ 
3: else
4:   Initialize  $\mathbf{n}$ ,  $\sum_j n_j = 0$ ,  $|n_j| \leq \epsilon, \forall j$ 
5:    $\mathbf{s}' = softmax(\mathbf{s} + \mathbf{n})$ 
6:    $i = 1$ 
7:   while  $i < max\_iter$  and ( $argmax\{\mathbf{s}'\} \neq argmax\{\mathbf{s}\}$  or  $|g(\mathbf{s}_o, \mathbf{s}') - 0.5| \geq 1e - 4$ )
do
8:      $L_1 = |g(\mathbf{s}_o, \mathbf{s}') - 0.5|$ 
9:      $L_2 = max_j\{s_j + n_j\} - (s_m + n_m)$ .
10:    Loss= $L_1 + L_2$ 
11:    gradient= $\frac{\partial Loss}{\partial s'}$ 
12:    gradient = gradient/ $\|\text{gradient}\|_2$ 
13:     $\mathbf{s}' = \mathbf{s}' - \alpha * \text{gradient}$ 
14:     $\mathbf{s}' = softmax(\mathbf{s}')$ 
15:     $i=i+1$ 
16: end while
17: end if
18: return  $\mathbf{s}'$ 

```

If the input confidence vector \mathbf{s} is sufficiently resistant to MIAs, the algorithm directly outputs \mathbf{s} (i.e., lines 1–3). Then, line 4 generates a random noise vector \mathbf{n} , constrained by $\sum_j n_j = 0$ and $|n_j| \leq \epsilon, \forall j$ to satisfy Eqs. 4 and 5. ANP then performs loop optimisation for \mathbf{n} . Before the start of each loop, line 7 firstly checks whether \mathbf{s}' has satisfied the defense requirement. If it is not satisfied, lines 8–15 are performed iteratively to optimize the noise using the gradient descent based on the loss function in Eq. 9.

4 Defense Against Label-Only Attacks

4.1 Label Perturbation Method Overview

In this section, we propose the two-stage LP approach for defending against label-only attacks. The underlying assumption of label-only attacks is that training samples are usually far from the decision boundary, and the farther away from the boundary they are, the higher the confidence in being inferred as a member. Thus, given a target sample \mathbf{x} , LP moves it towards the decision boundary and prompts the model to output its label at the new position. However, LP sacrifices samples that are located near the decision boundary, as they might be dragged to the other side, resulting in the model outputting incorrect labels. This is the model accuracy expense required to defend against label-only attacks.

Specifically, LP consists of following stages:

- *Locating a boundary supporting sample:* Given a query input sample \mathbf{x} , we utilized the knowledge representation approach [26] to find the boundary support sample $\bar{\mathbf{x}}$. We move from \mathbf{x} to $\bar{\mathbf{x}}$ toward the decision boundary. Given a predefined distance d , we drag \mathbf{x} to a substitute sample $\hat{\mathbf{x}}$, where the distance $|\mathbf{x} - \hat{\mathbf{x}}|$ is d .
- *High-dimensional sphere sampling:* To increase randomness in the second stage, we constructed a high-dimensional sphere around $\hat{\mathbf{x}}$. Several samples are randomly taken from this sphere, and their confidence scores are averaged to produce the final label for \mathbf{x} .

The LP workflow is illustrated in Fig. 2.

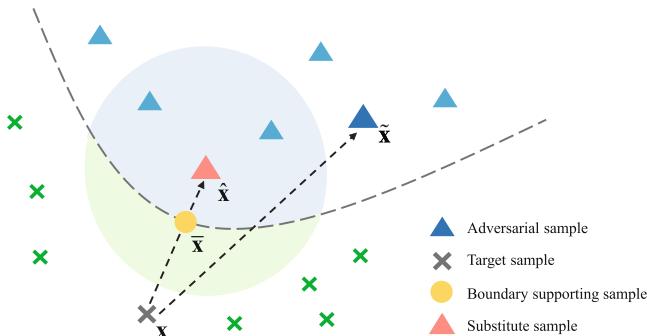


Fig. 2. Workflow overview against label-only attacks

As shown in Fig. 2, LP causes the model to alter the original prediction labels when the input target sample is close to the decision boundary. This mechanism prevents the attacker from finding the true boundary samples, thereby leading to the failure of label-only attacks.

4.2 Label Perturbation

In this section, we detail the two stages of LP, combining them into a linear workflow to randomize the model’s output.

Locating the Boundary Supporting Sample. In this stage, the direction in which the target sample is moving toward the decision boundary is determined. For each input sample \mathbf{x} , we randomly selected an adversarial sample $\tilde{\mathbf{x}}$ in the category c which differs from the predicted \mathbf{x} label. After that, we employed the MinAD algorithm [26] to find the boundary supporting sample $\bar{\mathbf{x}}$ based on \mathbf{x} and $\tilde{\mathbf{x}}$.

Specifically, given \mathbf{x} and $\tilde{\mathbf{x}}$, the MinAD algorithm iteratively performs the following steps to find the projection sample $\bar{\mathbf{x}}$ of \mathbf{x} on the decision boundary.

This becomes the boundary supporting sample: (i) Use binary search to find the boundary sample of \mathbf{x} in the direction of $\tilde{\mathbf{x}}$; (ii) Approximate the normal vector at the decision boundary in (i) using the Monte Carlo estimation method; (iii) Calculate the cosine similarity between the normal vector and the line connecting \mathbf{x} and $\tilde{\mathbf{x}}$ as the loss function; (iv) Update $\tilde{\mathbf{x}}$ using the gradient descent method based on the loss function in (iii).

Finally, we pulled \mathbf{x} to a substitute sample $\hat{\mathbf{x}}$ in the direction toward $\tilde{\mathbf{x}}$ using a predefined distance d .

High-Dimensional Sphere Sampling. After finding the substitute sample $\hat{\mathbf{x}}$, we constructed a high-dimensional sphere around $\hat{\mathbf{x}}$. Subsequently, m samples are randomly sampled from the sphere and input into the model to obtain all the predictions. Then, these predictions are averaged as the model’s final prediction of \mathbf{x} . Specifically, we used a zero-mean Gaussian noise with variance σ^2 to perturb $\hat{\mathbf{x}}$ to simulate the high-dimensional sphere’s random sampling process. The radius r of the sphere is directly affected by the value of σ taken.

As shown in Fig. 2, high-dimensional sphere sampling can confuse the classification results of samples in the decision boundary’s vicinity. The degree of confusion is positively correlated with the σ value [22]. This stage causes the attacker to obtain inaccurate sample information, thereby defending the data. The LP flow is displayed in Algorithm 2.

Algorithm 2. Label perturbation

Input: input sample \mathbf{x} , distance d , sample size m , noise variance σ^2 , target model M
Output: Defended label y'

```

1: Initialize an random adversarial sample  $\tilde{\mathbf{x}}$ 
2:  $\tilde{\mathbf{x}} = \text{MinAD}(\mathbf{x}, \tilde{\mathbf{x}}, M)$ 
3:  $\mathbf{u} = \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\|\tilde{\mathbf{x}} - \mathbf{x}\|}$ 
4:  $\hat{\mathbf{x}} = \mathbf{x} + d \cdot \mathbf{u}$ 
5:  $\text{predictions} \leftarrow \{\}$ 
6: for  $i = 1$  to  $m$  do
7:    $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2 I)$ 
8:    $\mathbf{x}_{noisy} = \hat{\mathbf{x}} + \mathbf{n}_i$ 
9:    $\text{predictions} \leftarrow \text{predictions} \cup \{M(\mathbf{x}_{noisy})\}$ 
10: end for
11:  $y' \leftarrow \text{average}(\text{predictions})$ 
12: return  $y'$ 

```

Lines 1–4 correspond to the locating the boundary supporting sample stage. First, the boundary supporting sample $\tilde{\mathbf{x}}$ is computed based on the $\tilde{\mathbf{x}}$ and \mathbf{x} in line 2. Subsequently, the direction vector \mathbf{u} from \mathbf{x} to $\tilde{\mathbf{x}}$ is unitized in line 3. Then, in line 4 \mathbf{x} is dragged toward \mathbf{u} using a distance d to obtain the substitute sample $\hat{\mathbf{x}}$. In lines 5–11 a high-dimensional sphere sampling sampling is implemented around $\hat{\mathbf{x}}$. Specifically, Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ is integrated into \mathbf{x} in lines 7–8 to generate perturbation samples \mathbf{x}_{noisy} . Then, the predicted labels of each

\mathbf{x}_{noisy} are recorded in line 9. Finally, all the \mathbf{x}_{noisy} predictions are averaged in line 11 to produce the final label for the external output.

Moreover, the LP algorithm's sample prediction randomization process negatively impacts model performance. Hence, we strive to strike a balance between the defense efficacy and performance degradation. Therefore, the optimal d and σ values should be predetermined for the model using methods such as bisection lookup. We explore this perspective in Sect. 5.2.

5 Experiments

This section discusses the experiments conducted on various datasets representing distinct scenarios to comprehensively assess the ANP and LP algorithms' performance. Specifically, we explore the ANP algorithm's efficacy across various datasets and target models. Then, we analyze the ANP's defense success rate and its impact on the unlearned model's performance. Finally, we examine the ANP's convergence rate and discuss its practical application value. For LP, we demonstrate its effectiveness and advantages compared to existing methods. In addition, we explore the potential relationship between defense efficacy and model performance loss.

5.1 Settings, Datasets, and Measurements

Datasets. To evaluate our proposed method's performance in different scenarios, we conducted experiments using four publicly available datasets.

UCI Adults [2]: Adults is a publicly available category dataset comprising approximately 49,000 samples, and 14 demographic and employment-related attributes. These attributes are used for predicting whether the annual income of each sample's corresponding citizen exceeds \$50,000.

US Accident [3]: This dataset is a publicly available category dataset that contains approximately 3.5 million detailed records on motor vehicle accidents in 49 states. Each sample contains approximately 30 effective features.

MNIST [4]: The MNIST dataset is a widely used benchmark in the image classification field. It comprises a training and test set of approximately 60,000 and 10,000 samples, respectively. Each sample in the dataset is a grayscale image of a handwritten digit from 0 to 9, with a dimension of 28×28 pixels.

CIFAR-10 [5]: The CIFAR-10 dataset comprises a training and test set of approximately 50,000 and 10,000 images, respectively. Each sample in the dataset is a color image with a dimension of 32×32 pixels. This dataset aims to classify 10 common object categories.

Settings. To validate the proposed method, constructing is necessary construct the target model and the attack classifier to simulate the real attack scenario. Therefore, we divided the original dataset into target D_t and an attack datasets D_a . Within the ANP framework, D_t is further partitioned into a dataset D_d for training the discriminator.

To ensure fair comparison, we constructed the target model and the attack classifier following the workflow adopted by Chen et al. [8] and Lu et al. [19]. Subsequently, we trained a logistic regression model similar to the attack classifier as the discriminator on D_d with the Adam optimizer by Pytorch. Specifically, we sampled n_o subsets from each dataset. Each subset S contains n_s samples as the initial training set for the original model M_o . Then, we separately unlearned n_u samples from M_o to obtain n_u values that correspond to the unlearned model M_u . We selected single-sample unlearning because it can better validate our method’s effectiveness based on its vulnerability to MIAs [8, 19]. Furthermore, we retrained the model from scratch to replicate the most rigorous unlearning scenario.

By default, we set $n_o = 20$, $n_s = 1000$, and $n_u = 10$. For ANP, we set the *max_iter* of the optimizer to 100 for the category and 500 for the image group. Meanwhile, we fixed $\epsilon = 0.1$ and $\alpha = 1$. For LP, the sample distance d is set to 10. These settings demonstrated excellent performance in achieving good balance between defense and model performance in Sect. 5.2.

These experiments were conducted on a server with a 12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40 GHz, 90 GB of RAM, and an RTX 3080 Ti (12 GB) GPU using the ubuntu 20.04 operating system.

Evaluation Metrics. We investigated our method from three key perspectives: defense efficacy, impact on model performance, and convergence rate. Specifically, we introduced three metrics: *accuracy (ACC)*, *area under the receiver operating characteristic curve (AUC)*, *successful defense rate (SDR)*.

- *ACC* measures the overall correctness of the classification results, indicating the proportion of correctly classified instances among the total number of instances.
- *AUC* quantifies the model’s ability to discriminate between positive and negative instances across various threshold settings, providing insight into the model’s predictive power and robustness against false positives and negatives.
- *SDR* represents ANP’s success rate in perturbing confidence scores, transforming them into adversarial samples.

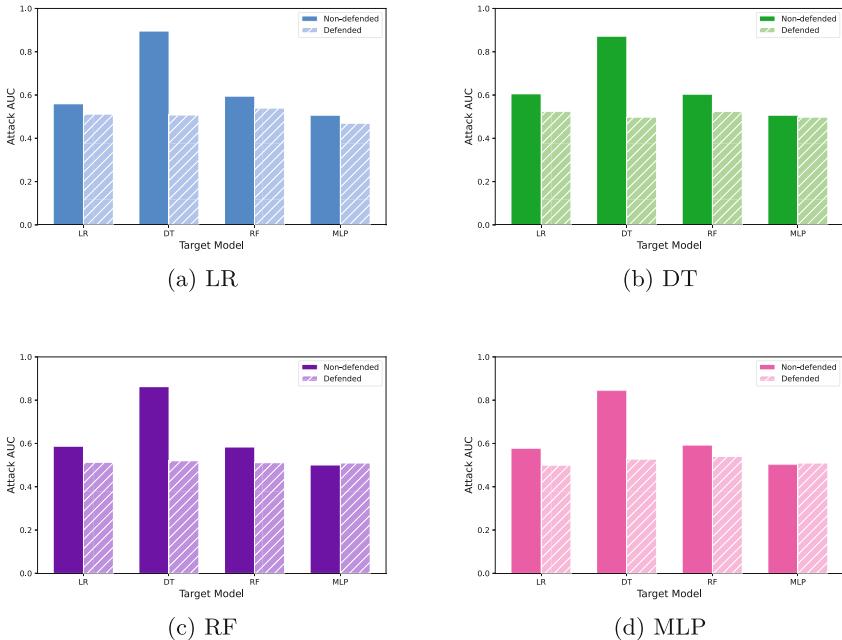
5.2 Experiment Results and Comparisons

ANP Efficacy Against Confidence Attacks. We explore ANP’s defense efficacy using the category and image datasets. We constructed each category datasets target model using regression (LR), decision tree (DT), random forest (RF), and multi-layer perceptron (MLP). We selected SimpleCNN and DenseNet for the MNIST and CIFAR10 datasets, respectively. During this process, every other parameters were fixed except for target model change(e.g. ϵ and α .).

First, we selected DT as the attack model and chose the set demonstrating the most effective attack on each dataset to explore ANP’s defensive performance. The experimental results in Table 1 show that ANP can significantly converge the

Table 1. Attack performance comparisons.

Dataset	Without ANP		With ANP	
	ACC	AUC	ACC	AUC
Adults	0.834	0.881	0.497	0.524
Accident	0.874	0.910	0.504	0.521
MNIST	0.522	0.518	0.501	0.513
CIFAR10	0.696	0.794	0.553	0.569

**Fig. 3.** Defensive performance on the Adults dataset. Each subgraph represents different attack models.

ACC and AUC attack classifier metrics to 50%, which approaches the random guess level.

To further explore ANP'S generalizability on different attack models, we conducted experiments with multiple attack classifiers using LR, DT, RF, and MLP. The experimental results are shown in Fig. 3 and Fig. 4.

For the category dataset, Fig. 3 illustrates that when the target model structure remains relatively stable, ANP can effectively defend against the attack classifiers. Although DT's structure is fragile and more prone to privacy breaches, our defense framework can mitigate the attack AUC by up to 38.9%. Regarding image datasets and neural network models, as demonstrated in Fig. 4, ANP also

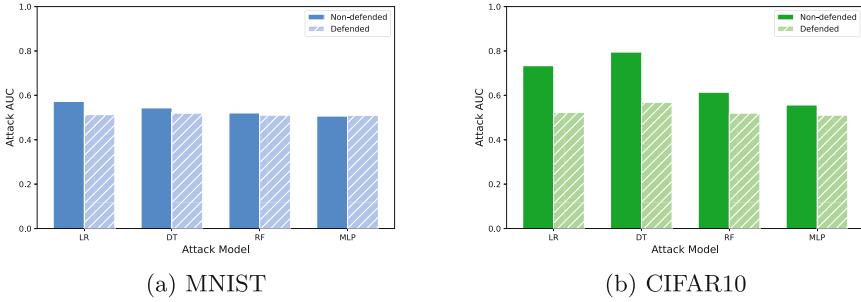


Fig. 4. Defensive performance on image datasets. In each subgraph, the groups on the x-axis represent different attack models.

Table 2. Model performance comparisons.

Dataset	Adults	Accidents	MNIST	CIFAR 10
Without ANP	0.842	0.763	0.969	0.474
With ANP	0.833	0.756	0.971	0.452
SDR	0.975	0.954	0.912	0.623

achieves a satisfactory defensive performance. For the DenseNet model, which is prone to overfitting [8], the adversarial noise effectively reduces the attack AUC from 79.4% to 56.9%. The above results confirm that our proposed ANP algorithm effectively safeguards against confidence attacks on machine unlearning and is broadly applicable across diverse datasets and target models.

ANP Impact on Model Performance. To further explore the actual impact of adversarial noise on the model output, we examine the performance on each dataset. The model which achieves the best performance without ANP is selected as the tested model. We used ACC to evaluate the performance of each model and measure the corresponding SDR. The experimental results are shown in Table 2.

The experimental results demonstrate that the unlearned models defended by ANP suffer little performance loss. This is attributed to Eq. 8 limiting the excessive noise perturbation. Meanwhile, the noise optimization continuously smooths the confidence score, thereby reducing the impact of overfitting on the model's accuracy.

Convergence Rate of ANP. We investigate the ANP's convergence rate on multiple datasets with progressive iterations. Specifically, we continuously varied the *max_iter* parameter in Algorithm 1 to observe the changes in defense performance, using the attack AUC and SDR as references. The experimental results in Fig. 5 show that ANP exhibits an adequate convergence speed on both the category and image datasets.

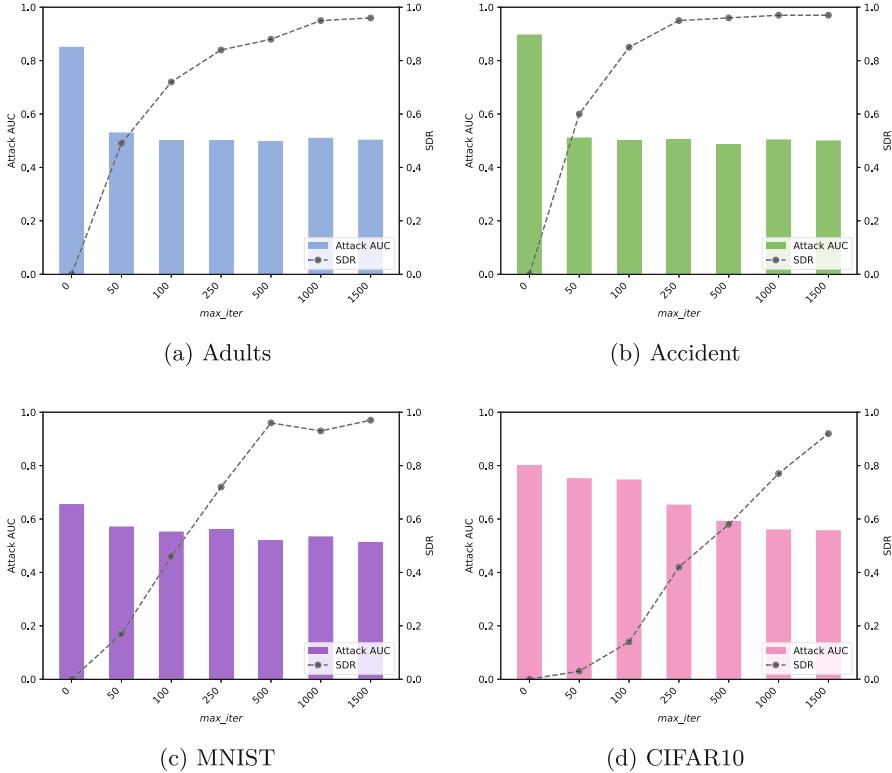


Fig. 5. Convergence rate of ANP with varying max_iter .

As the max_iter increases, ANP can effectively perturb over 90% of the confidence vectors, significantly reducing the attack AUC to nearly 50%. In addition, the convergence rate is higher on the category dataset. This can be attributed to the need to remove a large volume of sensitive information through more perturbation on the image dataset. Simultaneously, ANP demonstrates consistent convergence thresholds across category and image datasets, indicating its capacity for efficient computational performance across diverse machine learning scenarios.

LP Against Label-Only Attack. Finally, we evaluated LP's defense efficacy and its impact on model performance using the MNIST dataset. We divided the baselines into groups without defense and those with the differential privacy (DP) exponential mechanism. Additionally, we used LDL [22] for comparison. Considering the impact of hyperparameters on LP, we designed multiple σ settings.

The results shown in Fig. 6 demonstrate that LP is effective in defending against label-only attacks, reducing the AUC can be reduced from 65% to 53%.

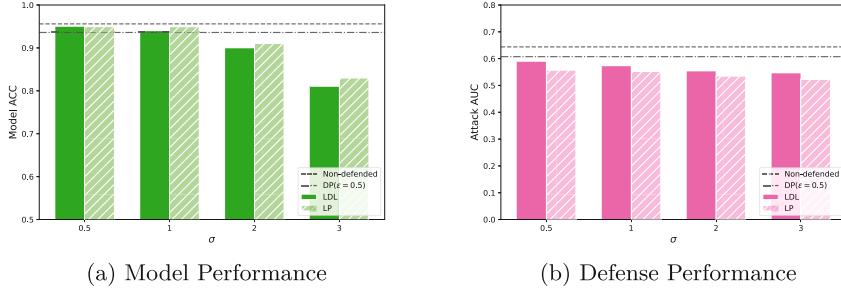


Fig. 6. Comparisons with LDL. The horizontal axis represents different σ settings.

Among the three approaches, LP achieves the best defense efficacy with the same hyperparameters or the similar level of performance loss. In addition, the experimental results indicate that the defense efficacy is negatively correlated with model performance. In addition, as the radius of the high-dimensional sphere increases, the model's final output becomes more randomized.

6 Conclusions

In this paper, we address the additional privacy leakages caused by the differences in performance between the original and unlearned model. However, achieving a balance between defense effectiveness and model performance with existing methods is challenging [19]. To overcome these problems, we proposed a posterior perturbation method to enhance machine unlearning privacy. First, we applied ANP to the unlearned model's confidence scores to resist *Confidence attacks* [8]. Furthermore, we proposed a controllable LP algorithm for the target model's output to defend against *Label-only attacks* [19]. The experimental results obtained from several real-world datasets demonstrate that our defense strategies can resist MIAs to a level similar to random guessing while ensuring model performance. Future research can be extended chained unlearning scenarios, enabling privacy protection for systems that have multiple active historical versions.

References

1. <https://openai.com/sora/>
2. <https://archive.ics.uci.edu/dataset/2/adult>
3. <https://www.kaggle.com/sobhanmoosavi/us-accidents>
4. <http://yann.lecun.com/exdb/mnist/>
5. <https://www.cs.toronto.edu/~kriz/cifar.html>
6. Bourtoule, L., et al.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 141–159. IEEE (2021)

7. Chen, J., Sun, J.: Understanding the Chinese data security law. *Int. Cybersecur. Law Rev.* **2**(2), 209–221 (2021)
8. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y.: When machine unlearning jeopardizes privacy. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (2020). <https://api.semanticscholar.org/CorpusID:218502126>
9. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.S.: Zero-shot machine unlearning. *IEEE Trans. Inf. Forensics Secur.* **18**, 2345–2354 (2023)
10. Graves, L., Nagisetty, V., Ganesh, V.: Amnesiac machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11516–11524 (2021)
11. Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. arXiv preprint [arXiv:1911.03030](https://arxiv.org/abs/1911.03030) (2019)
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, vol. 70, pp. 1321–1330 (2017)
13. Gupta, V., Jung, C., Neel, S., Roth, A., Sharifi-Malvajerdi, S., Waites, C.: Adaptive machine unlearning. *Adv. Neural. Inf. Process. Syst.* **34**, 16319–16330 (2021)
14. Harding, E.L., Vanto, J.J., Clark, R., Hannah Ji, L., Ainsworth, S.C.: Understanding the scope and impact of the California Consumer Privacy Act of 2018. *J. Data Protect. Priv.* **2**(3), 234–253 (2019). <https://ideas.repec.org/a/aza/jdpp00/y2019v2i3p234-253.html>
15. Hoofnagle, C.J., van der Sloot, B., Borgesius, F.Z.: The European union general data protection regulation: what it is and what it means. *Inf. Commun. Technol. Law* **28**(1), 65–98 (2019). <https://doi.org/10.1080/13600834.2019.1573501>
16. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: 28th USENIX Security Symposium (USENIX Security 19), pp. 1895–1912 (2019)
17. Jia, J., Salem, A., Backes, M., Zhang, Y., Gong, N.Z.: Memguard: defending against black-box membership inference attacks via adversarial examples. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 259–274 (2019)
18. Liu, Y., et al.: Ml-doctor: holistic risk assessment of inference attacks against machine learning models. In: 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, 10–12 August 2022, pp. 4525–4542 (2022)
19. Lu, Z., Liang, H., Zhao, M., Lv, Q., Liang, T., Wang, Y.: Label-only membership inference attacks on machine unlearning without dependence of posteriors. *Int. J. Intell. Syst.* **37**, 9424 – 9441 (2022). <https://api.semanticscholar.org/CorpusID:251664530>
20. Mahoto, N.A., Shaikh, A., Sulaiman, A., Reshan, M.S.A., Rajab, A., Rajab, K.: A machine learning based data modeling for medical diagnosis. *Biomed. Signal Process. Control* **81**, 104481 (2023). <https://doi.org/10.1016/j.bspc.2022.104481>. <https://www.sciencedirect.com/science/article/pii/S1746809422009351>
21. Muhammad, K., Ullah, A., Lloret, J., Ser, J.D., de Albuquerque, V.H.C.: Deep learning for safe autonomous driving: current challenges and future directions. *IEEE Trans. Intell. Transp. Syst.* **22**(7), 4316–4336 (2021). <https://doi.org/10.1109/TITS.2020.3032227>
22. Rajabi, A., Sahabandu, D., Niu, L., Ramasubramanian, B., Poovendran, R.: Ldl: a defense for label-based membership inference attacks. In: Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security, pp. 95–108 (2023)

23. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18 (2016). <https://api.semanticscholar.org/CorpusID:10488675>
24. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18 (2017). <https://doi.org/10.1109/SP.2017.41>
25. Tang, X., et al.: Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In: 31st USENIX Security Symposium (USENIX Security 2022), pp. 1433–1450 (2022)
26. Tian, Z., Wang, Z., Abdelmoniem, A.M., Liu, G., Wang, C.: Knowledge representation of training data with adversarial examples supporting decision boundary. *IEEE Trans. Inf. Forensics Secur.* (2023)
27. Xu, H., Zhu, T., Zhang, L., Zhou, W., Yu, P.S.: Machine unlearning: a survey. *ACM Comput. Surv.* **56**(1) (2023). <https://doi.org/10.1145/3603620>
28. Ye, D., Zhu, T., Gao, K., Zhou, W.: Defending against label-only attacks via meta-reinforcement learning. *IEEE Trans. Inf. Forensics Secur.* (2024)



PEbfs: Implement High-Performance Breadth-First Search on PEZY-SC3s

Weihao Guo^{1,2}, Qinglin Wang^{1,2}(✉), Xiaodong Liu³, Muchun Peng^{1,2}, Shun Yang^{1,2}, Yaling Liang^{1,2}, Yongzhen Shi^{1,2}, Ligang Cao^{1,2}, and Jie Liu^{1,2}

¹ Laboratory of Digitizing Software for Frontier Equipment, National University of Defense Technology, Changsha 410073, China

{guoweihaowangqinglin, pengmuchun, yangshun, yalingliang, shiyongzhen, caoligang, liujie}@nudt.edu.cn

² National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Changsha 410073, China

³ Engineering Research Center for National Fundamental Software, National University of Defense Technology, Changsha 410073, China

liuxiaodong@nudt.edu.cn

Abstract. The breadth-first search (BFS) algorithm is a fundamental algorithm in graph theory, and its parallelization can significantly improve performance. Therefore, there have been numerous efforts to leverage the powerful parallel computing capabilities of hardware like GPGPU to implement high-performance BFS algorithms. However, the energy efficiency is relatively low due to the high power consumption of the platforms on which the algorithm is adapted to. To deal with these challenges, this paper introduces PEbfs that is a high-performance BFS algorithm based on the PEZY-SC3s efficient processor. We integrated three search algorithms, two algorithm optimization strategies, and a directional optimization scheme into PEbfs. Through multiple evaluations of the performance of PEbfs on the public SNAP dataset, the results demonstrate that the average energy efficiency ratio of PEbfs is higher than that of Enterprise and Tigr, the two most advanced implementations on Nvidia's GPGPU: It achieves 3.08× the average energy efficiency ratio of Enterprise and 4.53× that of Tigr.

Keywords: Graph Algorithms · BFS · Parallelization · PEZY-SC3s

1 Introduction

Graph is a universal representation for encoding relationships (e.g., social networks), connections (e.g., road maps), and structures (e.g., molecules) [1], it becomes increasingly crucial to analyze and extract the key information concealed within this data in a timely manner [2]. In the era of information explosion, the scale and complexity of graph data are experiencing exponential growth. Therefore, in order to process data more effectively, there is a continuous need to optimize data processing solutions to better suit the current challenges, and

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025
T. Zhu et al. (Eds.): ICA3PP 2024, LNCS 15256, pp. 249–267, 2025.

https://doi.org/10.1007/978-981-96-1551-3_17

the efficient implementation of the breadth-first search (BFS) algorithm is one important solution.

BFS is a fundamental traversal algorithm for graph data structures, and represent a vital sequential traversal approach and serving as a crucial component in various graph algorithms, such as connected components algorithms, centrality algorithms, and maximum flow algorithms [3]. In addition, it effectively characterizes the runtime behavior of graph algorithms, with BFS being employed as a strategy for evaluating the graph traversal capability of computer systems in prominent large-scale graph processing system performance benchmark suites like Graph500 [4]. Diverging from traditional scientific computing paradigms, BFS exhibits a minimal proportion of numerical computations, with its primary operations focusing on random data access, modification, and writing. Consequently, in contrast to conventional scientific computing applications, BFS poses unique challenges to computer systems.

There have been numerous classical works about high-performance implementations of BFS, such as Xbfs [3], Enterprise [4], iBFS [5], and Tigr [6], which have demonstrated excellent graph search performance by leveraging Nvidia's high-performance GPGPU. However, these solutions consume significant power which cannot meet the needs of scenarios with strict energy constraints, such as in some wearable devices, Internet of Things (IoT) devices, remote sensors, and portable medical devices. Considering the aforementioned challenges and concerns, this paper proposes a high-performance breadth-first search algorithm PEbfs, based on the PEZY-SC3s high-efficiency processor. By deeply integrating the search scheme with the unique PEZY architecture and adopt multiple strategies to optimize the search scheme, PEbfs is tailored to adapt to this architecture effectively. Comparative experiments conducted on the SNAP dataset [7] against enterprise and Tigr demonstrate superior performance of PEbfs on the majority of test datasets, showcasing its effectiveness in addressing the identified challenges. The main contributions of this article are as follows:

- As far as we are aware, there have been some attempts to accelerate the BFS algorithm on PEZY-SC processor, but the performance is average and the project has not been fully disclosed yet [8]. PEbfs is the first BFS solution designed specifically for the PEZY-SC3s processor.
- Building upon two classes of sub-algorithms to implement BFS, this paper optimizes the search performance by deeply integrating the search strategies with the hardware architecture. Additionally, strategies such as atomic counting, iterative search, and load balancing are employed to individually enhance the performance of the two sub-algorithms. We devised a parameter selection strategy to determine the optimal directional optimization strategy.
- This paper will provide insights and references for the design of other high-performance algorithms on this platform in the future.

The remainder of this paper is organized as follows. Section 2 provides background information. Section 3 analyzes the core algorithm selection and considerations. Section 4 details the algorithm design and optimization strategies.

Section 5 presents the experimental evaluation of PEbfs. Finally, Sect. 6 concludes the paper.

2 Background

This section introduces some basic background knowledge which includes the BFS algorithm, the architecture of PEZY-SC3s, and some related classic works.

2.1 Breadth-First Search

BFS is a traversal algorithm used for graph data structures, which starts from the initial node and expands layer by layer to discover all reachable nodes and determine their distance from the root. Parallelization can significantly improve performance by improving computational efficiency, speed, and system throughput, and is an important strategy for improving algorithm performance. However, parallel BFS also faces challenges such as data contention, problem irregularity, and poor memory access locality.

The direction-optimized hybrid search method, which integrates both top-down and bottom-up BFS search strategies, represents the current mainstream form. The top-down search strategy starts from the specified root node, checks all neighbor nodes of that node and updates the information of the nodes that have not yet been accessed, and puts them into the frontier queue. The nodes in the frontier queue are called frontier nodes. Repeat this process until all reachable nodes in the graph are traversed to form an effective BFS tree [3].

Another bottom-up search strategy is slightly different from the previous method, where the frontier queue consists of all unreachable nodes. It queries each neighboring node of each frontier node, and if there are any neighbor nodes that have been visited, updates the status information of the current node accordingly. For ease of understanding, we will use the definitions and examples from the classic work Enterprise [4] to illustrate.

Definition: At level i , if there exists a node N in a graph structure G that is a frontier node, then it can only be one of the following two cases.

- Top-down BFS: N is visited at level $i - 1$; or
- Bottom-up BFS: N is unvisited until level $i - 1$.

Compared with the top-down BFS method, the bottom-up BFS strategy is more suitable for situations where there are a large number of nodes waiting to be accessed at the current level. In the top-down BFS method, each neighboring node of each frontier node needs to be accessed, so the workload of each frontier node is equal to its degree. However, in the bottom-up BFS strategy, the child node only need to have one “parent” node. Therefore, when checking its neighbor nodes, once a visited node is found, the neighbor node can be considered as its parent node. At this point, it is no longer necessary to access neighbor nodes that have not been accessed subsequently. This greatly reduces the workload,

and the situation known as early termination [3]. Figure 1 shows an example of a graph data structure containing 15 nodes and a BFS tree originating from node 0.

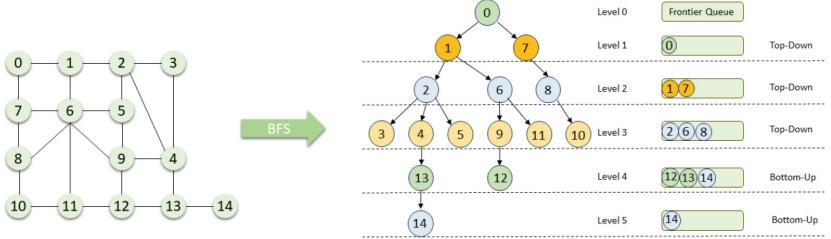


Fig. 1. The left figure is a graph data structure, and the right one is a BFS tree with zero as the starting node.

2.2 PEZY-SC3s

The PEZY-SC3s is a highly energy and area-efficient processor based on the PEZY architecture. This platform combines fine-grained multi-threading technology and non-coherent cache technology, focusing on handling tasks with high thread-level parallelism [9]. Based on a MIMD multicore design as Fig. 2, it achieves higher operational efficiency and offers greater programmability compared to architectures relying solely on function-limited specialized tensor units or wide SIMD architectures [9]. Additionally, its lower power consumption results in significantly superior energy efficiency. In the Green 500 rankings of November 2018, Japan’s Shoubu system B ranked first, utilizing the PEZY architecture [10]. This underscores the unparalleled effectiveness of the PEZY architecture in energy efficiency.

As shown in the Fig. 4, PEZY-SC3s has a hierarchical structure consisting of four layers, namely state, city, village, and processor unit (PE). Each PE has dedicated fast access local storage for fast storage and retrieval. Although the processor includes a multi-level storage structure, it lacks coherent cache technology, so a refresh mechanism is necessary to ensure that the correct data is written back to the specified location. And refresh mechanism as shown in the Fig. 3. The maximum operating speed of PEZY-SC3s atomic operations is 32 operations per clock. PEZY-SC3s has an atomic cache with a size of 32 KB [11]. There is no implicit consistency between atomic cache and other data caches, so it is necessary to refresh atomic cache to maintain data consistency after using atomic operations.

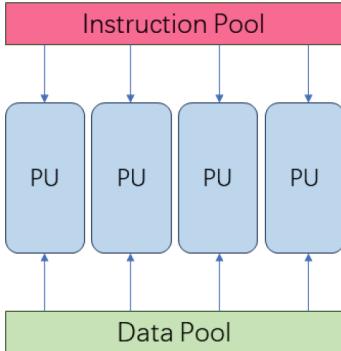


Fig. 2. MIMD Structural Diagram

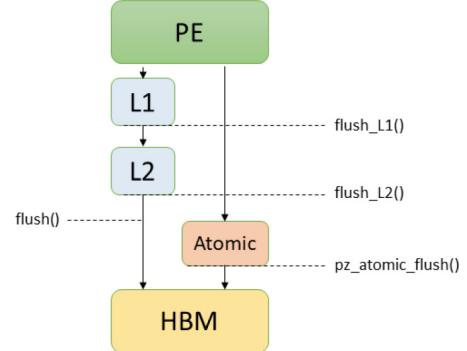


Fig. 3. The refresh mechanism of PEZY-SC3s

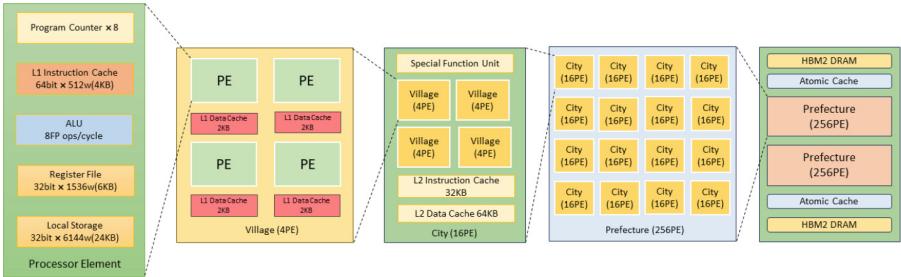


Fig. 4. The schematic diagram of PEZY-SC3s shows its multi-level hierarchical structure from left to right

The PEZY-SC3s processor programming environment supports PZCL, a programming interface analogous to OpenCL. As shown in Fig. 5, programs running on these processors comprise two types of subroutines: kernel programs and host programs. The host program, which is written in C/C++, executes on the host CPU. The PZCL API facilitates memory allocation on PEZY-SC3s, data transfer between the CPU and PEZY-SC3s, and the initiation of kernel programs. Additionally, the SDK (Software Development Kit) for kernel programs includes mathematical function libraries and atomic operation libraries. Kernel programs, written in PZCL C and compiled using LLVM, are executed on PEZY-SC3s. Notably, PZCL C is almost identical to OpenCL C, ensuring ease of transition and familiarity for developers.

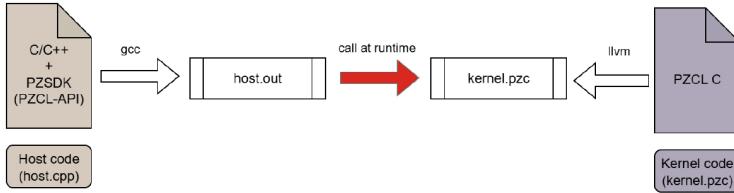


Fig. 5. compilation and execution flow (offline compilation) of PEZY-SC3s programming model

2.3 Related Work

There have been some research works on the PEZY series processors, Yatsuyanagi through numerical simulations on the PEZY-SC supercomputer, the critical role of the drift term in enabling self-organization and energy inverse cascade at negative absolute temperatures in a two-dimensional point vortex system [12]. In 2019, Kazuya Matsumoto proposes performance tuning techniques for general matrix multiplication on the PEZY-SC2 many-core processor [13]. T Hishinuma et al. optimized the high-precision arithmetic library pzdq for Double Precision Matrix Multiplication (DD Rgemm) for the PEZY-SC2 multi-core processor [14]. Guo presents an optimization strategy for sparse matrix-vector multiplication on the novel MIMD architecture Pezy-SC3s [11].

Pawan Harish and P.J. Narayanan first implemented the BFS algorithm with CUDA [8]. Luo et al. found that the BFS performance on GPU at the time was inferior to similar works on CPU, they proposed a hierarchical queue management technique and a three-layer kernel scheduling strategy to enhance BFS performance on GPU, especially with large-scale and irregular graph data [15]. In the classic work “Enterprise”, the authors argued that high-performance BFS optimization work targeting GPUs should focus more on leveraging their memory hierarchy and deeply integrating their architecture. They optimized GPU thread scheduling by eliminating thread contention and redundant data in the frontier queue, proposing a strategy based on node degree grouping to balance workload to some extent and quantifying the impact of central node direction switching. This work ranked 45th in the Graph500 list and first in the Green-Graph500 list in November 2014 [4]. In 2016, several scholars from George Washington University proposed efficient concurrent execution strategies for BFS on GPU, achieving high-performance concurrent graph traversal on large-scale GPU clusters through shared frontier utilization, a GroupBy strategy based on node out-degree, and bitwise optimization strategies [5]. In 2019, the Xbfs work published at HPDC pointed out that, with the continuous improvement in GPU performance, frontier queue generation based on atomic operations is faster than mainstream prefix-sum-based methods. Additionally, they introduced the first asynchronous traversal to achieve multilevel node updates within a single-layer

iteration [3]. In 2023, a solution called wave argued that previous works had high computational overhead and suffered from over fitting limited data. They proposed making more appropriate direction selection decisions by extracting features from input graph data and computing central node probabilities through sampling vertices from the input graph [16].

Most of the aforementioned works are primarily based on CUDA for implementation on Nvidia GPU, lacking exploration on other development platforms and architectures. Therefore, this paper implements a high-performance breadth-first search algorithm on the PEZY-SC3s using PZCL.

3 Comparison and Analysis

The search strategy is a fundamental and critical determinant of the overall performance of a search algorithm. Selecting an appropriate search strategy directly impacts search efficiency and resource utilization, while also playing a crucial role in the system's stability and scalability. This paper implements multiple search strategies on the PEZY-SC3s platform and conducts repeated measurements across various datasets to accurately evaluate and analyze their performance. These results serve as essential references for constructing an optimized overall search algorithm to achieve the best outcomes in diverse application scenarios.

Strategy one is the most basic version, which integrates node scanning updates and frontier queue generation into one step using atomic instructions. The performance of this strategy should be poor because the platform lacks cache coherence, which may lead to redundant counting of frontier nodes. Strategy two is a top-down search method that uses threads as the basic parallel unit. Both strategy tow and strategy three are top-down approaches, but strategy three involves two parallel levels: PE and threads. From this perspective, strategy three shows significant advantages over strategy two. This advantage stems from the implementation of a certain degree of load balancing in strategy three, where threads accessing the same node's neighbors within the same processing unit reduce memory access latency and improve cache hit rates. Strategy four is a bottom-up search method that also uses threads as separate parallel units. In scenarios with a larger number of nodes, the reverse update strategy (strategy four) demonstrates superior performance. This is attributed to the early termination feature of the reverse update strategy, which allows the completion of parent node search tasks while avoiding unnecessary search overhead. We compared the time consumption of different strategies for scanning and updating different datasets at different levels and provided specific results. By analyzing their performance on various datasets in depth, we can determine which strategy exhibits the best efficiency and resource utilization on the PEZY-SC3s.

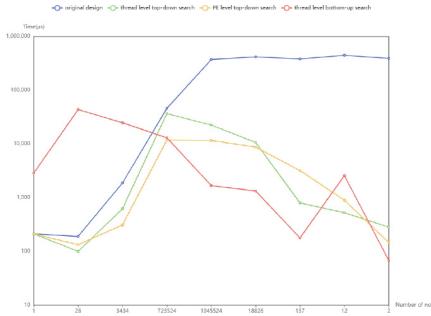


Fig. 6. Comparison results on wikipedia-topcats

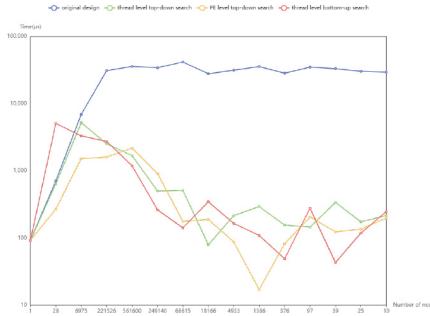


Fig. 7. Comparison results on com-youtube

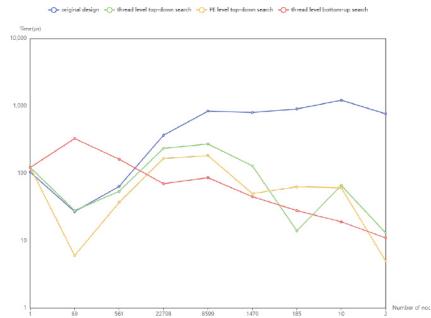


Fig. 8. Comparison results on Email-Enron

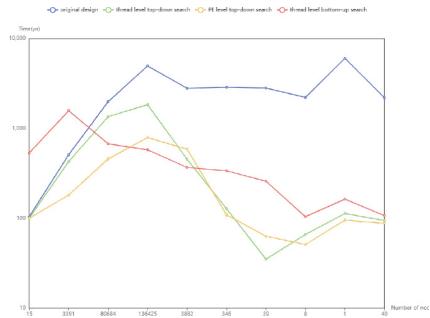


Fig. 9. Comparison results on Email-EuAll

When considering various search strategies for different datasets, we conducted five rounds of repeated executions for each level node and each search strategy, taking the average time consumption as reference data. Smaller time values indicate better performance of the strategy, the results are shown in Fig. 6, Fig. 7, Fig. 8 and Fig. 9. The actual performance of the four strategies closely aligns with our previous analysis and predictions, the overall performance of strategy three in the first half is better, while strategy four in the second half has better overall performance. Therefore, we consider strategy three and four to be the core components of the PEbfs proposed in this paper.

4 Design Section

This section consists of four subsections: algorithm design, load balancing, incomplete search and direction optimization. Each part will be explained in detail below.

4.1 Algorithm Design

The section includes four contents: direct update, scan update, reverse update and atomic counting.

Direct Update. As a level-order traversal algorithm, BFS starts the search from the root node, which is considered the 0th level node. Neighbor nodes of the root node are the 1st level nodes, all of which are directly accessible. During this stage, there is no need to scan the entire graph structure to obtain the required frontier nodes for the next level. Therefore, we refer to this stage as the direct update stage, which represents a top-down search strategy. As shown in Fig. 10, node 0 belongs to level 0, its neighboring nodes 1 and 7 were identified as unvisited. Consequently, the status of these neighboring nodes was updated to 1. The entire state array was then scanned, and nodes 1 and 7 were added to the front queue as frontier nodes. It is important to note that this scanning process is preparatory for subsequent operations.

Scan Update. This scheme consists of two parts: state information updating and frontier queue generation.

In the state information updating part, based on the frontier queue for the previous level, we use PE as the smallest parallel unit. With one PE processing one frontier node, each thread within a PE loads one neighboring node of the frontier node and updates the state information of the nodes that have not been traversed yet. This results in a dual-level parallelization of frontier nodes and their neighboring nodes. This approach represents another form of top-down search strategy.

In the frontier queue generation part, we no longer consider PE as the smallest parallel unit, but instead use thread as the smallest parallel unit, the scanning method used involves parallel operation of all threads. In each round, the total number of threads is used as the stride, simultaneously scanning the state information of each node corresponding to each thread. If the state information of a node is marked as visited and belongs to the previous level, it is written into the frontier queue using atomic instructions. After scanning all nodes, the generation of the frontier queue for this level is completed.

As shown in Fig. 11, frontier nodes 1 and 7 are scanned by a PE, and each neighboring node of the node is checked for status information and updated by a thread within the PE. The information of nodes 2 and 6 in the figure is updated by a thread in the same PE, while node 8 is updated by a thread in another PE. Then scan the entire state array to generate the frontier queue for the current level.

Reverse Update. In contrast to the common top-down breadth-first search strategy, there is also a bottom-up breadth-first search strategy. It also includes two parts: frontier queue generation and state information updating.

In the frontier queue generation part, similarly, all threads globally scan all nodes. However, in contrast to the previous strategy, this approach utilizes atomic operations to write all nodes that have not yet been traversed into the frontier queue, thereby completing the generation of the frontier queue for this level.

In the state information updating part, this strategy uses thread as the smallest parallelization unit, with one thread responsible for one node. The thread checks the state information of the neighboring nodes of the node. If the state information of a neighboring node has been updated, then this node is the child node of its neighboring node. At this point, the traversal of the neighboring nodes of the node can be stopped. Unlike the top-down search strategy where a parent node can have multiple child nodes, but a child node has only one parent node, in this strategy, once we have confirmed the parent node of the current node, there is no need to traverse subsequent neighboring nodes. This not only reduces the workload of the current node, but also optimizes the overall performance of the search strategy.

As shown in Fig. 12, To determine the nodes in level 3, the process begins by scanning the state array to identify all unvisited nodes and add them to the frontier queue. Each thread then examines the neighboring nodes of a given node. Known nodes 2, 6, and 8 are identified as level 2 nodes, a total of six neighboring nodes are connected to nodes 2, 6, or 8. This allows for the determination and update of level 3 nodes. Subsequently, the state array is scanned again, and unvisited nodes are added to the frontier queue.

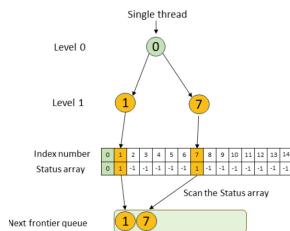


Fig. 10. Direct Update

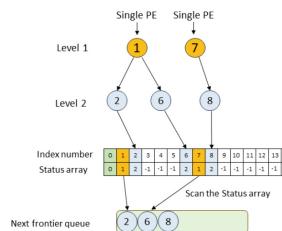


Fig. 11. Scan Update

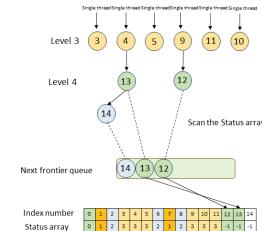


Fig. 12. Reverse Update

Atomic Counting. The generation of the frontier queue is a collaborative effort of multiple threads, each scanning its corresponding nodes and writing the qualifying ones to specified positions in the frontier queue. Ensuring thread safety and data consistency during concurrent execution is crucial in this context. Compared to traditional prefix-sum-based schemes that determine each thread's frontier position, existing work suggests that implementations based on atomic operations not only avoid inter-thread competition and simplify programming but also yield better results [3].

Given the superior atomic operation performance on PEZY-SC3s, the counting of frontier nodes during queue generation is implemented using atomic operations. The current node is written to the correct position in the frontier queue based on the return value of the atomic operation. PEZY-SC3s has a dedicated high-speed cache for atomic operations, which operates independently from the regular data cache. When using atomic operations, the cache should be flushed using the atomic flush instruction before reading the data. However, this rule doesn't always need to be strictly followed.

In this paper, we utilize a fraction of the atomic cache solely for implementing frontier queue counting, a section subject to repeated reading and writing. The platform supports atomic load instructions that can directly read data from specified addresses. Therefore, after completing the counting, we bypass the atomic flush operation and directly read the required data from the atomic cache. This approach avoids the significant overhead of flushing the cache and enhances program stability.

4.2 Load Balancing

We have three search strategies to adapt to different situations, but if the allocation strategy results in an imbalanced workload, it can significantly undermine the benefits from the existing optimizations. It is evident that the degree of a node is closely related to its workload. Balancing the workload among parallel units is a key factor in further improving algorithm performance.

In the top-down search strategy, nodes have varying degrees, and each node's neighbors must be completely traversed to finish processing that node. This often results in unbalanced workload distribution. To mitigate this issue, we categorize frontier nodes into three groups based on their degrees: nodes with degrees between 0 and 8 are in the small group, between 8 and 64 are in the medium group, and those with degrees greater than 64 are in the large group. Thus, it achieves the transformation of one frontier queue into three frontier sub queues.

In this approach, each PE is considered the smallest parallel unit for node processing. For the small group, each frontier node requires at most one processing unit cycle to complete scanning. For the medium group, with node degrees between 8 and 64, we assume the degree and number of nodes follow a normal distribution, so each frontier node requires at most five processing unit cycles to complete scanning. For the large group, with node degrees greater than 64, these nodes have higher degrees and are not within a finite range. We use 64 as the benchmark for our calculation, with each frontier node requiring at least eight processing unit cycles to complete scanning. This allows us to establish a linear relationship for global PE task allocation.

$$PE_{small} + PE_{medium} + PE_{large} = Max_{pid} \quad (1)$$

$$\frac{Num_{small}}{PE_{small}} = \frac{5 \cdot Num_{medium}}{PE_{medium}} = \frac{8 \cdot Num_{large}}{PE_{large}} \quad (2)$$

In this context, Max_{pid} represents the total number of PE in the processor. PE_{small} denotes the number of PE assigned to execute the search tasks for the small group nodes. Similarly, PE_{medium} and PE_{large} represent the number of PE assigned to execute the search tasks for the medium and large group nodes, respectively. Num_{small} indicates the number of frontier nodes in the small group, Num_{medium} indicates the number of frontier nodes in the medium group, and Num_{large} indicates the number of frontier nodes in the large group. Therefore, based on the linear relationship between the data presented in Eq. 1 and 2, the number of PE required for different frontier queues can be determined. In a multithreaded environment, the overall task execution time is contingent upon the execution time of the last thread to complete the task, as shown in Fig. 13, the left side depicts the workload distribution among threads without employing optimization strategies, while the right side illustrates the workload distribution with optimization strategies implemented. Evidently, the adoption of optimization strategies leads to a notable reduction in the overall task execution time.

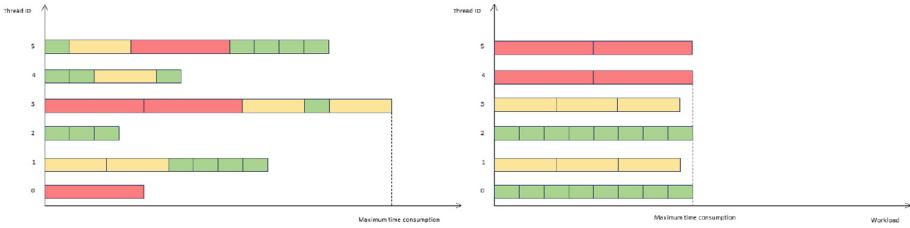


Fig. 13. The different colored boxes in the figure represent tasks with different workloads. The left side shows the load situation of different threads before load balancing, and the right side shows the load situation of different threads after completing load balancing.

4.3 Incomplete Search

When the search strategy switches to the bottom-up breadth-first search strategy, the first round of the frontier queue will include all unvisited nodes. With continuous iterative updates, the number of frontier nodes in subsequent levels will decrease. Each round's frontier queue will be a subset of the previous round's frontier queue. Therefore, except for the first round, which requires scanning all nodes to generate the frontier queue, subsequent rounds only need to scan the state information of the frontier nodes from the previous round. This approach significantly reduces redundant work and the time required to generate the frontier queue and avoids duplicate scanning of visited nodes, compared to scanning all nodes in each round, thereby further improving the algorithm's performance.

4.4 Direction Optimization

We previously mentioned two types of search strategies, each suitable for different problem states. The top-down breadth-first search strategy is ideal for scenarios where the current level's frontier queue has a small number of nodes, while the bottom-up breadth-first search strategy is better suited for scenarios where the current level's frontier queue has a large number of nodes. Therefore, it is essential to skillfully combine these two strategies, selecting the most appropriate one for each situation to leverage their strengths and avoid their weaknesses, thus optimizing the performance of the search strategy.

Previous work primarily relied on fixed threshold values derived from the node and edge information within their experimental datasets as switching criteria. However, datasets vary widely, and such values often overfit the experimental data, resulting in poor generalization. This paper proposes a new switching parameter estimation scheme that dynamically adjusts the threshold to the most suitable value.

When switching from scan-based updates to bottom-up updates, we use the ratio of the number of explored nodes to the total number of nodes, referred to as α . The switch occurs when α exceeds a certain threshold, known as the switching point. Unlike previous approaches, this method initially provides a range and a step size for the switching point, which can be user-defined. The switching point starts at the beginning of the range and increments by the step size in each iteration until it reaches the end of the range. This process generates multiple switching points, and by running the program for each value, the switching point that results in the fastest traversal speed is recorded as the optimal value for that dataset.

Figure 14 illustrates the overall design of PEbfs, which integrates the search update strategies, optimization strategies, and directional optimization strategy proposed in this paper.

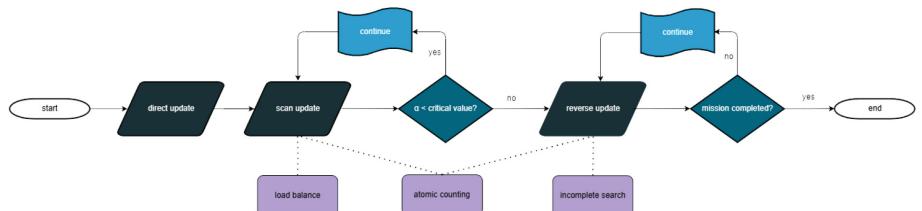


Fig. 14. The figure illustrates the overall structure of PEbfs, encompassing three search update strategies, two optimization strategies, and one directional switching strategy from left to right. Atomic counting is simultaneously applied to both search update strategies.

5 Experiment and Evaluation

5.1 Experimental Environment

In this paper, we implemented PEbfs based on PZCL. The experiments were conducted on a server equipped with an Intel Xeon Silver 4314 CPU (2.40GHz) and the PEZY-SC3s processor, running the CentOS 9 operating system. The theoretical bandwidth of the PEZY-SC3s is 600GB/s, with support for 4096 threads [11]. For comparison, the experiments with Enterprise and Tigr were conducted on Nvidia A16 GPU [17], hosted on a server equipped with an Intel Xeon Gold 5318Y CPU and running the Ubuntu 20.04 operating system. A16 GPU is based on Samsung 8nm process and Nvidia Ampere architecture, its theoretical bandwidth is 800 GB/s, containing 5120 CUDA cores. Compared with PEZY-SC3s, A16 has similar hardware parameters and has little advantage, making it a good reference platform.

The Table 1 describes several graph datasets evaluated in our experiments. All these datasets are sourced from the Stanford Large Network Dataset Collection [7]. They encompass various real-world social network information. Following conventional approaches, we converted these datasets into CSR (Compressed Sparse Row) format before use. The CSR structure can greatly reduce the storage space of data, which is very important for devices with limited memory [18].

Table 1. Detailed information of the dataset from SNAP

Datasets	Abbr.	Type	Nodes	Edges
as-skitter	SK	Undirected	1696415	11095298
Email-Enron	EN	Undirected	36692	183831
Email-EuAll	EU	Undirected	224832	790540
soc-Slashdot0922	SO22	Directed	82168	948494
soc-pokec	SOPK	Directed	1632803	30622564
soc-Slashdot0811	SO11	Directed	77360	905468
WikiTalk	TK	Directed	2394385	5021410
amazon0312	AMZ	Directed	400727	3200440
com-LiveJournal	LJ	Undirected	3997962	34681189
com-youtube	YTB	Undirected	1134890	2987624
wiki-topcats	CAT	Directed	1791489	28511807

5.2 Performance Evaluation of Optimization Strategies

Upon determining the search strategy, selecting an appropriate optimization strategy becomes essential for further performance enhancement. We adopt the load balancing technique to optimize the scan update processes, alongside incomplete search method to refine reverse update processes. Performance tests were

conducted on the YTB and EU datasets to evaluate the optimization strategy's effectiveness across different levels. The results, illustrated in the accompanying Fig. 15 and Fig. 16, indicate that solutions employing optimization strategies generally outperform those that do not. These optimization measures significantly enhance the overall efficiency of the search strategy, confirming their feasibility and effectiveness in practical applications. Furthermore, careful observation reveals that certain levels perform better with optimized scan update, while others excel with optimized reverse update. Different search strategies demonstrate varying performance across different scenarios. These observations will be analyzed and leveraged appropriately in the subsequent sections.

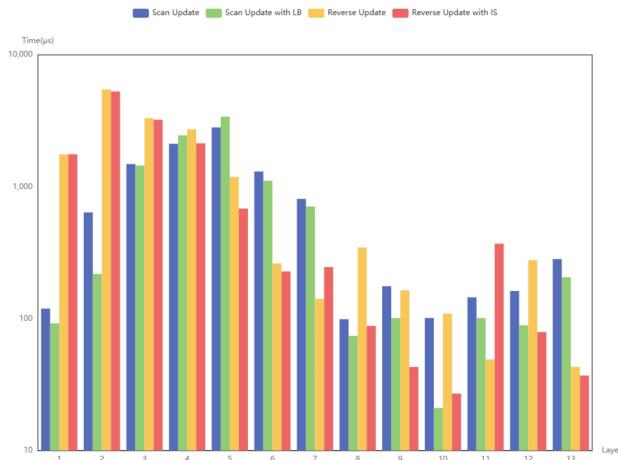


Fig. 15. Results of com-youtube, It includes two original strategies and two optimized strategies, and LB is load balancing, IS is incomplete search

After observing the performance variations of different strategies across various levels in the above experiments, we aimed to fully capitalize on the strengths of each strategy and we introduced a directional optimization strategy to switch search strategies at appropriate times, this directed optimization strategy establishes suitable switching conditions and transitions between search strategies at optimal times. We compared the two optimized strategies with the combined search approach incorporating the directional optimization. The results, shown in the accompanying Fig. 17, across all datasets, the combined search method incorporating directional optimization consistently outperforms the two individual search methods. This demonstrates the effectiveness of the directional optimization strategy. The combined search method, referred to as PEbfs, is thus validated as the superior approach proposed in this paper.

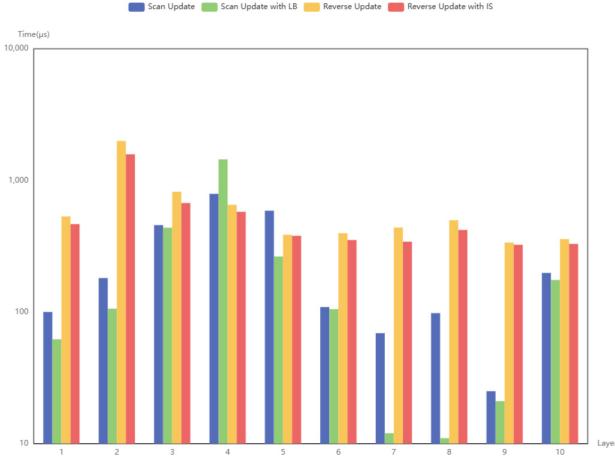


Fig. 16. Results of Email-EuAll, It includes two original strategies and two optimized strategies, and LB is load balancing, IS is incomplete search

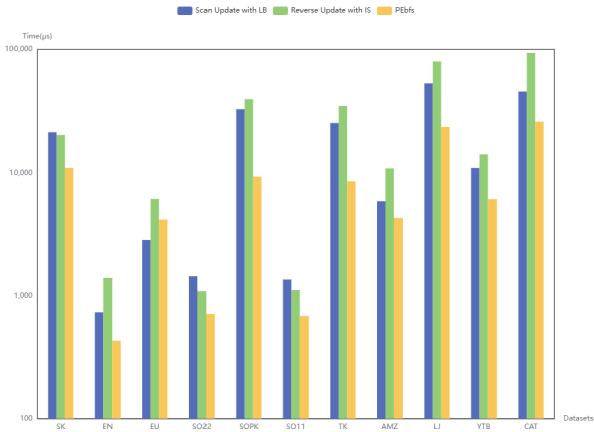


Fig. 17. Performance Comparison Results of Three Schemes on Eleven Datasets. The figure illustrates that the integrated search method, which includes directional optimization, consistently surpasses the performance of the two individual search methods.

5.3 Overall Performance

Currently, Graph500 encompasses three kernels: BFS, SSSP, and the BFS-based GreenGraph500 [18], which uses the MTEPS/W (Million TEPS per Watt) metric to measure energy efficiency [10]. Given the differences in architecture, specifications, and manufacturing processes between the experimental devices, We use the MTEPS/W to evaluate and compare the performance of the experiments.

Table 2. Experimental results on eleven datasets, MTEPS/W (million traversed edges per second per Watt) and Efficiency Ratio are two indicators for experimental comparison, it can be observed that in most cases, performance of PEbfs is better.

Datasets	Performance (MTEPS/W)			Efficiency Ratio	
	Enterprise	Tigr	PEbfs	PEbfs/Enterprise	PEbfs/Tigr
as-skitter	58.45	60.08	132.81	2.27	2.21
Email-Enron	11.21	35.62	54.66	4.88	1.53
Email-EuAll	6.65	22.04	33.55	5.04	1.52
soc-Slashdot0922	25.94	64.45	66.82	2.58	1.04
soc-pokec	177.94	75.21	337.00	1.90	4.48
soc-Slashdot0811	22.72	64.85	86.80	3.82	1.34
WikiTalk	32.56	449.22	102.57	3.15	0.23
amazon0312	42.66	24.53	131.88	3.10	5.38
com-LiveJournal	76.37	81.87	169.33	2.22	2.07
com-youtube	29.68	44.89	102.64	3.46	2.29
wiki-topcats	215.99	11.63	322.23	1.49	27.72
Average	63.65	84.94	140.03	3.08	4.53

The experimental results are shown in the Table 2, where PEbfs is compared with existing state-of-the-art solutions. Note that our PEZY-SC3s has fewer parallel processing units, non-coherent cache and lower read/write bandwidth than the NVIDIA A16. Despite these disadvantages, PEbfs outperforms the advanced BFS solutions in most cases. Compared to Enterprise, PEbfs shows an average performance improvement of approximately 3.08 times, with a maximum improvement of about 5.04 times. Compared to Tigr, PEbfs demonstrates an average performance improvement of about 4.53 times, with a maximum improvement of around 27.72 times.

6 Conclusion

In this work, we have developed the high-performance Breadth-First Search (BFS) algorithm, PEbfs, leveraging the high-efficiency PEZY-SC3s processor. PEbfs integrates three distinct search strategies, two optimization schemes, and one directional optimization approach. Despite the slightly inferior hardware parameters of the PEZY-SC3s compared to the Nvidia A16, our PEbfs exhibits significant performance gains. Specifically, compared to Enterprise, PEbfs achieves an average performance improvement of approximately 3.081 times, with a maximum improvement of about 5.042 times. Similarly, compared to Tigr, PEbfs demonstrates an average performance enhancement of about 4.527 times, reaching a peak improvement of around 27.717 times.

Acknowledgments. This study was funded by the National Key Research and Development Program of China (2023YFA1011704) and the National Key Research and Development Program of China (2021YFB0300101).

References

1. Hu, Y., Du, Y., Ustun, E., Zhang, Z.: GraphLily: accelerating graph linear algebra on HBM-equipped FPGAs. In: 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–9. IEEE (2021)
2. Hsieh, C.-Y., Cheng, P.-H., Chang, C.-M., Kuo, S.-Y.: A decentralized frontier queue for improving scalability of breadth-first-search on GPUs. In: 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1–6. IEEE (2023)
3. Gaihre, A., Wu, Z., Yao, F., Liu, H.: XBFS: eXploring runtime optimizations for breadth-first search on GPUs. In: Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing, pp. 121–131 (2019)
4. Liu, H., Huang, H.H.: Enterprise: breadth-first graph traversal on GPUs. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–12 (2015)
5. Liu, H., Huang, H.H., Hu, Y.: ibfs: concurrent breadth-first search on GPUs. In: Proceedings of the 2016 International Conference on Management of Data, pp. 403–416 (2016)
6. Nodehi Sabet, A.H., Qiu, J., Zhao, Z.: Tigr: transforming irregular graphs for GPU-friendly graph processing. ACM SIGPLAN Not. **53**(2), 622–636 (2018)
7. Leskovec, J., Krevl, A.: SNAP datasets: stanford large network dataset collection. <http://snap.stanford.edu/data>. Accessed 02 June 2024
8. Harish, P., Narayanan, P.J.: Accelerating large graph algorithms on the GPU using CUDA. In: Aluru, S., Parashar, M., Badrinath, R., Prasanna, V.K. (eds.) HiPC 2007. LNCS, vol. 4873, pp. 197–208. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77220-0_21
9. Hatta, N., Tsunoda, S., Uchida, K., Ishitani, T., Shioya, R., Ishii, K.: PEZY-SC3: a MIMD many-core processor for energy-efficient computing. arXiv preprint [arXiv:2301.07510](https://arxiv.org/abs/2301.07510) (2022)
10. TOP500: Green500 List - November 2018. <https://www.top500.org/lists/green500/>
11. Guo, J., Liu, J., Wang, Q., Zhu, X.: Optimizing CSR-based SpMV on a new MIMD architecture Pezy-SC3s. In: International Conference on Algorithms and Architectures for Parallel Processing, pp. 22–39. Springer, Heidelberg (2023). https://doi.org/10.1007/978-981-97-0801-7_2
12. Yatsuyanagi, Y., Hatori, T., Ebisuzaki, T.: Self-organizing effect of drift term against diffusion term in point vortex system evidenced by numerical simulations on PEZY-SC. Fluid Dyn. Res. **53**(3), 035510 (2021)
13. Matsumoto, K., Nakasato, N., Hishinuma, T.: Effectiveness of performance tuning techniques for general matrix multiplication on the PEZY-SC2. In: Proceedings of the 10th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, pp. 1–6 (2019)
14. Hishinuma, T., Nakata, M.: pzqd: PEZY-SC2 acceleration of double-double precision arithmetic library for high-precision BLAS. In: Okada, H., Atluri, S.N. (eds.) ICCEES 2019. MMS, vol. 75, pp. 717–736. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-27053-7_61

15. Luo, L., Wong, M., Hwu, W.: An effective GPU implementation of breadth-first search. In: Proceedings of the 47th Design Automation Conference, pp. 52–55 (2010)
16. Yoon, D., Jeong, M., Oh, S.: WAVE: designing a heuristics-based three-way breadth-first search on GPUs. *J. Supercomput.* **79**(6), 6889–6917 (2023)
17. NVIDIA: NVIDIA A16 GPU. <https://www.nvidia.cn/data-center/products/a16-gpu/>. Accessed 02 June 2024
18. Gan, X., Guo, P., Wu, G., Li, T.: GreenBFS: space-efficient BFS engine for power-aware graph processing. In: 2022 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 489–496. IEEE (2022)



Textual Data De-Privatization Scheme Based on Generative Adversarial Networks

Yanning Du¹ , Jinnan Xu¹ , Yaling Zhang¹ (✉) , Yichuan Wang^{1,2} , Zhoukai Wang¹

¹ School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China
y1zhang@xaut.edu.cn

² Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an, China

Abstract. In many fields, such as healthcare, finance, and scientific research, data sharing and collaboration are critical to achieving better outcomes. However, the sharing of personal data often involves privacy risks, so privacy-preserving techniques are needed to ensure data security and privacy. The superior performance and flexibility of generative models in data representation have led to significant progress in the development of data privacy. This paper proposes a textual data de-privatization scheme based on generative adversarial networks that well combines generative adversarial networks with privacy protection. The generated de-privacy data retains the statistical properties of the original data and effectively removes sensitive information. Moreover, feature sorting and reshaping modules are introduced to enable the generator to better capture the relationships between features, thus improving the quality of synthetic data. In this paper, the utility of synthetic data is evaluated from three aspects, including similarity assessment, privacy assessment, and model utility assessment. Experimental results show that this method achieves a good trade-off in terms of textual data privacy protection and data quality maintenance.

Keywords: Generative Adversarial Networks · Synthetic Data Generation · Privacy Protection

1 Introduction

With the proliferation of technologies such as the Internet, mobile devices, and the Internet of Things (IoT), large amounts of data are being generated and collected, the size and complexity of data are growing exponentially. This has led to greater concerns about personal privacy and the need for more effective technologies to ensure the security of personal data. In many fields, such as healthcare, doctors or researchers may need to share medical data for research, such as genomic data, medical history information, etc., in order to better understand diseases and develop new treatments [1]. In the financial sector, financial institutions may share customers' credit information to assess credit risk and determine credit terms [2]. Among these data, textual data are particularly prevalent and carry significant personal information. Data sharing and collaboration are essential to achieving better results. But the sharing of personal data often involves privacy risks,

privacy protection technologies are needed to ensure data security and privacy. And as public awareness of privacy protection increases, users have higher requirements for the privacy and control of personal data.

Anonymization is one of the most commonly used data protection measures. Samarati and Sweeney proposed the $k - \text{anonymity}$ principle [3]. After anonymizing the personal information database, at least k records in each equivalence class have the same combination of attributes other than privacy attributes. The larger the k value, the higher the degree of privacy protection, accompanied by an increase in information loss. However, k -anonymity is flawed because it does not provide explicit protection for sensitive data. $(\alpha, k) - \text{anonymity}$ [4] improves k -anonymity. While ensuring that k -anonymity is satisfied, each record associated with any attribute value in each published equivalence class does not exceed α . Thus, privacy protection is further strengthened. However, by merely anonymizing user information, an illegal attacker with access to this data can still identify sensitive information or personal relationships through protocol attacks or background knowledge attacks, resulting in privacy leakage.

Researchers have proposed synthetic data as an alternative to data transformation [5]. Synthetic data is artificially generated with statistical characteristics similar to the original data, but does not rely on the direct disclosure of real data. This approach provides stronger privacy protection because synthetic data will not contain real information from the original data. The process of generating synthetic data is usually based on models and algorithms that maintain similar statistical characteristics as the original data. Synthetic data has usability in many application scenarios. For example, the sensitivity and confidentiality of medical data make access to and sharing of real data limited, and the introduction of synthetic data can compensate for this. By using synthetic data, researchers can perform data analysis, model training and algorithm testing without accessing real sensitive data.

Based on the above analysis, this paper proposes a textual data de-privatization scheme based on generative adversarial networks. The main contributions of this work include: (1) This paper adds a privacy loss term to the original GAN loss function to measure the difference between the original data and the synthesized data, emphasizing sensitive information. This ensures that the generated data effectively protects sensitive information while maintaining the statistical properties of the original data, thereby enhancing privacy protection and data authenticity. (2) Feature sorting and reshaping modules are introduced in the generator to better capture the feature correlation in the dataset and improve the effectiveness of data synthesis. The feature sorting module enhances the robustness and consistency of data generation, while the feature reshaping module improves the quality and integrity of the synthesized data by better capturing feature relationships.

2 Related Work

2.1 Generating Adversarial Networks

Generative Adversarial Network (GAN) is a deep learning model proposed by Ian Goodfellow and his team in 2014 [6]. GAN consists of two main parts: a Generator and a Discriminator. It achieves learning by competing with each other through adversarial

training. Mehdi Mirza et al. proposed CGAN model [7]. CGAN extends the traditional GAN framework by adding conditional information (e.g., class labels) to both the Generator and the Discriminator to enable them to generate specific types of data. Arjovsky et al. proposed WGAN model [8]. WGAN improves the training stability of traditional GANs by replacing the binary classification task in the discriminator with the Wasserstein distance. The Wasserstein distance provides a more continuous and smoother gradient, which helps to alleviate the mode collapse and gradient vanishing problems of GANs. Gulrajani et al. proposed WGAN-GP model [9] by introducing gradient penalty on top of WGAN, which further improves the training stability and the quality of the generated samples of WGAN. Xu et al. proposed CTGAN model [10]. CTGAN specialized in generating unbalanced tabular data, the generator accepts inputs as a concatenation of noise vectors and conditional vectors, and controls the characteristics of the generated data through the conditional vectors. Zhao et al. designed CTAB-GAN model [11]. It can effectively model various data types, including a mixture of continuous and categorical variables.

2.2 Generative Adversarial Networks for Privacy Preservation

In recent years, significant progress has been made in the development of generative models for data privacy. Among them, Variable Auto-Encoder (VAE) [12] and GAN and their extensions are important directions of research because of their superior performance and flexibility in data representation. medGAN [13] successfully combines autoencoders and GANs to generate discrete attributes of records (e.g., binary and counting features) using pre-trained autoencoders to learn a compact representation of the input data. medGAN has been widely used to synthesize Electronic Health Record (EHR) data. medWGAN improves on medGAN by using a Wasserstein loss that is different from the standard GAN loss function. Although medGAN and medWGAN perform well on specific data types, they have limitations in generalizing to real-world scenarios as they only consider counting and binary features. GcGAN [14] was proposed by Yang et al. It generates realistic EHR by classifying and considering relationships between disease, treatment, and efficacy, using separate encoders and decoders for each variable group. PATE-GAN [15] further introduces the concept of differential privacy. It provides a solution for generating synthetic data with differential privacy, providing a higher level of protection for data privacy.

However, existing work does not integrate generative adversarial networks well with privacy protection, especially when dealing with textual data. Existing generative adversarial networks struggle to strike a good balance between the authenticity and privacy of synthesized data. In this paper, we propose a textual data de-privatization scheme based on generative adversarial networks. This model not only generates data that retains the statistical properties of the original data, but also effectively removes sensitive information from the data.

3 Model Design

The overall architecture of the model in this paper is shown in Fig. 1. It contains a total of three modules: generator (G), discriminator (D), and classifier (C). The core goal of GAN is to generate synthetic data that is highly similar to the original data through the generator. Through the interaction of the generators and the discriminators, the GAN is able to learn the distributional properties of the original data and generate synthetic data that can deceive the discriminators. During the training process, the classifier provides additional feedback to help ensure that the generated synthetic data maintains semantic integrity and structural consistency with the original data. The specific design of each component is as follows.

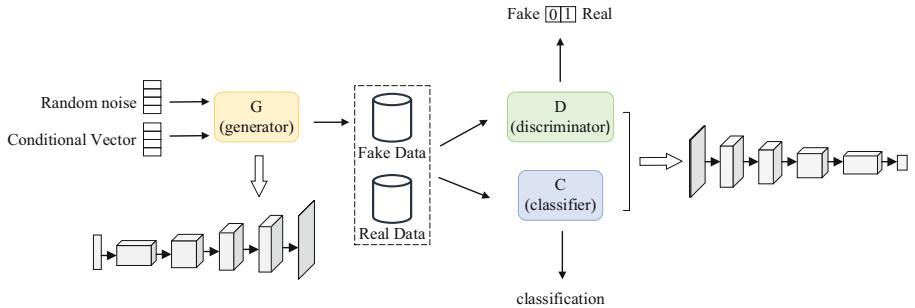


Fig. 1. Overall model architecture

3.1 Generator

The goal of the generator is to produce synthetic data with a joint distribution similar to the original data. The real data distribution is $x \sim P_{data}$, and the generator receives as input the latent noise vector extracted from an arbitrary noise distribution $z \sim P_z$, and the conditional vector extracted from the probability distribution of the conditional vectors $c \sim P_c$, and learns to map them to the data space χ . The generated synthetic data is defined as \hat{x} .

$$\hat{x} \leftarrow G(x, c)$$

The main structure of the generator is shown in Fig. 2. In this paper, the input of the generator is a potential noise vector z . Then the input data is processed by the feature sorting and reshaping module. The linear transformation layer performs linear mapping on the vectors after feature sorting and reshaping. The feature map is then gradually converted into a two-dimensional matrix through the deconvolution layer, the size of which corresponds to the generated synthetic data. The activation function layer and batch normalization layer perform nonlinear processing and batch normalization operations, and the final output layer outputs the generated two-dimensional matrix.

Only the output layer of the generator uses the Tanh function as the activation function, and the other layers use the Leaky Relu function.

This study incorporates a feature sorting and reshaping module into the generator to better capture the correlation of characteristics across different datasets. The correlation matrix between features is first calculated on the input data. The correlation matrix is sorted to determine which features are most relevant, and a feature index list is generated based on the sorting results. The reshaping module sorts the features of the input data using the feature index list generated by the feature sorting module and rearranges the sorted feature vectors into a square matrix. The above modules are added to enable the model to capture correlations between features more efficiently and to reduce boundary effects, resulting in a higher quality and usefulness of the generated data.

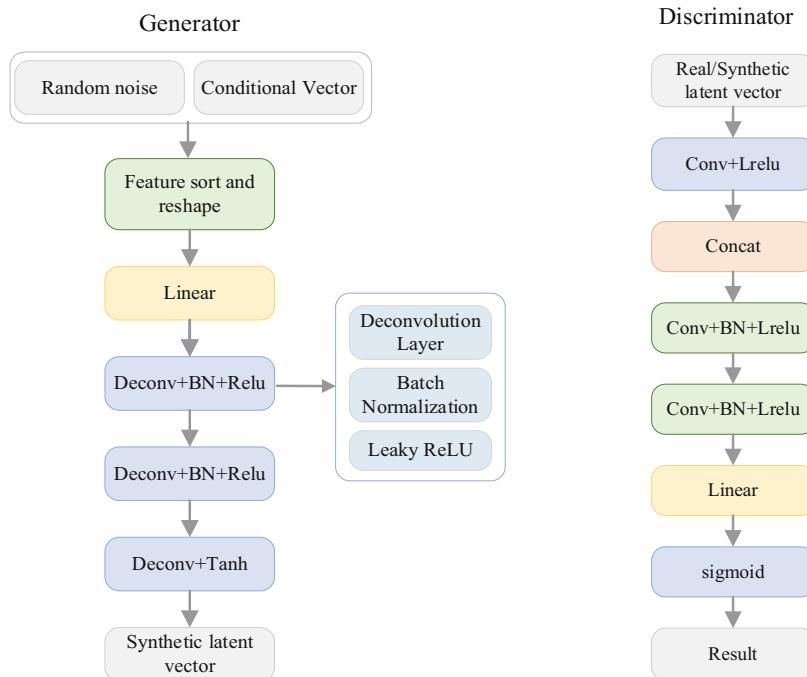


Fig. 2. Generator and discriminator structure

3.2 Discriminator

The task of the discriminator is to classify the input data as real or synthetic. In this study, CNN is used as the discriminator to forecast the validity of the data and separate it from the generator's synthetic data. The main structure of the discriminator is shown in Fig. 2. The inputs to the discriminator are the real data distribution and the synthetic data distribution, both of which are processed and classified by the discriminator. The final outputs are classified as “true” or “false”.

The discriminator consists of multiple convolutional layers, which are used to extract key features from the input data. After each convolutional layer, LeakyReLU is used as the activation function, which allows small negative gradients and helps to speed up the training process and enhance the stability of the model. With these convolutional layers and activation functions, the discriminator is able to effectively extract and classify features from the input data to accurately predict the authenticity of the data.

3.3 Classifier

The classifier in this article is also implemented using a CNN. The classifier is trained on the original data and learns the correlation between the attributes of the original data to better interpret the semantic completeness. For a given synthetic record, the classifier can determine whether the record is semantically correct and teach the generator to generate more accurate synthetic data, which plays a role in detecting semantic integrity.

4 Loss Function Design

The loss function in this paper contains a total of three components: discriminator loss, classifier loss, and generator loss. The discriminator is trained using the original loss function of the GAN. The classifier is trained using the classifier loss. The generator is trained using the original loss function, classifier loss, and privacy loss of the GAN.

4.1 Original Function

The learning process of the GAN model is based on G and D neural networks and performs minimax (i.e., adversarial) training. The initial loss function of GAN is as follows:

$$\min_D \max_G V(D, G) = E_{x \sim p_{data}} \log D(x) + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (1)$$

$G(\cdot)$ represents the generator function, $D(\cdot)$ represents the discriminator function. D tries to maximize $V(D, G)$, while G tries to minimize $\max_G V(D, G)$. The two are trained against each other so that the generator produces more realistic data and the discriminator more accurately distinguishes between true and false data.

4.2 Classifier Function

The classifier loss function is used to evaluate the difference between the label of a synthetic record and the label predicted by the classifier for that record.

$$L_{Cg} = E[|\ell(G(z)) - \mathcal{C}(G(z))|]_{z \sim P_z} \quad (2)$$

$$L_{Cd} = E[|\ell(x) - \mathcal{C}(x)|]_{x \sim P_{data}} \quad (3)$$

$\ell(\cdot)$ represents the labelling information of the original data. $\mathcal{C}(\cdot)$ represents the label information predicted by the classifier. L_{Cg} is used to measure the difference between synthetic data labelled values and predicted labelled values. L_{Cd} is used to measure the difference between real data labelled values and predicted labelled values. By adding the classifier loss the semantic integrity and structural consistency of synthetic data can be better ensured and the data accuracy of synthetic data can be improved.

4.3 Privacy Function

Let f_x and f_z represent the characteristics of the original data and synthetic data input to the discriminator respectively. Use weighted Euclidean distance to measure the difference between the original data and synthetic data.

$$L_p = E \| w \cdot ([f_x]_{x \sim p_{data}} - [f_z]_{z \sim p_z}) \| \quad (4)$$

The weight of the weighted Euclidean distance is $w = 1/H(f_x)$. Using the inverse of discrete entropy as the weight can capture the differences between features more effectively, making sensitive and critical features more distinguishable. For sensitive and critical features, the model will focus more on their protection.

If two sets of data differ significantly on a key feature, even if they are similar on other features, the distance between them will increase accordingly. Since this feature has a large weight, it is necessary to strengthen the protection of this sensitive feature. In contrast, if there is a large difference between two data on a feature with a lower weight. Because the weight corresponding to the feature is small, the contribution of this difference to the overall distance will then be relatively small. This means that although the two data points differ significantly on this feature, the overall distance between them may not be particularly large due to other features. The discrete entropy reflects the uncertainty of the feature. When the discrete entropy of a feature is smaller, its weight will be larger, and this feature has higher discriminability and sensitivity. By increasing the weight of these sensitive features, the model will focus more on protecting these features, thus improving privacy protection for sensitive data.

4.4 Model Training

In the model training part, this paper adopts an integrated strategy that includes the initialization of D , C and G . The training process mainly involves the following steps: firstly, drawing a small batch of samples through the real dataset \mathcal{D} and the noise distribution P_z ; secondly, updating each model by training the loss functions of D , C and G . The specific process is shown in Algorithm 1.

Algorithm 1. Training algorithm of model

Model Training

Initialize: θ_d for D , θ_g for G , θ_c for C

Input: Real dataset \mathcal{D} , $z \sim P_z$

while training loss has not converged **do**

 Draw mini-batch samples from real data \mathcal{D} , $\{x_k\}_{k=1}^n$

 Draw mini-batch samples from noise distribution P_z , $\{z_k\}_{k=1}^n$

for $k = 1, \dots, n$ **do**

$$\hat{x}_k \leftarrow G(x_k, c)$$

end for

 Update Discriminator parameters θ_d

$$\tilde{V} = \frac{1}{n} \sum_{k=1}^n \log D(x_k) + \frac{1}{n} \sum_{k=1}^n \log(1 - D(\hat{x}_k))$$

$$\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$$

 Update Classifier parameters θ_c

$$\tilde{V} = \frac{1}{n} \sum_{k=1}^n |\ell(x_k) - \mathcal{C}(x_k)|$$

$$\theta_c \leftarrow \theta_c - \eta \nabla \tilde{V}(\theta_c)$$

 Update Generator parameters θ_g

$$\tilde{V} = \frac{1}{n} \sum_{k=1}^n [\log D(\hat{x}_k) - |\ell(x_k) - \mathcal{C}(\hat{x}_k)| + \lambda L_p(x_k, \hat{x}_k)]$$

$$\theta_g \leftarrow \theta_g + \eta \nabla \tilde{V}(\theta_g)$$

end while

5 Experiments

The experiment in this article consists of three parts: (1) Evaluating the similarity between the original data set and the synthetic data set; (2) Evaluating the privacy of the synthetic data set; (3) Evaluating the effectiveness of the model. This article compares the proposed model with Pategan, Ctgan and Ctabgan.

5.1 Dataset Description

In this paper, the original dataset is divided into two disjoint subsets: the training set and the test set. 80% of the data is randomly selected for the training set and the rest of the data is used to form the test set. The synthetic dataset is also randomly divided into training and test set using the same 80% training and 20% test division. In this paper, the generated synthetic dataset is evaluated on the basis of various metrics of the test set.

The Adult dataset is chosen as the dataset to be de-privatized. The Adult dataset, also known as the “Census Income” dataset, is extracted from the U.S. Census database and contains a total of 4884 records. This dataset contains 14 key attribute variables, eight

of which are discrete categorical variables and six of which are continuous numerical variables.

5.2 Similarity Assessment

In this paper, we compare the overall distribution characteristics of the original data as well as the synthetic data using the Cumulative Distribution Function (CDF). The CDF is an integral of the probability density function that completely describes the probability distribution of a real random variable. By comparing the CDF of the original data and the synthetic data, the similarity between the synthetic data and the original data in terms of the overall distribution can be observed intuitively.

Figure 3 (a–d) shows the images of the cumulative distribution function of the model proposed in this paper, Pategan, Ctgan and Ctabgan for the education as well as capital gains in the dataset, respectively. The blue line in the figure represents the actual values in the original data and the orange line represents the values in the synthetic data. Among the four methods, the cumulative distribution of capital gains for the synthetic data generated by Pategan is less similar to the original data, and the others are statistically similar to the original data. However, the figure shows that the synthetic performance of the model in this paper is better compared to the other models.

5.3 Privacy Assessment

5.3.1 Distance Based Metric

The distance based metric uses Euclidean distance as a performance metric. The formula for Euclidean distance is as follows:

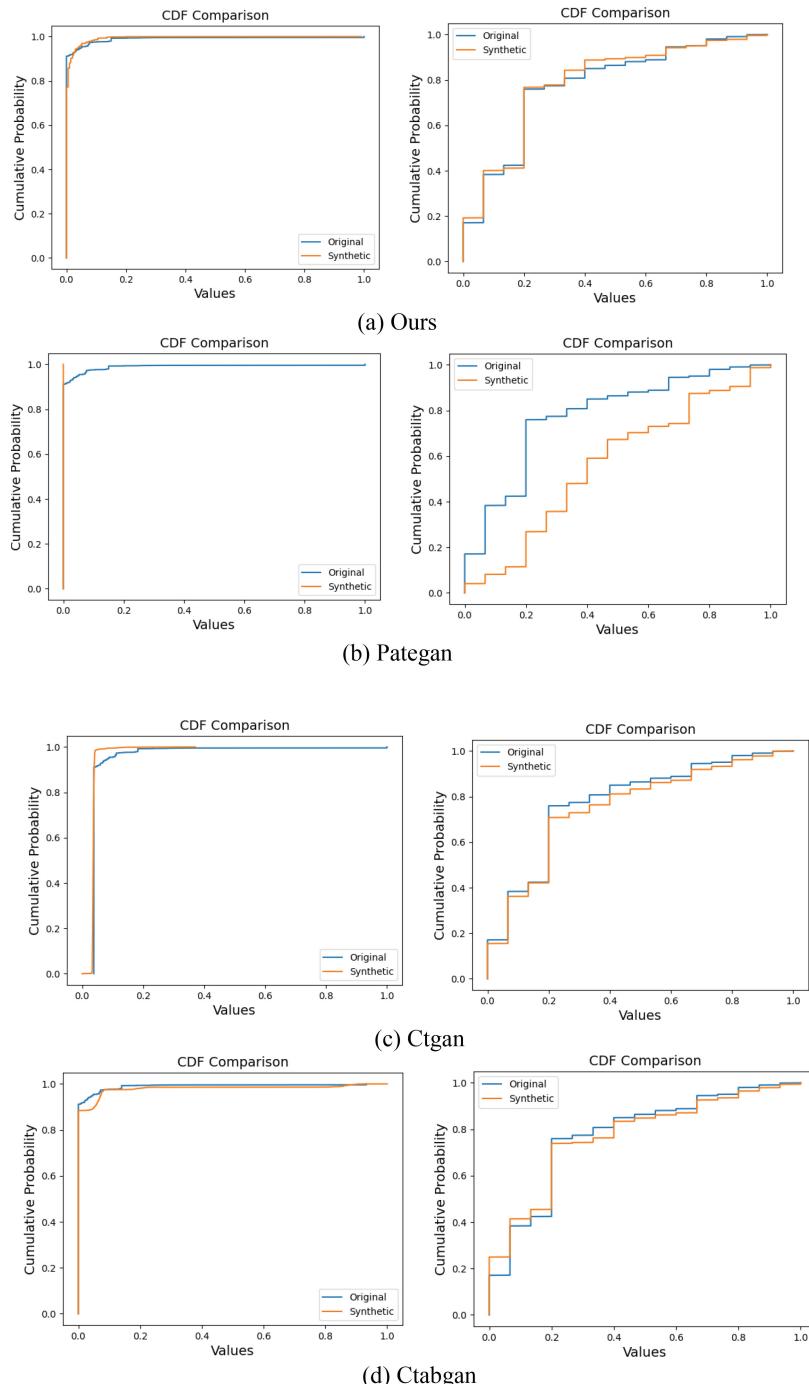
$$D_{Euclidean} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

x_i and y_i respectively represent the value of the i-th feature in the two records. n represents the number of features and indicates the degree of difference between two records in all features. The distance-based metric is used to measure the minimum Euclidean distance between records of original data and synthetic data. If the distance between the records of original data and synthetic data is too large, it means that the quality of the synthetic data is poor. If the distance is too small, then it means that the data has the risk of leaking sensitive information. The distance measure is 0 means that the synthetic data leaks the information of the original data.

5.3.2 Hit-Based Metric

The hit-based metric is used to find the proportion of similar records in the original data as well as in the synthetic data. Firstly, 1000 records are randomly selected from the synthetic data and then for each sampled record, similar records are found in the original data. The conditions for determining similarity include:

- (1) The values of all classification attributes are exactly the same.

**Fig. 3.** Similarity assessment

- (2) The values of numerical attributes are within the set threshold. The threshold is calculated by dividing the attribute range by 20.

As shown in Table 1, Ctabgan has the smallest distance measure between the original data and the synthetic data, and has the highest hit rate. This shows that Ctabgan has the risk of privacy leakage. Pategan has the largest distance measure between the original data and the synthetic data, and has a hit rate of 0. This shows that Pategan is better at privacy protection. However, the distance metric of the model proposed in this paper is second only to Pategan, and the hit rate is also 0, which cannot identify sensitive information from synthetic data, and the statistical similarity of the model is better than Pategan.

Table 1. Privacy assessment.

Method	Distance Measure	Hitting Rate
Our method	0.081	0
Pategan	0.473	0
Ctgan	0.064	0.005
Ctabgan	0.029	0.017

5.4 Model Utility Assessment

In this section, the model utility is evaluated in terms of F1 score, accuracy and AUC using logistic regression model, linear support vector machine (SVM), and multilayer perceptron (MLP). This is used to assess the closeness of the model utility of machine learning models when trained on synthetic and original data.

5.4.1 F1 Score

The horizontal axis of the F1 score plot represents the F1 score on the original dataset and the vertical axis represents the F1 score on the synthetic dataset. Each point on the scatterplot represents the performance of the model on the original and synthetic data under different parameter settings. The dashed gray line represents the ideal diagonal, when performance on the original and synthetic datasets is equal.

As shown in (a-d) in Fig. 4. The figure shows the F1 score plots for the model proposed in this paper, Pategan, Ctgan and Ctabgan. The closer the points on the scatter plot are to the diagonal line, the better the performance of the model on the synthetic data, which is closer to the performance on the original data. Observing the F1 score plots of these three models, most of the scatter points of the model proposed in this paper are concentrated near the diagonal line, showing better model utility. The distribution of Pategan scatter points on the diagonal is relatively small, indicating that its performance is relatively poor.

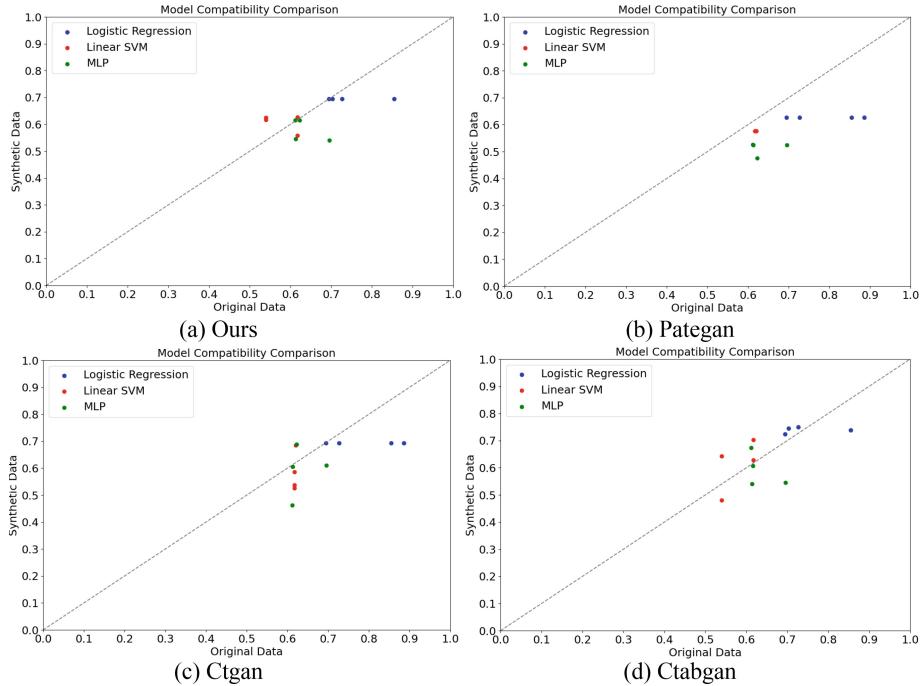


Fig. 4. F1 score results

5.4.2 Accuracy Score

Accuracy refers to the proportion of samples correctly predicted by the model to the total number of samples. For both synthetic and original data, accuracy shows how well the model performs in terms of classification. By comparing the average accuracy difference between original data and synthetic data, this research assesses the model's generalizability for synthetic data.

5.4.3 AUC Score

The AUC score is the area under the ROC curve, which indicates the classification performance of the model for real data and synthetic data. The performance of the model is better when the AUC value is closer to 1; the model performs poorly when it is closer to 0.5. By comparing the average AUC difference between the original data and the synthetic data, this research assesses the model's generalizability for synthetic data.

Table 2 illustrates that our model performs well in data categorization since it has the lowest average accuracy difference and average AUC difference. When it comes to creating high-quality synthetic data and training models using it, our model works effectively. Pategan, Ctgan, and Ctabgan have somewhat better accuracy and AUC than our model. Therefore, the present model can be more reliably used as a replacement or supplement to the original dataset for model training and evaluation.

Table 2. Model utility evaluation

Method	Accuracy	AUC
Our method	0.042	0.059
Pategan	0.187	0.158
Ctgan	0.060	0.161
Ctabgan	0.043	0.072

6 Conclusion

This paper proposes a text data de-privacy scheme based on generative adversarial networks, which better combines generative adversarial networks with privacy protection and achieves a good balance between the authenticity and privacy of synthetic data. First, the overall design of the model is carried out based on existing research; secondly, the structure of each part of the model and the loss function are analyzed in detail; finally, the effectiveness of the model is evaluated from three aspects: similarity, privacy, and model utility. In future work, we will further optimize the performance and stability of the model and explore more application scenarios and data types.

Acknowledgments. This research work is supported by the National Natural Science Founds of China (62072368, 62302389), Key Research and Development Program of Shaanxi Province (2022CGKC-09), and Natural Science Basic Research Program of Shaanxi Province (2023-JC-QN-0742).

References

1. Hulsen, T.: Sharing is caring—data sharing initiatives in healthcare. *Int. J. Environ. Res. Public Health* **17**(9), 3046 (2020)
2. Bank, T., Segev, N., Shaton, M.: Relationship banking and credit scores: evidence from a natural experiment (2023). SSRN 4567568
3. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression (1998)
4. Wong, R.C.W., Li, J., Fu, A.W.C., Wang, K.: (α, k) -anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 754–759 (2006)
5. Bellovin, S.M., Dutta, P.K., Reitinger, N.: Privacy and synthetic datasets. *Stanf. Technol. Law Rev.* **22**, 1 (2019)
6. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in neural Information Processing Systems, vol. 27 (2014)
7. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434) (2015)
8. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR (2017)
9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

10. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
11. Zhao, Z., Kunar, A., Birke, R., Chen, L.Y.: CTAB-GAN: effective table data synthesizing. In: Asian Conference on Machine Learning, pp. 97–112. PMLR (2021)
12. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
13. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. In: Machine Learning for Healthcare Conference, pp. 286–305. PMLR (2017)
14. Yang, F., Yu, Z., et al.: Grouped correlational generative adversarial networks for discrete electronic health records. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 906–913. IEEE (2019)
15. Jordon, J., Yoon, J., Van Der Schaar, M.: PATE-GAN: generating synthetic data with differential privacy guarantees. In: International Conference on Learning Representations (2018)



Modeling and Simulation Verification of Operating Mode Switching of Train Control System Based on Train-to-Train Communication

Qiang Li^{1,2(✉)}, Ian Liao³, Sheng Wen², and Yang Xiang²

¹ Lanzhou Jiaotong University, Lanzhou, China

QiangLee402@gmail.com

² Swinburne University of Technology, Melbourne, Australia

³ Top-Rail Service, Pittsburg, USA

Abstract. The train control system needs to accommodate various operational scenarios through the design of distinct working modes, each triggered by changing input conditions. These modes correspond to specific speed limits and safety requirements, ensuring the safety, efficiency, and continuity of train operations. The future of next-generation urban rail transit lies in train control systems based on train-to-train communication. Due to structural changes, traditional CBTC (Communication-Based Train Control) model designs are no longer applicable, and there is currently no comprehensive model design available. This paper examines the mode-switching function and processes appropriate for train-to-train communication control systems. A new mode definition and switching scheme is proposed. In the event of train-to-train communication failure, the train switches to a backup operation mode using train-to-ground communication to acquire position information from ground equipment. To describe this transition, we use the theory of Colored Petri Nets to establish a mode-switching model based on Hierarchical Timed Colored Petri Nets (HTCPN). The model effectively transitions trains from train-to-train mode (T2T) to train-to-ground mode (T2G) under failure conditions. We studied the impact of different operational intervals in the backup mode on system performance through simulations, which show that trains can effectively switch to train-to-ground mode during communication failures. Analysis of successful mode switches and switching times under different backup mode intervals reveals that the success rate increases with longer intervals. An interval of 240 s minimizes train operation delays to within 3 min. The HTCPN model established serves as a reference for verifying and analyzing other multi-mode train control systems.

Keywords: Train-to-Train · Train Control System · Petri Nets · Mode

1 Introduction

The train control system is the core component of urban rail transit dispatch and control, ensuring operational safety and efficiency. Currently, the predominant system for urban rail transit train control is the Communication-Based Train Control (CBTC) system, which utilizes WLAN (Wireless Local Area Network) technology to enable bidirectional communication between trains and ground equipment. This technology offers high transmission rates and large volumes of data, making it suitable for trains of varying speeds, capacities, and traction types [9]. However, the CBTC system's complexity poses several challenges. It requires extensive ground and trackside equipment such as Zone Controllers (ZC), transponders, and track circuits, leading to prolonged construction periods, increased construction costs, and maintenance difficulties. The interfaces between various system components are complex, necessitating extensive information exchanges, which complicates interconnection and interoperability. Additionally, the CBTC system utilizes a “train-ground-train” communication mode and control structure, preventing direct information interactions between leading and trailing trains. Instead, all trains relay their positions, speeds, and other data to the ground-based ZC, which then calculates movement authorization for each train under its jurisdiction. A failure of the ZC can impact all trains within its jurisdiction, potentially causing widespread delays.

To address these issues and ensure the efficient and safe operation of urban rail transit, a new type of train control system based on train-to-train communication has been developed. This system simplifies the structure of the traditional CBTC train control system by integrating the functions of some ground and trackside equipment into onboard equipment. This integration significantly reduces the need for ground equipment, alleviates the system's dependence on it, and enhances overall system integration, resulting in substantial savings in construction and maintenance costs [11]. The system eliminates the limitations of the traditional “train-ground-train” communication mode by utilizing LTE (Long Term Evolution) in Device to Device (D2D) technology, which enables direct communication between trains. This direct communication allows for faster information transmission and higher operational efficiency [12]. However, as this is a new type of train control system, the design of its structure and functional logic must first be carefully planned. Following this, formal modeling and validation must be conducted to ensure that the train control system meets “fault-safety” requirements through thorough demand analysis.

This paper analyzes the structure and functionality of the train-to-train communication control system, defines various working modes and switching mechanisms, and proposes a train-to-ground communication mode as the backup for the new train control system. We establish a mode-switching model using Colored Petri Nets and analyze the system switching times within this model. The impacts of different design time intervals on train operations under the backup mode are also examined.

2 Related Work

As a novel train control system, vehicle-to-vehicle communication technology has been extensively studied both domestically and internationally. Research encompasses various aspects, including the implementation of communication technology and the structural-functional modeling of the system.

Regarding the feasibility of a new type of train control system, the European Railway Union [4] proposed the Next Generation Train Control (NGTC) system. This system addresses issues associated with existing trackside equipment and maintenance difficulties, and proposes an all-IP communication network to provide a comprehensive standardized solution for mainline railroads and urban rail transit. Wang et al. [5] introduced the ERTMS-Regional system proposed by the UIC of the International Union of Railways for the European ETCS-3 level system. This system eliminates ground signaling machines and station track circuits, streamlines trackside equipment, and achieves approach functions via central equipment and object controllers. Train tracking safety is ensured by the transponder and the RBC. Zhao et al. [8] introduced the Positive Train Control (PTC) system in the United States, which integrates train operation control, central dispatching command, and communication automation based on the existing train control system. The movement authorization for the train is calculated by the PTC central server and transmitted to the train via wireless communication. Chen et al. [10] proposed a direct communication scheme between two trains using ultra-short wave technology, verifying the feasibility and safety of direct communication between trains.

Regarding the modeling and validation of the system, Chen et al. [7] employed timed automata theory to model typical operational scenarios of the train control system based on train-to-train communication. Their study considered the efficiency of turnback under both moving block and fixed block systems, validating that the new system can meet the requirements for these specific scenarios. Zhang et al. [15] proposed a train-centered approach control method that relies on onboard equipment to control the trackside object controller, thereby managing the occupancy and release of approach resources. By employing the theory of Colored Petri Nets, they established an approach resource requisition model that enhances the efficiency of approach resource requisition for trains, providing valuable references for the approach control methods in vehicle-to-vehicle communication train control systems. Chen et al. [14] introduced a stochastic Petri net-based reliability evaluation method for train-to-train communication. They developed SPN reliability models for train-to-train communication in different modes, demonstrating that the D2D technology in LTE-R wireless communication systems can effectively improve the service quality of train-to-train communication.

It is evident that the new train control system based on train-to-train communication has become a significant area of research, with extensive studies conducted on communication quality, safety analysis, and specific functional implementations. However, there remains a gap in the functional design of emergency handling logic for the new train control system under various abnormal situa-

tions. A comprehensive train control system must define corresponding working modes based on real-time information. Each mode should align with specific train control logics and safety speed limits, and the system should be capable of mode transitions when input conditions change. This approach ensures the continuity, efficiency, and safety of train operations, which is crucial for the effective operation of the train control system.

3 System Definition

3.1 System Structure

The train control system based on train-to-train communication eliminates the need for Zone Controllers (ZC) and Interlocking Control equipment on the ground by integrating the functions of these original ground devices into onboard equipment. This integration creates an onboard control structure as the core, enhancing the autonomy of the train. The structure of the train control system based on train-to-train communication, as shown in Fig. 1, includes three main components: central equipment, vehicle equipment, and trackside equipment [6]. The central equipment comprises the Intelligent Train Supervision (ITS) and the Train Information Management Center (TMC). The ITS is responsible for formulating and issuing train plans and supervising train operations, akin to the ATS function in the CBTC system. The TMC acts as a database for storing information; all trains must register their details in the TMC upon going online, including train numbers, IP addresses, and locations. The TMC also provides information about other trains within its jurisdiction and serves as a reporting hub for rescue trains and non-communication trains entering the line. Onboard equipment includes modules such as the onboard interlocking module, onboard communication module, and onboard movement authorization module. The onboard interlocking module controls approach commands and issues control instructions for trackside equipment. The onboard communication module incorporates both train-to-ground and train-to-train communication modules, overseeing communication between the train and ground, and between trains, respectively. The onboard movement authorization module is responsible for granting movement authorization to the vehicle. Trackside equipment primarily consists of transponders and Object Controllers (OC), which gather and report the status of the turnout for ITS and train queries, and execute commands sent by the train.

After departure from the depot and completion of a power-on self-test, the train registers with the ground equipment TMC. The TMC then sends line information, an electronic map, and other static information to the train, aiding in the electronic map proofreading process. The ITS sends the train its running plan. Subsequently, the train queries the TMC for information about other trains and matches the electronic map in the onboard module to verify the occupancy of the station's virtual sections. Upon receiving the position information of other trains in the station from the TMC, the onboard module matches this data with the electronic map to check the occupancy of the virtual sections within the

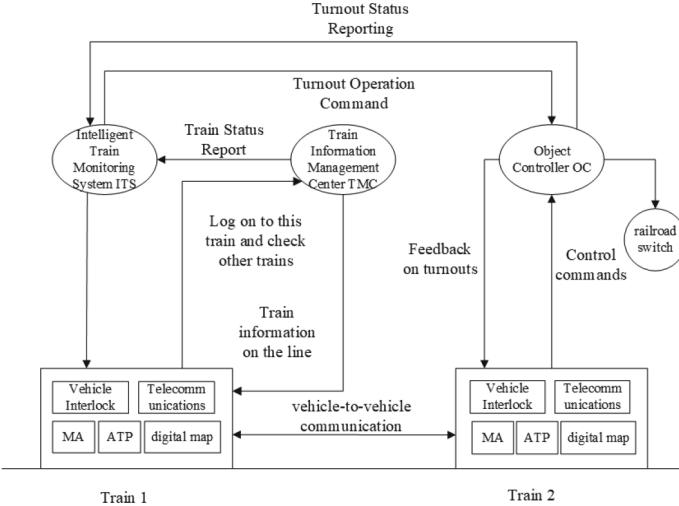


Fig. 1. Schematic diagram of the train-to-train communication system

station. Simultaneously, the train queries switch status information from the OC, requisitioning the required switch according to the approach table. It then performs the switching and locking of the switch, completing the occupancy of the approach. After the route selection in the station is completed, the train uses the position information of other trains obtained from the TMC to filter out those within its approach range. It establishes train-to-train communication with these trains, identifying and determining the unique lead vehicle. The train maintains communication with this unique lead train and terminates communication with others. The onboard movement authorization module then generates real-time movement authorization, which is sent to the onboard ATP, ensuring the safety of train operations. In the event of a train-to-train communication failure, the train reverts to backup mode operation.

3.2 Working Mode Definition

When a train is running normally within a zone, it must maintain communication with the lead train to obtain its position information. Simultaneously, it communicates with the ground TMC to cyclically update its own position information. Hence, the system operates in two modes: Train-to-Train (T2T) and Train-to-Ground (T2G) [12]. Train driving modes are mainly categorized into Automatic Driving mode (AM/AM-G), Fully Monitored mode (SM/SM-G), and Restricted Manual driving mode (RM). In scenarios where the train-to-train communication module in the onboard communication equipment fails, the train cannot acquire position information from the lead train. However, it can still communicate with the TMC through the train-to-ground communication module to obtain the position information of the associated train stored in the TMC. In

this case, the train switches to T2G mode, and movement authorization is generated based on the position data provided by the ground equipment. Additionally, the train can switch to T2G mode if the distance between trains becomes too large, or if the quality of the train-to-train communication link deteriorates to the extent that it cannot guarantee reliable communication.

3.3 Workflow

During normal operation within a zone, the train operates in T2T mode and runs in Automatic Driving mode (AM), as illustrated in Fig. 2. In this mode, the rear train and the front train maintain continuous train-to-train communication. The rear train uses the position information obtained from the front train to calculate the movement authorization (MA) in real time. This MA is then sent to the onboard ATP module to form the ATP curve. Simultaneously, both trains periodically report their position information to the TMC.

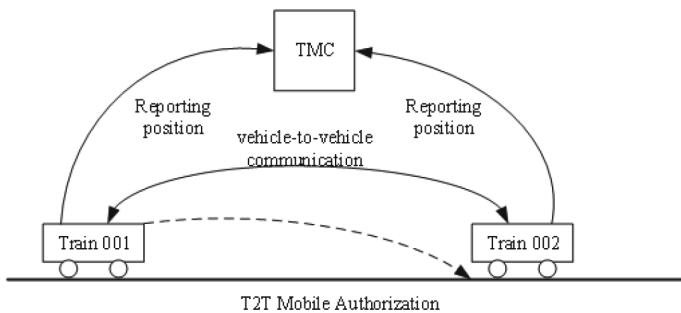


Fig. 2. Operation diagram of T2T mode

When communication between the rear train and the front train is interrupted due to a fault, the rear train's ATP triggers emergency braking. After mitigation, the driver selects the backup mode [1]. Upon confirmation with ITS, the train switches to Restricted Manual driving mode (RM) with a speed limit of 25 km/h. The onboard ATP generates the braking curve based on the last known position of the front train received in the previous communication cycle. Simultaneously, the rear train requests the position information of the front train from the TMC. If the TMC is able to update the front train's position information and communicate normally with the rear train, the rear train switches to the T2G mode. Subsequently, the rear train may upgrade its driving mode to either Automatic Driving mode with ground assistance (AM-G) or Fully Monitored mode with ground assistance (SM-G), based on the position information obtained from the TMC. The switching process is illustrated in Fig. 3.

When the train is in T2G mode, the TMC transmits the position of the front train to the rear train, as illustrated in Fig. 4. The front train operates in the regular T2T mode and periodically reports its position to the TMC. The

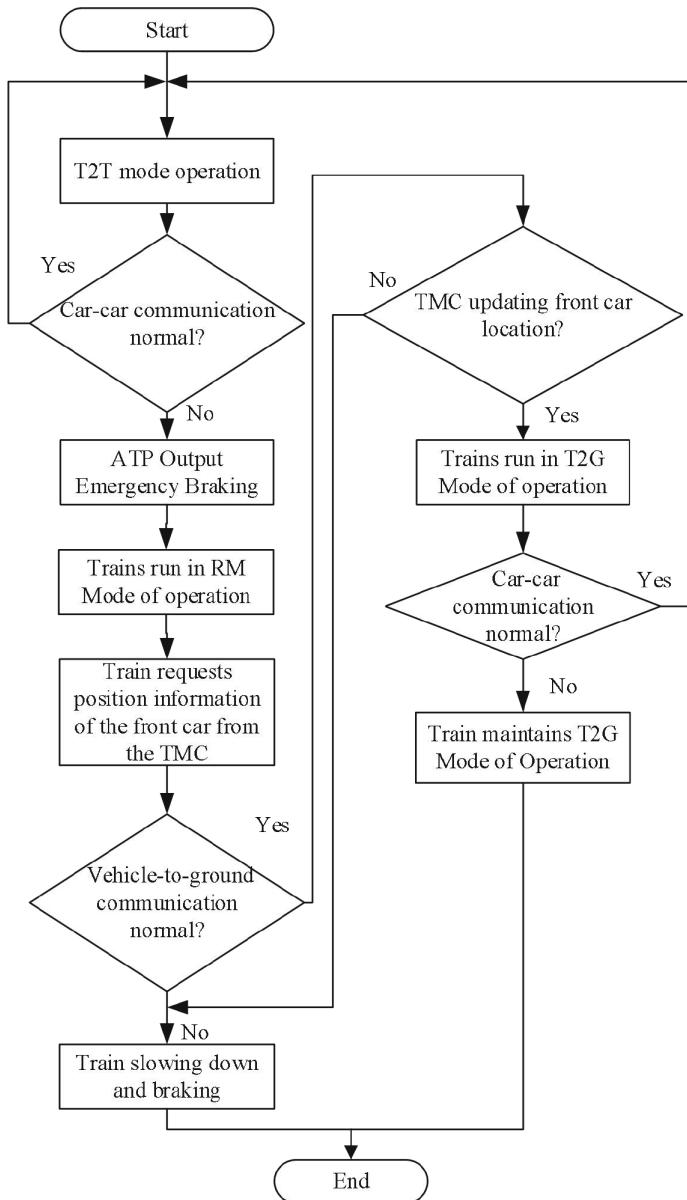


Fig. 3. Flowchart of system mode switching

TMC then transfers this position information to the rear train, enabling the rear train to generate a movement authorization based on the front train's position provided by the TMC. As the train position stored in the TMC corresponds

to the previous communication cycle, the tracking interval distance of the rear train becomes shorter.

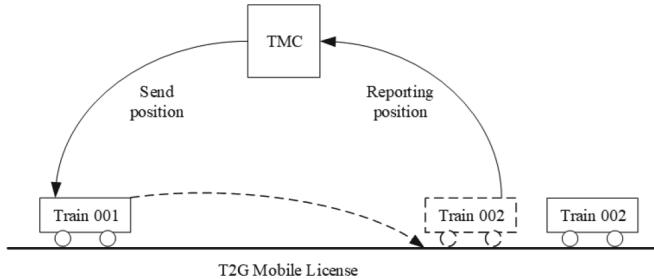


Fig. 4. Operation diagram of T2G mode

4 System Modeling

When the train is in T2G mode, the train-to-train communication system utilizes continuous wireless transmission. Therefore, even in this mode, the train can establish communication with the front train at any point. Upon successfully connecting with the front train and receiving valid position information, the train can seamlessly switch back to T2T mode for real-time communication [13]. As a result, the primary focus lies in modeling and analyzing the transition from T2T mode to T2G mode.

When the train switches from T2T mode to T2G mode, the immediate availability of T2G mode is delayed due to processing times inherent in the center equipment. The ground equipment and onboard equipment operate on independent clocks, resulting in a delay as the TMC must wait until the next communication cycle to relay the front vehicle's position information to the rear vehicle. There is also a delay in the transmission of train-to-ground information. Additionally, given the limited jurisdiction of the TMC, time must be allotted to determine if the front train falls within its jurisdiction. To facilitate the switch from T2T mode to T2G mode, an HTCPN model, depicted in Fig. 5, is established following process and function analysis. This model comprises the place set P and the transition set T [2].

The place set $P = \{\text{Train}, \text{T2T Invalid}, \text{decelerate}, \text{Fault}, \text{Confirm}, \text{RM}, \text{Model}, \text{K_RM}\}$.

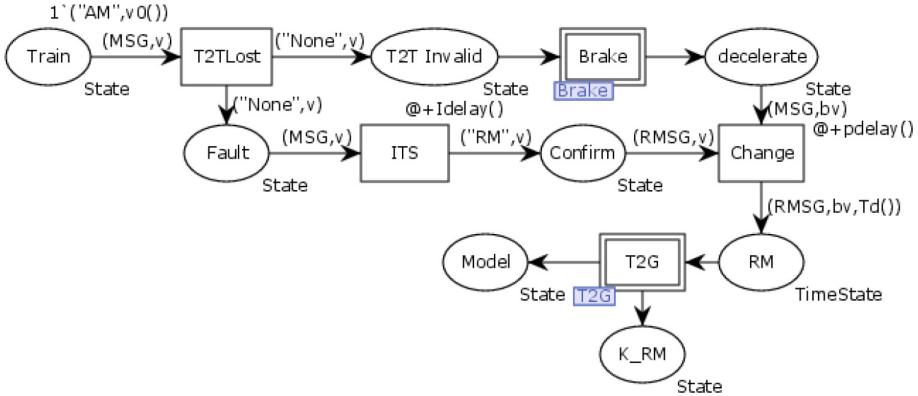


Fig. 5. Model of T2T switching to T2G

In the set P, Train and Model represent the initial T2T mode state and the mode state after switching, T2T Invalid and Fault represent the T2T unavailable state, RM and K_RM represent the RM mode of the train, decelerate represents the state of the train after deceleration, and Confirm represents the state that the train has been confirmed to switch mode after the ITS has received the train's request.

transition set $T = \{T2TLost, Brake, ITS, Change, T2G\}$.

Among them, T2TLost describes that the train has lost the position of the front train after the train-to-train communication failure, and the train starts to prepare for mode switching, ITS indicates that the train sends a mode switching request to the center ITS, according to the processing process, usually within 2min, using the delay function Idelay() to generate a random delay value, Change is the reaction time of the train driver to switch the mode switch after receiving the consent message, which is usually 0.4–1.2 s in complex cases, and is expressed by the function pdelay(), Brake is a replaceable transition containing a sub-page, which describes the process of decelerating and braking the train after the train-to-train communication failure, and T2G is also a replaceable transition, which describes the process of initialization of T2G mode.

State is the definition of TOKEN in the model, which is compounded by mode and speed (MSG, v), indicating the current mode and speed of the train, and the modes include AM and SM modes under T2T, AM-G and SM-G modes under T2G, RM mode under the speed limit, and None mode, the initial speed of the train is generated by the function v0(), which is taken to be 25–80 km/h, and the initial speed of the train is randomly generated using v0() to generate a 7–22 m/s initial speed. The time function Td() is added to Token TimeState to calculate the delay required before mode switching.

The top-level model describes that when the train is running normally in the zone, due to the train-to-train communication failure, the train decelerates and brakes to RM mode on the one hand, and confirms the request for switching

modes with the ground ITS on the other hand. When the ITS agrees to do so, it will go through the transition of T2G to determine whether the switching conditions are met, if they are met completely, it switches to the T2G mode, otherwise it keeps the braking in the RM mode.

Replaceable transition Brake describes the process of train deceleration braking in the case of unavailability of train-to-train communication, as shown in Fig. 6, the Place T2T Invalid is the state of the train after the failure of train-to-train communication, the transition Delay represents the process of the train's ATP triggering the emergency braking, and the delay includes the reaction time of the emergency braking triggering and the on-board equipment operation computation time, which is represented by the timefunction delay, it takes 0.75 s and 0.35 s respectively. After the train triggers braking, Release is the train braking state. Since the train speed is a randomly generated value, according to the current speed of the train, the concurrent states are utilized to describe the train. When the train speed is less than 7 m/s, the current speed is maintained in the transition Keep and the braking curve is planned by the ATP. When the train speed is greater than 7 m/s, the train is slowed down to 7 m/s braking in transition Braking, and finally output to decelerate, where the braking acceleration is taken as 0.8 m/s^2 .

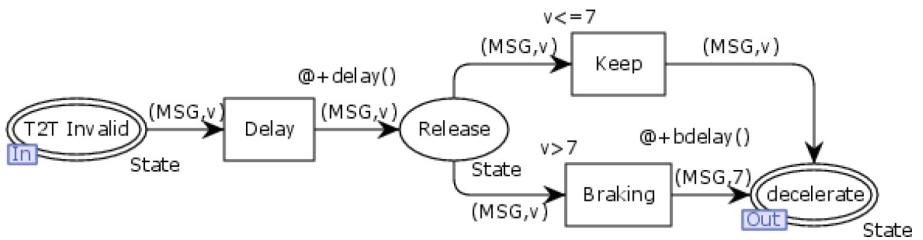


Fig. 6. Schematic diagram of Brake sub-mode

Replaceable transition T2G is the process of mode switching, as shown in Fig. 7. The train decelerates and switches to RM mode after confirmation from ITS, and then requests the position information of the front train from the ground equipment TMC. If the ground communication also fails, the train brakes in RM mode by the last received position information of the front train. If the ground communication is normal, in the transition ConfirmB, the train confirms whether the front train is within the jurisdiction of the current TMC and the running time left in the jurisdiction of the current TMC. If the front train is not within the jurisdiction of the current TMC, the train stays in the RM mode until it travels to the next TMC and then inquires about the position information of the front train. If the two trains belong to the same TMC jurisdiction, the transition TMCSSend send the communication situation and position of the front train to the rear car, and the rear train can switch the mode to T2G mode by on-board judgment.

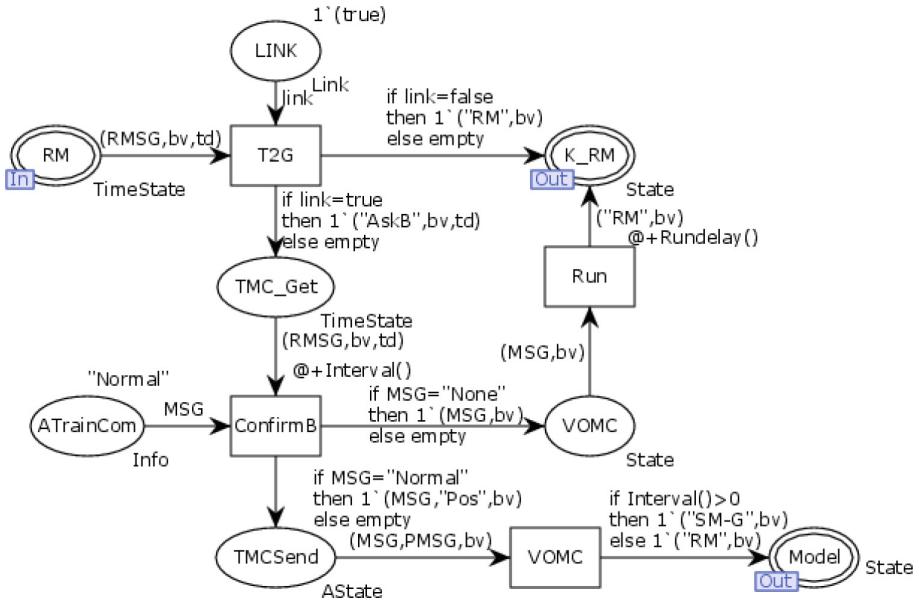


Fig. 7. Schematic diagram of T2G sub-mode

When the rear train communication is interrupted, if the two trains are under the jurisdiction of the same TMC1, as shown in Fig. 8, t_1 is the time that the train has been running in the jurisdiction of the current TMC1, and the time value is randomly generated because the train-to-train communication failure will occur at any moment, the normal tracking interval of the trains under the train-to-train communication is t_{T2T} , and the design interval of the T2G mode is t_{T2G} , and t_d is the time required to switch the rear train mode time required. Assuming that train 1 breaks down at t_1 and successfully switches mode after t_d time, at this time train 2 has been running at normal speed for t_d time, and then after t_2 time it will be out of the jurisdiction of TMC1, then when train 1 mode is successfully switched, the time when train 1 can be running in T2G mode is also t_2 . If during the t_d time of mode switching, train 2 has already been out of the jurisdiction of TMC1 ($t_2 < 0$), then Train 1 is also unable to obtain the position information of the front train by the ground TMC1 and can only operate in RM mode. That is, the train can only successfully switch modes when $t_2 > 0$, and the time t_2 that train 1 can operate in T2G mode is $t_2 = t_{T2G} - t_1 - t_{T2T} - t_d$. Where t_2 is represented in the model by the function Interval().

When the rear train has a train-to-train communication failure, if the front train has already left the jurisdiction of the current TMC1, then at this time, the rear train is unable to obtain the position information of the front train from the TMC1, then it generates a movement authorization with the end position of the current TMC1 jurisdiction and runs from the position where it was located at the time when the train-to-train communication was unavailable at the speed

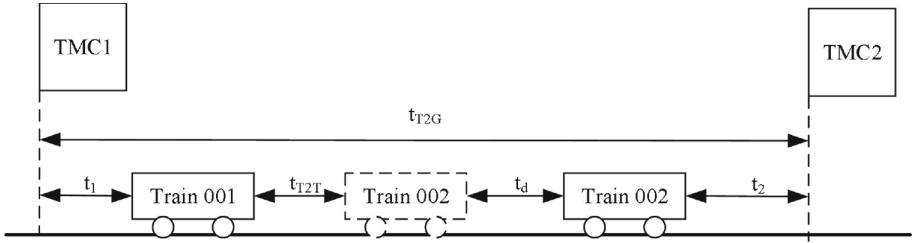


Fig. 8. Confirm B time parameter diagram

of the RM mode to the next TMC2 jurisdiction, and the time it takes for the time maximum ($t_{T2G}-t_1$) is the time required to run the distance ($v_0 \cdot t_{T2G}$) of the T2G design running interval at the speed (v_{RM}) of RM, the formula is $(v_0 \cdot t_{T2G})/v_{RM}$ where v_0 is the maximum speed at which the train can run, and since train-to-train communication failures can occur at any position in the run, this time value is generated in the model using the random function Rundelay(). When the rear train runs to the next TMC2 jurisdiction then register and query the position information of the front train [3].

5 Simulation Verification

Based on the established mode switching model, the key parameters are as follows: the train's maximum speed is 80 km/h, the speed limit in RM mode is 25 km/h, the tracking interval for train operation is set at 90 s in T2T mode, and the design intervals for T2G mode are 180 s, 210 s, and 240 s. A total of 10,000 simulations are conducted for these running intervals, and the resulting number of successful mode switches under various T2G mode design intervals is depicted in Fig. 9. Analysis indicates that the number of successful mode switches increases proportionally with the design interval. Switching failures predominantly occur when the front train exits the jurisdiction of the current TMC during the rear vehicle's switching process.

The time parameters of trains that can successfully switch to T2G mode are counted, and the distribution of mode switching time and runnable time of trains is obtained as shown in Fig. 10.

According to the data obtained from the simulation, statistics of the mode switching success rate and the mode switching time are shown in Table 1.

Assuming that the distance between the two stations A and B is 2000 m, the normal running speed of the two trains is 20 m/s, the maximum acceleration of the two trains is 1 m/s², and the braking acceleration is 0.8 m/s². Taking a set of data in the simulation results of the T2G mode with a design interval of 240 s, when the train has a train-to-train communication failure at 30 s, t_1 is 30 s, and the center ITS processing time is 80 s, then the train can run for 45 s under the T2G mode, and the speed-time curves in the two cases are shown in Fig. 11 and Fig. 12. It can be seen that under normal condition, the train departs from

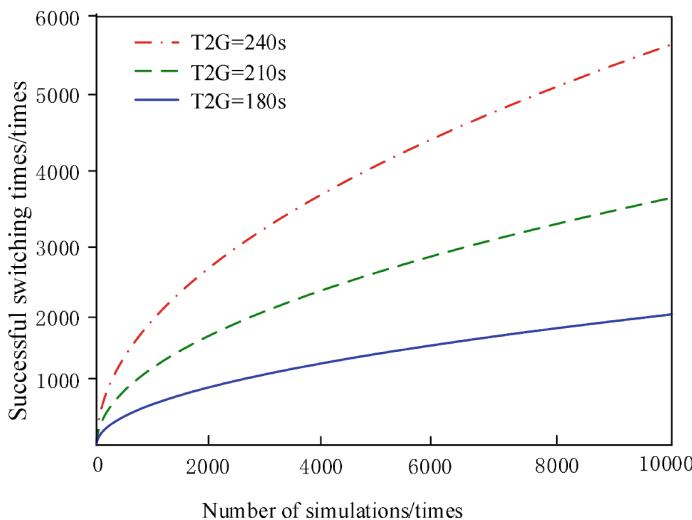


Fig. 9. Mode switching success times

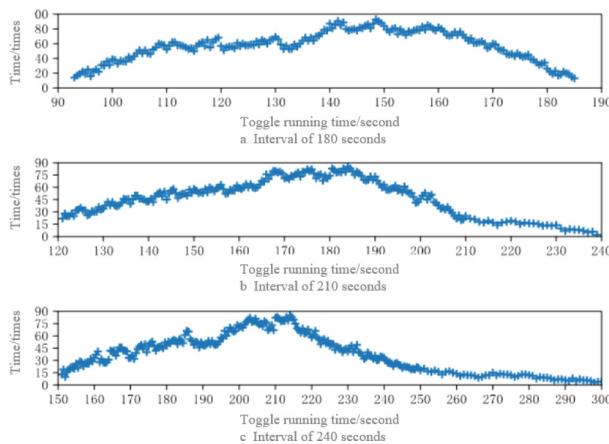


Fig. 10. Three kinds of T2G design interval switching operation time distribution

Table 1. Real-time system switching under different T2G design intervals

	$t(T2G) = 180\text{ s}$	$t(T2G) = 210\text{ s}$	$t(T2G) = 40\text{ s}$
Switch success rate/%	18.3	34.7	56.1
maximum/s	185	239	300
minimum/s	93	121	151
average/s	151	172	219
sample deviation /s	12	17	26

station A and arrives at station B after 123 s. When the train has a train-to-train communication failure, the train switches mode and runs in T2G mode, and finally arrives at station B after 184 s. That is to say, the delay is 61 s, which satisfies the 3 min delay requirement. At the same time, the red line is the train running in T2G mode for 20 s, which meets the running time within 45 s.

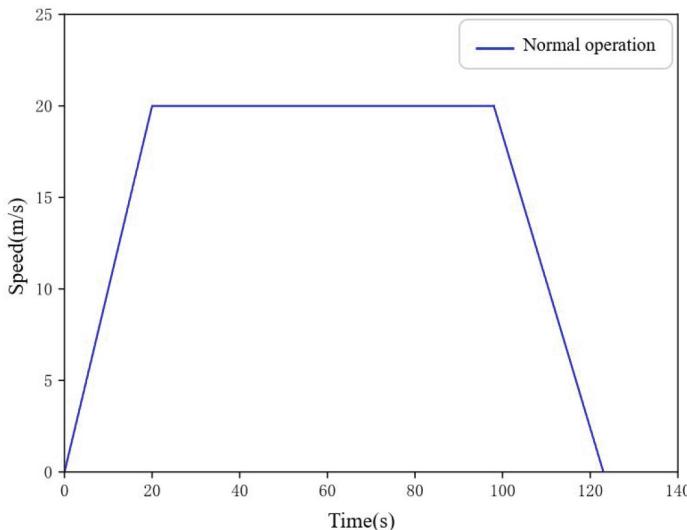


Fig. 11. Speed-time curve of normal train operation

After analyzing the charts provided, it is evident from the simulation results that switching the train from T2T mode to T2G mode during a train-to-train communication failure yields a higher success rate as the design interval of T2G mode increases. To ensure that train lateness does not exceed 3 min, a longer runtime for T2G mode is recommended. Specifically, when the design interval is set at 240 s, there is a higher likelihood of maintaining train lateness within the acceptable 3-min threshold. For compliance with the requirement of arriving less than 3 min late, the design interval of T2G mode should exceed 240 s. A larger design interval for T2G mode is more favorable, given the communication conditions permit it.

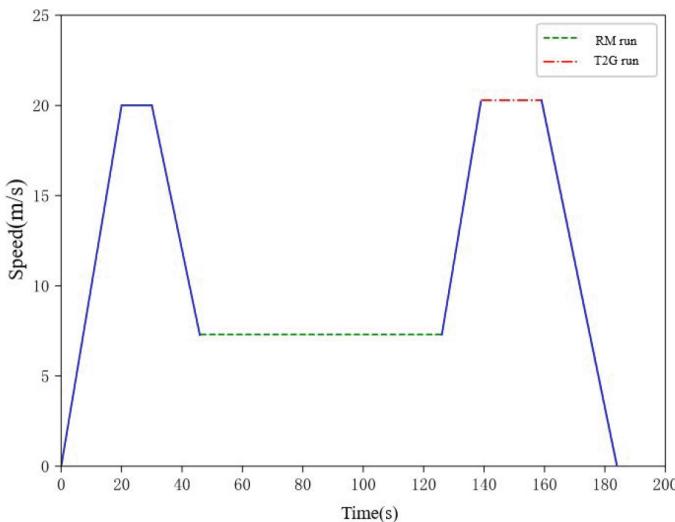


Fig. 12. Speed-time curve of train failure condition

6 Conclusion

- Based on the functional structure of the train-to-train communication system, two operation modes are proposed: train-to-train mode (T2T) and train-to-ground mode (T2G). T2G mode serves as the backup mode for the train control system, eliminating the necessity for the train to move to the beacon prior to mode switching. This approach enhances efficiency compared to the CBTC train control system.
- To assess the real-time capabilities of transitioning from T2T mode to T2G mode, an HTCPN model for mode switching is constructed. This model effectively captures and describes the process of mode switching within the train-to-train communication train control system.
- The simulation analysis of the model reveals that the success rate of train mode switching improves as the design interval of T2G mode increases. When a train is downgraded due to train-to-train communication failure, a 240-s design interval for T2G mode effectively limits delays to less than 3 min. For lines with distinct operational quality requirements, Table 1 can be consulted for designing the T2G mode interval to meet specific operational needs.

References

1. Atilgan, C., Kaymakçı, ÖT.: Modelling and hierarchical control of CBTC, pp. 1–6. IEEE, Istanbul (2018). isbn: 978-1-5386-7642-4. <https://doi.org/10.1109/CEIT.2018.8751762>
2. Droździel, P., et al.: Drivers' reaction time research in the conditions in the real traffic. Open Eng. **10**(1), 35–47 (2020). <https://doi.org/10.1515/eng-2020-0004>

3. Fakhereldine, A., Zulkernine, M., Murdock, D.: TrainSec: a simulation framework for security modeling and evaluation in CBTC networks. In: Milius, B., Collart-Dutilleul, S., Lecomte, T. (eds.) Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification, pp. 22–39. Springer, Cham (2023). isbn: 978-3-031-43366-5. https://doi.org/10.1007/978-3-031-43366-5_2
4. Gurník, P.: Next generation train control (NGTC): more effective railways through the convergence of main-line and urban train control systems. Transport. Res. Procedia **14**, 1855–1864 (2016). issn: 2352-1465. <https://doi.org/10.1016/j.trpro.2016.05.152>. <https://www.sciencedirect.com/science/article/pii/S2352146516301533>
5. Wang, J., et al.: Review on development of ERTMS-regional. J. China Rail. Soc. **34**(1), 60–64 (2012)
6. Chen, J., Kai, F., Sun, W.: Technical research on railway train control system based on vehicle-to-vehicle communication. China Rail. **21**(01), 79–84 (2021). <https://doi.org/10.19549/j.issn.1001-683x.2021.01.079>
7. Chen, J.: Modeling and Verification of Operational Scenario of Train Control System Based on Train to Train Communication. MA thesis. Beijing Jiaotong University (2018)
8. Zhao, L., He, C.: Analysis of the differences between the US PTC system and the European ERTMS. Rail. Signal. Commun. **47**(11), 56–59 (2011)
9. Pascoe, R.D., Eichorn, T.N.: What is communication based train control? IEEE Veh. Technol. Maga. **4**(4), 16–21 (2009). <https://doi.org/10.1109/MVT.2009.934665>
10. Chen, Q.: Research on train to train direct communication based on ultra-short wave. MA thesis. Lanzhou Jiaotong University (2015)
11. Song, H., Schnieder, E.: Availability and performance analysis of train-to-train data communication system. IEEE Trans. Intell. Transport. Syst. **20**(7), 2786–2795 (2019). <https://doi.org/10.1109/TITS.2019.2914701>
12. Wang, X., et al.: Enhancing communication-based train control systems through train-to-train communications. IEEE Trans. Intell. Transport. Syst. **20**(4), 1544–1561 (2019). <https://doi.org/10.1109/TITS.2018.2856635>
13. Yang, J., Zhang, Y., Jin, Y.: Technology and application of multiple train cooperative control system based on train to train communication. In: 2021 4th International Conference on Information Systems and Computer Aided Education. ICIS-CAE 2021, pp. 1654–1657. Association for Computing Machinery, Dalian (2021). isbn: 9781450390255. <https://doi.org/10.1145/3482632.3484011>
14. Chen, Y.: Reliability modeling and analysis of vehicle communication based on random Petri network. J. Saf. Environ. **21**(03), 984–989 (2021). <https://doi.org/10.13637/j.issn.1009-6094.2019.1663>
15. Zhang, Y., Ma, M., Wang, J.: Research on train route control method in onboard-centered train control system. J. China Rail. Soc. **43**(07), 77–86 (2021). <https://doi.org/10.3969/j.issn.1001-8360.2021.07.010>



Performance Evaluation of NLP Models for European Portuguese: Multi-GPU/Multi-node Configurations and Optimization Techniques

Daniel Santos^{ID}, Nuno Miquelina^{ID}, Daniela Schmidt^{ID}, Paulo Quaresma^{ID},
and Vítor Beires Nogueira^(✉)^{ID}

VISTA Lab, ALGORITMI Research Center, University of Évora, Évora, Portugal
`{dfsantos,daniela.schmidt,pq,vbn}@uevora.pt, d37384@alunos.uevora.pt`

Abstract. Natural Language Processing (NLP) research has predominantly focused on the English language, leading to a wealth of resources and advancements tailored to English. However, there is a growing need to extend these capabilities to other languages, such as European Portuguese, to ensure the inclusivity and accessibility of NLP technologies. In this study, we explore the evaluation of NLP models in the European Portuguese language using a multi-GPU/multi-node machine. We utilized various tools such as PyTorch, Accelerate, Transformers, and DeepSpeed with ZeRO Stage 3 to handle the computational demands of large-scale model training. We provide all the key aspects of our methodology to evaluate various models on translated GLUE tasks. Additionally, we introduce AiBERTa, a base model with 110 million parameters, developed and pre-trained on a corpus tailored for European Portuguese. This research highlights the effectiveness of advanced tools and distributed computing in scaling NLP model training, providing a foundation for future enhancements in European Portuguese language processing.

Keywords: NLP · Model Evaluation · Distributed Training

1 Introduction

Natural Language Processing (NLP) has made significant strides over the past decade, primarily driven by advancements in machine learning techniques and the availability of large-scale datasets. However, the majority of NLP research has predominantly focused on the English language, leading to a wealth of resources and advancements tailored specifically for English. This focus has created a disparity in NLP capabilities across different languages, with Portuguese being notably underrepresented despite its status as one of the most widely spoken languages globally [3].

The need to develop robust NLP models for Portuguese is critical, to ensure the inclusivity and accessibility of NLP technologies. Recent efforts have begun

to address this gap by translating key NLP benchmarks, such as the GLUE benchmark, into Portuguese. [9] The GLUE benchmark, introduced by Wang et al. [15], is a collection of diverse tasks designed to evaluate and compare the performance of NLP models on tasks such as sentiment analysis, textual entailment, and similarity scoring.

In this study, we explore the evaluation of NLP models in the European Portuguese language using a multi-GPU/multi-node machine. We employed various state-of-the-art tools to handle the computational demands of small to large-scale model training. These tools enabled us to efficiently manage and scale the training of multiple NLP models, leveraging the computational power of our supercomputer setup.

Furthermore, we introduce AiBERTa, our under-development base model with 110 million parameters, developed and pre-trained on a corpus tailored for European Portuguese.

This research highlights the effectiveness of advanced tools and distributed computing in scaling NLP model training. By providing detailed insights into our methodology and experimental setup, we aim to contribute to the growing body of work dedicated to enhancing NLP for European Portuguese and other underrepresented languages.

2 AiBERTa Model Development

AiBERTa [7] is a BERT-based language model tailored for European Portuguese. It aims to fill the gap in natural language processing resources specific to this language variant. The text source comes from Arquivo.pt¹, a project initiated by the Foundation for National Scientific Computing (FCCN), focuses on preserving Portugal's digital heritage by archiving web content related to the Portuguese cultural and scientific sphere.

2.1 Corpus Creation

The process to retrieve content from Arquivo.pt, involved several steps to ensure the collected data is relevant, clean, and suitable for building the AiBERTa language model. Here is a detailed explanation of the process:

1. Accessing Arquivo.pt APIs: this repository offers several APIs designed to facilitate the retrieval and analysis of archived web content. The main APIs used include:
 - Search API: allows performing text searches within the archived web content.
 - URL Search API: retrieves archived versions of a specific web page by providing its URL.
 - CDX Server API: provides access to the CDX index files.

¹ <https://arquivo.pt>.

2. Index Files - CDX Files: CDX files act as detailed indexes for archived web content, enabling efficient lookup based on specific criteria like URL, timestamp, and more. Each line in a CDX file represents a single archived document and contains metadata such as:
 - url of the captured content
 - timestamp of the capture
 - digest (hash) of the content
 - MIME type of the content
 - HTTP status code
 - offset and filename within the archive
3. Processing and Filtering CDX Files: to ensure only relevant content is processed:
 - Decompressing ZIP Files: each ZIP file contains multiple index files.
 - Filtering by MIME Type and HTTP Status: only content with specific MIME types (e.g., text/plain, text/html, application/pdf) and HTTP 200 status are considered.
4. Retrieving Content: once the relevant URLs are identified:
 - Parallel Processing: multiple processes and threads retrieve content in parallel.
 - Text Extraction: based on MIME type, different parsers are used:
 - text/HTML: uses the Trafilatura Python library.
 - binary Files (PDF, RTF, Word): Uses the Apache Tika Python library.
 - hash calculation: ensures uniqueness of the extracted text by comparing hashes.
 - database insertion: The cleaned text is stored in a database for further processing.
5. Text Processing: after retrieval:
 - sentence Extraction: The text is divided into unique sentences.
 - perplexity calculation: Perplexity values are calculated to ensure linguistic quality.
 - hash calculation for sentences: ensures uniqueness at the sentence level.

This structured process ensures that the corpus built from Arquivo.pt is diverse, with high-quality, and suitable for training the AiBERTa language model.

2.2 Pre-training

Pre-training is the initial phase of training a language model on a large corpus of text data before it is fine-tuned for specific downstream tasks. The objective is to learn general language representations that capture syntax, semantics, and various nuances of the language. With the corpus collected, using content from Portuguese sites provided by Arquivo.pt repository, is expected to capture the essence of the European-Portuguese language.

A tokenizer was built using the collected corpus. This tokenizer was created using WordPiece technique and has a size of 20K tokens. This helps in handling out-of-vocabulary words and reduces the size of the vocabulary.

The Masked Language Modeling (MLM) objective was performed during the pre-training. This objective randomly masks some tokens in the input and trains the model to predict the masked tokens based on the context. For example, the sentence “Está um bom jogo. O jogador deveria de ter marcado aquele [MASK]” /“It’s a good game. The player should have scored that [MASK]”, is suggested the token “golo” /“goal” to replace the [MASK] hidden word.

The pre-traning resorted to the BERT base model (uncased) with the following parameters 10 epochs, learning rate of 2e-5 and weight decay of 0.01.

2.3 Base Model Overview

The Bert base model (uncased) is one of the most widely used configurations of the BERT (Bidirectional Encoder Representations from Transformers) model. AiBERTa was pre-trained from the start and only sharing the same configuration as this model, avoiding any bias from the previous trainings. It is a powerful and versatile transformer-based language model. Having approximately 110 million of parameters, the model has the following technical specifications:

- layers: 12
- hidden size: 768
- attentions heads: 12
- maximum input length: 512 tokens

3 Tools and Infrastructure

In this section, we provide an overview of the tools and infrastructure that were essential for our experiments in evaluating NLP models in European Portuguese. We utilized several key libraries and frameworks to handle the complexities of training and fine-tuning large-scale language models.

3.1 PyTorch, Transformers and Accelerate

PyTorch² is an open-source deep learning framework widely adopted for its flexibility and ease of use. Developed by Facebook’s AI research group, PyTorch provides a Python package with high-level features such as tensor computation with robust GPU acceleration [10].

Transformers [16], developed by Hugging Face³, is a library that provides a general-purpose framework for developing and deploying NLP models. It supports a wide array of models such as BERT and GPT, facilitating easy access to pre-trained models and efficient fine-tuning on specific tasks. The Transformers library integrates seamlessly with PyTorch.

² <https://pytorch.org>.

³ <https://huggingface.co>.

Accelerate⁴ is a library designed to streamline the deployment of deep learning models on various hardware configurations [4]. It provides a high-level interface to manage multi-GPU and multi-node setups, significantly reducing the complexity involved in scaling up machine learning procedures. Accelerate was created for PyTorch and enables researchers and practitioners to focus more on model development and experimentation rather than on the specifics of distributed computing.

Accelerate optimises tensor operations across devices, utilizing techniques such as gradient accumulation and mixed precision training. These optimisations are critical in HPC environments to maximise computing efficiency and resource utilisation. We configured Accelerate to manage our distributed training tasks by specifying the appropriate device allocations, gradient accumulation, and precision levels, ensuring that each training process was optimized for the available hardware.

In our experiments, we utilized PyTorch and Transformers library to load pre-trained models and fine-tune them for specific NLP tasks, enhancing the performance and speed of our experimentation process. The Accelerate library was employed to handle the distribution of computations across multiple GPUs and nodes, ensuring efficient use of available resources and minimizing training time.

3.2 DeepSpeed and ZeRO

DeepSpeed⁵ is a deep learning optimization library developed by Microsoft, to enable the efficient training of large-scale models. It offers a wide range of features, including mixed precision training, gradient accumulation, and advanced memory optimization techniques. One of its most notable contributions is the ZeRO [11] (Zero Redundancy Optimizer) technology, which significantly reduces memory and computational load during training.

ZeRO is structured into three stages, each providing progressively greater memory efficiency:

- Stage 1: The optimizer states are partitioned across the processes.
- Stage 2: Adds gradient partitioning.
- Stage 3: Incorporates parameter partitioning, allowing the training of large-scale models by distributing memory and compute load across multiple GPUs.

In our experiment we focused on ZeRO Stage 3, which enables the training of large models by partitioning all model states (optimizer states, gradients, and parameters) across the available devices. This allows for a substantial reduction in memory consumption, enabling the training of models that would otherwise be too large to fit into GPU memory.

DeepSpeed with ZeRO Stage 3 was crucial for handling the largest models in our experiments while maintaining high training speeds. By leveraging these

⁴ <https://github.com/huggingface/accelerate>.

⁵ <https://github.com/microsoft/deepspeed>.

tools, we could efficiently manage memory and computation, enabling us to fine-tune models with millions of parameters across a multi-GPU/multi-node environment.

4 Experimental Setup

In this section, we outline the key components and configurations of our experimental setup. This includes the hardware specifications, the tools' setup, and the preparation of the dataset. These elements are crucial for replicating our experiments and understanding the context in which our results were obtained.

4.1 Hardware Configuration

Our experiments were conducted on a supercomputer, which comprises two compute nodes. The detailed specifications are as follows:

- CPU: Each node features Dual AMD Rome 7742, with 128 cores per node.
- System Memory: 1 TB of RAM per node.
- GPUs: 8 NVIDIA A100 Tensor Core GPUs per node, each with 40 GB of GPU memory, combine to total 320 GB of GPU memory across the node.

Slurm [17] was used to control the scheduling and distribution of computational tasks on the supercomputer. Slurm is a scalable cluster management and job scheduling system. Slurm enables the allocation of resources and balances the computational load among multiple users of the supercomputer. In particular, Slurm's advanced scheduling capabilities allowed us to prioritize jobs, manage queues, and ensure fault tolerance, all of which are critical in maintaining the efficiency of large-scale HPC environments.

However, this also introduced an additional layer of complexity. Implementing and integrating all the necessary libraries and tools, such as PyTorch, Accelerate, Transformers, and DeepSpeed with ZeRO Stage 3, required ensuring compatibility within the Slurm-managed environment.

4.2 Tools Setup

An important component of our setup was the script provided by Hugging Face for text classification task examples⁶, which served as the foundation for our workflow. Specifically, we used the `run_glue_no_trainer.py` script from the Hugging Face Transformers repository. This script, similar to `run_glue.py`, allows fine-tuning of any model available on the Hugging Face Hub for a text classification task, whether it is a GLUE task or custom data in a CSV or JSON file. The main difference is that this script exposes the bare training loop, enabling quick experimentation and the addition of any necessary customizations. This

⁶ <https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification>.

flexibility was crucial for adapting the script to meet our specific needs, including handling different model architectures and training configurations.

To run this script efficiently in our multi-GPU setup managed by Slurm, we configured the accelerate library. We found the examples from accelerate library particularly useful as a baseline for making accelerate work with Slurm:

- `submit_multigpu.sh`⁷
- `submit_multinode.sh`⁸

These examples provided a starting point to use both the Slurm job scripts and the configuration of accelerate using the accelerate launch command. With these resources, we were able to effectively set up and manage our distributed training environment, ensuring optimal utilization of our hardware resources. The scripts and configurations detailed in these examples served as a robust starting point, allowing us to modify and extend them according to the specific requirements of our experiments.

In summary, the combination of the accelerate library, and the Slurm job management system enabled us to perform efficient and scalable NLP model training and evaluation on a multi-GPU and multi-node setup.

4.3 Dataset Preparation

In the domain of Natural Language Processing, rigorous evaluation is crucial for assessing the capabilities and generalization of models. One benchmarking tool widely recognized in the NLP community is the General Language Understanding Evaluation (GLUE). This dataset is designed to provide a comprehensive suite of tests that measure a model's performance across multiple linguistic tasks.

GLUE Benchmark. The GLUE benchmark, introduced by Wang et al. [15], is a collection of nine diverse tasks designed to evaluate and compare the performance of NLP models on tasks such as sentiment analysis, textual entailment, and similarity scoring. These tasks include:

- CoLA - The Corpus of Linguistic Acceptability, measuring grammatical correctness.
- SST-2 - The Stanford Sentiment Treebank, for sentiment analysis.
- MRPC - The Microsoft Research Paraphrase Corpus, for paraphrase detection.
- STS-B - The Semantic Textual Similarity Benchmark, assessing the similarity between sentences.

⁷ https://github.com/huggingface/accelerate/blob/main/examples/slurm/submit_multigpu.sh.

⁸ https://github.com/huggingface/accelerate/blob/main/examples/slurm/submit_multinode.sh.

- QQP - Quora Question Pairs, evaluating whether a pair of questions are semantically equivalent.
- MNLI - The Multi-Genre Natural Language Inference, a task of textual entailment across multiple sources.
- QNLI - Question Natural Language Inference, adapted from the Stanford Question Answering Dataset.
- RTE - Recognizing Textual Entailment, focused on entailment tasks.
- WNLI - Winograd NLI, a test of coreference resolution based on the Winograd schema.

These tasks collectively challenge the model’s understanding of language, testing both breadth and depth of linguistic features.

Detailed Overview of Selected GLUE Tasks

Microsoft Research Paraphrase Corpus (MRPC) The MRPC task is aimed at identifying whether two sentences are paraphrases of each other, essentially evaluating a model’s ability to detect semantic equivalence. The corpus consists of sentence pairs automatically extracted from online news sources, with human annotations indicating whether each pair captures the same informational content. Evaluating on MRPC requires models to understand nuances in word choice and syntax that may change the surface form of a sentence while retaining or altering its meaning.

Recognizing Textual Entailment (RTE). The RTE task involves determining whether a given hypothesis can logically be inferred from a premise sentence. This task tests a model’s ability to handle logical reasoning and understand context within a text. Models are judged on their ability to accurately assess entailment or contradiction in diverse contexts.

Semantic Textual Similarity Benchmark (STS-B). The STS-B task requires models to assign a similarity score to pairs of sentences on a continuous scale from 1 (not similar at all) to 5 (semantically equivalent). This task assesses a model’s ability to interpret and compare the meanings of sentences. Performance on STS-B reflects a model’s nuanced understanding of semantic relationships.

Winograd Schema Challenge (WNLI). The WNLI task is designed to test a model’s ability to handle coreference resolution within the framework of the Winograd Schema Challenge. This task presents sentences containing pronouns whose antecedents are ambiguous and requires the model to determine the correct reference. It is a test of language understanding that probes a model’s common-sense reasoning and context processing capabilities.

The training of the models for these tasks was made through their online submission system which enabled us to upload the models predictions for evaluation on the secret test set labels. By using this platform, we obtain the standardized metrics for performance comparison and ensured that our evaluations were consistent with other research.

5 Overview of Evaluated Models

To assess the performance of Natural Language Processing (NLP) models tailored for European Portuguese, we employed the GLUE benchmark, adapted to Portuguese through the ExtraGLUE dataset [9]. This allowed for a consistent and robust comparison across a diverse set of models pre-trained for European Portuguese, each designed with unique capabilities and scale. The models evaluated include:

- GlórIA [6]: A large generative language model specifically focused on European Portuguese. GlórIA is built on the GPTNeo architecture, featuring 1.3 billion parameters, 24 layers, a hidden size of 2048, and functions as a decoder-only language model (LLM).
- Albertina PT [12]: This model family, an adaptation of the BERT architecture influenced by the DeBERTa model, consists of three variants:
 - Base model with 100 million parameters
 - Mid-size model with 900 million parameters
 - Large model with 1.5 billion parameters
- Gervásio PT [13]: Developed from the LLaMA-2 7B model, this decoder-only model of the LLaMA family boasts 7 billion parameters. It includes 32 hidden layers, 32 attention heads, and utilizes the Byte-Pair Encoding (BPE) algorithm via SentencePiece for its tokenizer, supporting a vocabulary size of 32,000.
- BLOOM [1]: As an autoregressive LLM, BLOOM generates coherent text in 46 languages and 13 programming languages. 1.1 billion parameter model was used.
- BLOOMZ mt 7.1B [8]: The BLOOMZ MT (Multitask) are multilingual models which are particularly suited for non-English prompting.
- Multilingual GPT model [14]: Utilizing GPT-2 sources along with sparse attention mechanisms provided by Deepspeed and Megatron, this model architecture reproduces GPT-3's functionality. It covers multiple languages, enhancing capabilities for low-resource languages.
- Carvalho 1.3B [2]: A 1.3 billion parameter model, which was trained using a GPT architecture and specifically fine-tuned on a Galician-Portuguese corpus to enhance their regional linguistic adaptability. The lexical and syntactic similarity between these two language variations is very high.
- AiBERTa Base: Our base model with 110 million parameters trained for European-Portuguese.

6 Experiments with Distributed Configurations

In this section, we present our experiments conducted with various distributed configurations to evaluate the impact on training performance and efficiency. The configurations tested include different numbers of GPUs and nodes to determine the optimal setup. In our experiment, we fine-tuned the Albertina-PTPT 900 m on the MRPC task and kept the total batch size constant at 64 across all configurations by adjusting the batch size per device.

Table 1. Time and memory comparison across multiple single and multi-gpu configurations

	1 GPU	2 GPUs	4 GPUs	8 GPUs
Multi-GPU				
Time	04:24	04:22	04:21	04:32
Peak Mem. Alloc. (Per GPU, MB)	28 285	16 795	9 999	6 582
Total Mem. Alloc. (All GPUs, MB)	28 285	33 589	39 995	52 658

Table 2. Time and memory comparison across various multi-node configurations

	1 GPU per node	2 GPUs per node	4 GPUs per node
Time	04:56	06:31	07:00
Peak Mem. Alloc. (Per GPU, MB)	16 794	9 999	6 582
Total Mem. Alloc. (All GPUs, MB)	33 589	39 996	52 655

According to the results shown in Table 1, increasing the number of GPUs did not reduce the training time. This could be attributed to potential inefficiencies introduced by small batch sizes per device, as well as the same number of training steps across experiments, as the total batch size remained constant. This means that while more GPUs process data in parallel, the overall training duration remains similar due to the fixed number of steps required to complete an epoch. Additionally, communication overhead can also affect training time.

The peak memory allocated per GPU decreased as the number of GPUs increased, from 28,285 MB for 1 GPU to 6,582 MB for 8 GPUs. This trend indicates effective memory distribution across GPUs, which is crucial for training larger models which may not fit on a single GPU. With more GPUs, the memory load is shared, reducing the peak memory demand on each individual GPU.

In multi-node configuration (see Table 2), the overhead of inter-node communication and synchronization can contribute to longer training times when compared to single node setup.

In conclusion, while using multiple GPUs and nodes can facilitate the training of larger models by distributing memory load, the training time benefits are limited when the total batch size remains constant.

Table 3. Time, memory and performance comparison across various multi-gpu configurations with variable total batch size

	1 GPU	2 GPUs	4 GPUs	8 GPUs
Total Batch Size	64	128	256	512
Time	04:24	02:15	01:14	00:39
Peak Mem. Alloc. (Per GPU, MB)	28 285	23 632	20 250	18 557
Total Mem. Alloc. (All GPUs, MB)	28 285	47 263	81 001	148 457
Accuracy	0.86	0.85	0.81	0.70
F1	0.90	0.89	0.85	0.81

In our second experiment, we fine-tuned the Albertina-PTPT 900 m model on the MRPC task, keeping the batch size per device constant, which means the total batch size will vary across different multi-GPU configurations. The results provide valuable insights into the trade-offs between training efficiency, memory usage, and model performance as shown in Table 3.

As expected, the training time decreased significantly with the increase in the number of GPUs. Peak Memory Allocation per GPU has also decreased slightly. This trend shows that while each GPU handles a constant batch size, the distribution of the total memory load becomes more balanced across multiple devices. Interestingly, the model performance, as measured by accuracy and F1 score, showed a decreasing trend with increasing total batch size. This decline in performance could be attributed to the impact of larger batch sizes on model convergence. Larger batch sizes often lead to noisier gradient estimates, which can hinder the model's ability to converge to an optimal solution [5]. In conclusion, this experiment emphasizes the significance of balancing batch size, training time, and model performance. While multi-GPU configurations can significantly decrease training time, careful attention is required to maintain model performance.

Table 4. Time, memory, and performance comparison across different mixed precision configurations

	FP32	FP16	BF16
Time	08:06	04:22	04:20
Peak Mem. Alloc. (Per GPU, MB)	28 124	16 795	16 794
Total Mem. Alloc. (All GPUs, MB)	56 249	33 589	33 588
Accuracy	0.87	0.86	0.87
F1	0.90	0.89	0.90

In our third experiment, we compared the performance, memory usage, and training time of the Albertina-PTPT 900 m model on the MRPC task using different precision formats: FP32, FP16, and BF16. Table 4 shows us that the training time and memory bandwidth is significantly reduced when using FP16 and BF16 while model performance remains consistent across all mixed precision formats. This analysis indicates that using mixed precision formats, specifically FP16 and BF16, can substantially improve training efficiency by reducing training time and memory usage without major drawbacks. Additionally, the reduced memory usage can allow for larger batch sizes or more complex models to be trained on the same hardware.

In our last experiment, we explored the impact of offloading the optimizer and parameters to the CPU on training time and memory usage. We used the Albertina-PTPT 900 m model on the MRPC task with a single GPU, comparing the scenarios of no offloading versus CPU offloading.

Table 5. Time and memory comparison with and without optimizer/parameter offloading to CPU

	No Offload	CPU Offload
Time	04:24	14:53
Peak Mem. Alloc. (MB)	28 285	16 437

Table 5 shows us that the training time increased significantly when offloading to the CPU. This substantial increase can be attributed to the introduced latency as parameters and optimizer states need to be continuously moved between the CPU and GPU during training, slowing down the process.

However, the peak memory allocated has decreased considerably. This reduction in memory usage demonstrates the effectiveness of offloading in freeing up GPU memory, making it possible to train larger models or use larger batch sizes within the same GPU memory constraints.

In conclusion, offloading optimizer and parameter states to the CPU can be a useful strategy for managing GPU memory usage, particularly in scenarios where GPU memory is a constraint. However, the significant increase in training time necessitates careful consideration. In cases where training speed is a priority, avoiding offloading may be preferable.

Figure 1 provides a comparative analysis of training time and memory usage across various configurations, including different precision formats (FP32, FP16, BF16), the use of 2 GPUs with double the batch size, and CPU offloading. As illustrated, using mixed precision formats like FP16 and BF16 significantly reduces training time and memory usage compared to FP32. Additionally, while the 2 GPU configuration with a batch size of 128 further improves training time, it comes at the cost of increased memory consumption. CPU offloading, while effective in reducing GPU memory usage, leads to a substantial increase

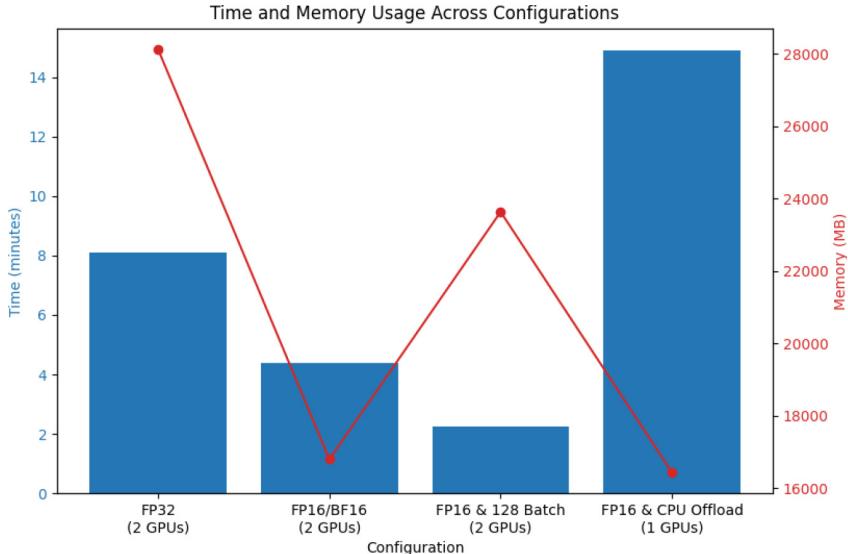


Fig. 1. Time and Memory Usage Across Configurations

in training time due to the overhead associated with data transfer between the GPU and CPU.

7 Evaluation and Results

In this section, we present the evaluation metrics and results of our experiments conducted on various NLP models.

The models were benchmarked using a variant of the GLUE dataset called ExtraGLUE, which is adapted to Portuguese through automatic translation of selected tasks from the GLUE and SuperGLUE benchmarks. This translation ensures the models are evaluated in a language-specific context, thereby providing more accurate and relevant results for Portuguese.

To optimize the training of our NLP models, particularly given the high computational overheads, we utilized a batch size of 16 per device and 2 gradient accumulation steps. For most models 2 GPUs were used, while Gervásio PT, due to its extensive parameter count (7 billion), we adjusted the batch size down to 4 per device to accommodate the model's demands within the available GPU memory and used 8 GPUs. Likewise, for Bloomz with 7.1 billion parameters, 8 GPUs were utilized with CPU offloading. Training epochs were set to five across all experiments.

7.1 Hyperparameter Tuning

The models underwent fine-tuning over a range of hyperparameters:

- Learning rates of 1e-4 and 1e-5
- Utilization of either a linear or constant scheduler to manage learning rate adjustments
- Fixed seeds for training reproducibility (41, 42, 43)

These hyperparameters were selected to optimize performance while ensuring that the models could adequately learn from the translated tasks.

7.2 Results

Table 6. Evaluation results on the Portuguese ExtraGLUE tasks.

Models	RTE		MRPC		STS-B		WNLI Acc
	Acc	F1	Acc	Pearson	Spearman		
Encoders							
AiBERTa Base	55.3	83.2	75.9	80.2	79.4	58.9	
Albertina-PTPT 100 m	55.4	87.6	83.1	84.5	84.1	65.1	
Albertina-PTPT 900 m	80.6	89.8	86.6	88.7	88.3	65.1	
Albertina-PTPT 1.5B	82.9	90.3	87.2	88.7	88.4	59.6	
Decoders							
Carvalho_pt-gl 1.3B	68.0	86.0	79.9	82.6	81.9	65.1	
Gloria 1.3B	63.8	85.2	79.1	82.0	81.2	65.1	
Gervásio 7B	83.2	90.5	87.0	87.9	87.6	64.4	
mGPT 1.3B	58.9	85.5	79.3	78.3	76.9	65.1	
Bloom 1.1B	71.5	87.7	82.7	85.1	84.3	63.7	
Bloomz mt 7.1B	81.4	89.1	85.4	86.4	85.5	65.1	

The evaluation results are shown in Table 6, where both encoders and decoders were evaluated in the four tasks previously described.

Decoder models, such as those designed for generative tasks, typically excel in generating text based on the input they receive. However, their architecture may not be inherently optimized for classification tasks like those found in the GLUE benchmark, where the goal is often to predict a label or a class based on input text. These tasks require precise understanding and categorization rather than open-ended text generation. Instead of adapting the decoder models to output a single classification label, another approach could be instruction tuning. This involves training the model on a range of NLP tasks given as instructions in natural language. For example, prompting a decoder model with instructions like

”Classify the sentiment of the following text:” could help it apply its generative capabilities within a classification framework.

Additionally, given the limited size of the WNLI dataset, leveraging knowledge transfer from larger datasets through multi-task learning or transfer learning could significantly improve model robustness and generalization. Training models on other tasks and then fine-tuning on WNLI may provide the necessary depth of understanding navigating its complex coreference resolutions. Otherwise, our results suggest that the models are unable to outperform a baseline model.

8 Conclusion

In this study, we have explored the evaluation of NLP models for European Portuguese using a multi-GPU/multi-node machine, focusing on the impact of various distributed training configurations, mixed precision formats, and optimizer/parameter offloading strategies. Our experiments were conducted using state-of-the-art tools, including PyTorch, Accelerate, Transformers, and DeepSpeed with ZeRO Stage 3, to handle the computational demands of large-scale model training.

In our experiments with different multi-GPU/multi-Node configurations, we found that while distributing memory load across more GPUs can support the training of larger models, it is crucial to balance batch size, training time, and model performance to achieve optimal results. We also explored the impact of mixed precision training and optimizer/parameter offloading strategies on training efficiency and memory usage. Mixed precision training with FP16 and BF16 significantly reduced training time and memory usage without compromising model performance, highlighting its efficiency for large-scale NLP model training. Additionally, offloading optimizer and parameter states to the CPU effectively reduced GPU memory usage, allowing for larger models or batch sizes to be trained. However, this benefit came with increased training times due to the latency introduced by data transfer between the GPU and CPU. These findings highlight the significance of balancing memory management strategies with training efficiency in order to improve overall performance.

Additionally, we introduced AiBERTa, a custom base model with 110 million parameters developed and pre-trained on a corpus tailored for European Portuguese. AiBERTa’s initial performance marks our first step towards improving NLP capabilities for Portuguese, contributing to the inclusivity and accessibility of NLP technologies for the language.

Having gained a better grasp and insight into distributed training and optimizations, we can now maximize the usage of system used to further improve our model. Future work will focus on leveraging these insights to enhance the performance of AiBERTa.

In conclusion, our study highlights the effectiveness of advanced tools and distributed computing in scaling NLP model training. By providing detailed insights into our methodology and experimental results, we aim to contribute to

the expanding research dedicated to enhancing NLP for European Portuguese and other underrepresented languages. This research lays a foundation for future enhancements and optimizations in large-scale NLP model training and evaluation.

Acknowledgments. This work was supported by the Portuguese Foundation for Science and Technology (FCT) in the framework of the project AiBERTa - 2022.03882.PTDC.

References

1. BigScience: Bigscience language open-science open-access multilingual (bloom) language model. International (2022)
2. Gamallo, P., et al.: A galician-portuguese generative model (2024)
3. Garcia, G.L., et al.: Introducing bode: a fine-tuned large language model for portuguese prompt-based task (2024)
4. Gugger, S., et al.: Accelerate: training and inference at scale made simple, efficient and adaptable (2022). <https://github.com/huggingface/accelerate>
5. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: generalization gap and sharp minima (2017)
6. Lopes, R., Magalhaes, J., Semedo, D.: GlórIA: A generative and open large language model for Portuguese. In: Gamallo, P., Claro, D., Teixeira, A., Real, L., Garcia, M., Oliveira, H.G., Amaro, R. (eds.) Proceedings of the 16th International Conference on Computational Processing of Portuguese, pp. 441–453. Association for Computational Linguistics, Santiago de (2024). <https://aclanthology.org/2024.propor-1.45>
7. Miquelina, N., Quaresma, P., Nogueira, V.B.: Generating a European Portuguese bert based model using content from arquivo.pt archive. In: Yin, H., Camacho, D., Tino, P. (eds.) Intelligent Data Engineering and Automated Learning – IDEAL 2022, pp. 280–288. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21753-1_28
8. Muennighoff, N., et al.: Crosslingual generalization through multitask finetuning. arXiv preprint [arXiv:2211.01786](https://arxiv.org/abs/2211.01786) (2022)
9. Osório, T., et al.: Portulan extraglue datasets and models: kick-starting a benchmark for the neural processing of Portuguese (2024)
10. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library (2019)
11. Rajbhandari, S., Rasley, J., Ruwase, O., He, Y.: Zero: memory optimizations toward training trillion parameter models (2020)
12. Rodrigues, J., et al.: Advancing neural encoding of Portuguese with transformer albertina pt-* (2023)
13. Santos, R., Silva, J., Gomes, L., Rodrigues, J., Branco, A.: Advancing generative AI for Portuguese with open decoder gervásio pt-* (2024)
14. Shliazhko, O., Fenogenova, A., Tikhonova, M., Mikhailov, V., Kozlova, A., Shavrina, T.: mgpt: few-shot learners go multilingual (2022). <https://doi.org/10.48550/ARXIV.2204.07580>. <https://arxiv.org/abs/2204.07580>
15. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: a multi-task benchmark and analysis platform for natural language understanding (2019)

16. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics (2020). <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
17. Yoo, A.B., Jette, M.A., Grondona, M.: SLURM: simple linux utility for resource management. In: Feitelson, D., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2003. LNCS, vol. 2862, pp. 44–60. Springer, Heidelberg (2003). https://doi.org/10.1007/10968987_3



FusionFrame: A Fusion Dataflow Scheduling Framework for DNN Accelerators via Analytical Modeling

Liutao Zheng¹ , Huiying Lan² , Xiang Liu² , Linshan Jiang² , and Xuehai Zhou¹

¹ University of Science and Technology of China, Hefei 230026, China

² National University of Singapore, Singapore 119077, Singapore

hy.lan@nus.edu.sg

Abstract. The growing complexity of DNN models and the widening gap between compute power and memory bandwidth necessitate fusion dataflows to reduce off-chip memory access. However, designing these dataflows across diverse DNNs and DNN accelerator architectures requires a flexible and accurate scheduling framework to explore the vast design space efficiently. Unfortunately, current state-of-the-art (SotA) frameworks struggle to explore these fusion dataflows by mapping limited fusion patterns on less scalable accelerator architectures. In this paper, we present a fusion dataflow framework called FusionFrame. FusionFrame expands the schedule space by characterizing it from three dimensions: operator fusion, loop tiling, and hardware mapping. To capture the extensive schedule space, we design a memory-centric representation to explore various fusion dataflows. Based on this representation, we develop an analytical model that evaluates on-chip data movement and latency for flexible fusion dataflows, enabling hierarchical fusion of operators on DNN accelerators with multi-level memory architectures. This cost model is validated against a taped-out DNN accelerator, Cambricon-Acc. Extensive case studies are then conducted to explore the performance impacts of various schedule space trade-offs, showing that FusionFrame demonstrates a latency improvement of up to 63.9% compared to SotA methods.

Keywords: Accelerator · Fusion · Simulation and modeling · Tensor programs

1 Introduction

Deep Neural Networks (DNNs) have become a transformative force across various fields, including computer vision [4, 6, 20], natural language processing [23], and speech recognition [19]. However, their remarkable capabilities come at a cost: immense computational and memory demands. Efficient and scalable execution of DNNs on hardware platforms remains a significant challenge. Domain-specific accelerators (DSAs) [2, 3] have emerged as a powerful solution for accelerating DNN computations. These accelerators often leverage hierarchical on-chip

memory architectures to store the large weights and neuron activations DNNs require. This multi-level memory hierarchy offers benefits such as reduced data transfer between memory and processing units. However, it also introduces complexity in the scheduling process – the crucial task of mapping computations to the available processing elements (PEs) within the accelerator.

To solve this issue, scheduling frameworks [13, 15, 18] have been proposed to optimize the execution of DNN for factors such as latency and memory footprint. However, their traditional operator-by-operator approach suffers from excessive data movement between off-chip and on-chip memory, hindering performance on DSAs due to limited memory bandwidth. To address this, fusion dataflow frameworks have emerged [14, 25, 27]. These frameworks strategically combine adjacent operators in the DNN graph, keeping intermediate data on-chip. This eliminates unnecessary data transfers and significantly improves performance. Fusion dataflow scheduling essentially partitions the DNN graph into efficiently fused on-chip operator groups and optimizes data access patterns through loop transformations like tiling and dimension mapping.

However, designing an effective fusion scheduling framework presents three significant challenges:

- *Exploring the operator fusion space:* Identifying the optimal operator fusion strategy is a complex task. It employs search algorithms to traverse the entire compute graph and determine efficient fusion patterns. Due to the vast search space and inherent combinatorial complexity, exploring the entire operator fusion space becomes an NP-hard problem [17]. Existing frameworks [1, 14, 25, 27] often explore the fusion space using predefined fusion patterns or algorithms that rely on strong assumptions, limiting their overall flexibility.
- *Optimizing tiling strategies:* Selecting appropriate tiling sizes and dimensions to fit data into on-chip memory is a significant challenge in fusion dataflows [13, 15, 18]. The framework needs to be aware of the data layout and infer the tiled data shape for each dimension across all intermediate data elements. Additionally, the impact of tiling different dimensions on data shape inference must be carefully considered. Current frameworks like DeFiNES [14] and Genetic-A [12] have limitations in the tiling dimensions they consider, restricting their exploration of the tiling schedule space.
- *Mapping to multi-level memory hardware architectures:* Mapping DNNs onto diverse accelerators with multi-level memory hierarchies is a non-trivial problem. Existing works often exhibit limitations in this area. Some frameworks, like ConvFusion [24], and DNNFuser [9], only consider mapping to single-level memory. These approaches cannot explore schedule optimizations for DNN execution on hardware platforms with multi-levels of memory.

To address these challenges, we propose FusionFrame, a novel framework for optimizing DNN scheduling. FusionFrame enhances DNN schedules by establishing a three-dimensional schedule space and representing it with a memory-centric representation. Additionally, it introduces a unified analytical cost model that outperforms current SotA performance metrics across various fusion dataflows.

Our main contributions are as follows:

- We characterize the design space from three dimensions: *operator fusion*, *loop tiling*, and *hardware mapping*. These dimensions encompass all essential decisions for optimizing fusion dataflows. Furthermore, we propose a novel memory-centric representation that effectively expresses the DNN schedule space, allowing efficient exploration of different fusion dataflow optimization opportunities (Sect. 3).
- We present an analytical model that iteratively explores the schedule space, generates a memory-centric representation, and evaluates its performance. This model enables the efficient and accurate estimation of key performance metrics for DNN schedules, including memory footprint and latency (Sect. 4).
- We conduct comprehensive case studies to demonstrate the effectiveness of FusionFrame. These studies analyze the trade-offs associated with different selections across the three dimensions, providing valuable insights into optimizing DNN schedules (Sect. 6).

2 Background

2.1 Operator Fusion Exploration

The exploration space for operator fusion is vast due to the numerous possible combinations within different DNN architectures. Current SotA methods, such as TileFlow [27], DeFiNES [14], and ACCO [26], can be categorized into two main approaches.

The first approach focuses on fixed operator fusion patterns, like TileFlow [27], which emphasize solving the challenge of scheduling within fixed fusion patterns. This method is suitable for consistent building blocks, such as attention blocks [23] commonly used in NLP and large language models (LLMs). However, it is inadequate for DNNs used in computer vision, which involve complex network architectures and intricate computing and memory access patterns. The second approach involves automatic searching for operator fusion combinations, as seen in DeFiNES [14] and ACCO [26]. These methods employ a depth-first fusion strategy, greedily fusing operators into stacks until the weight size fits on-chip memory, then scheduling each fused stack to determine the memory levels for each operator. However, this fusion strategy does not consider tile size determination, relying instead on user-provided tile sizes. Consequently, the fusion strategy is unaware of hardware specifications, leading to sub-optimal performance.

2.2 Tiling Strategy Exploration

The performance impact of tiling various dimensions for single operators has been extensively studied [7, 13, 15], but it remains underexplored in the context of operator fusion. Fused operators are typically tiled along their output dimensions [12, 14, 16, 26] due to the complexities introduced by input tiling,

which requires synchronization of multiple output tiles from each input tile and becomes increasingly complex with deeper fusions. In contrast, output tiles can be treated as independent units, facilitating straightforward parallelization. For sliding-window-based operators (e.g., CONV, POOL), tiling in the output height and width dimensions can lead to significant redundant data overhead, which exacerbates with deeper fusion [14, 26, 27]. However, current scheduling frameworks predominantly focus on tiling the output height and width dimensions, neglecting the output channel dimension. This limitation results in the termination of fusion if the combined weight size of the fused operators exceeds the on-chip memory capacity, restricting their applicability in CNNs with large convolution weights. Taking ResNet-50 [4] as an example, the network’s weights increase with depth, and it includes convolutions with 3×3 kernels and 512×512 input and output channels. This requires at least 2.3MB of on-chip memory for a single operator with int8 data type, posing significant memory demands on deep learning accelerators [2, 3]. This issue is not addressed in current SotA works [14, 26, 27].

2.3 Hardware Mapping Exploration

Mapping DNN workloads to various DNN accelerators is complex due to the multiple levels of memory hierarchies involved. Deciding which operator should be mapped to which memory level requires careful consideration. While significant research has focused on single operator mapping—such as in Timeloop [18], MAESTRO [13], and ZigZag [15]—systematic approaches for fusion dataflows remain underexplored. Current SotA methods [1, 24] either focus on DRAM accesses, neglecting on-chip data movement, or address the issue by spilling excess data to higher memory levels [14, 26] without a comprehensive strategy for memory level assignment.

In summary, challenges persist in scheduling frameworks, particularly in operator fusion, tiling strategies, and hardware mapping. FusionFrame addresses these issues by defining a comprehensive scheduling space and proposing a systematic, memory-centric approach to resolve these challenges.

3 Schedule Space Identification

The schedule space for a fusion dataflow typically encompasses three axes: operator fusion, loop tiling, and hardware mapping [14, 27].

- *Operator fusion:* The schedule space should accommodate a wide range of fusion granularity. This encompasses fusing individual operators and enabling hierarchical fusion, combining multiple fused groups to form a complete workload. Figure 1a illustrates this concept.
- *Loop tiling:* The space should encompass various options for tiling dimensions and sizes. This requires the schedule space to be aware of the data layout and offer flexibility in adjusting the tile size for each dimension.

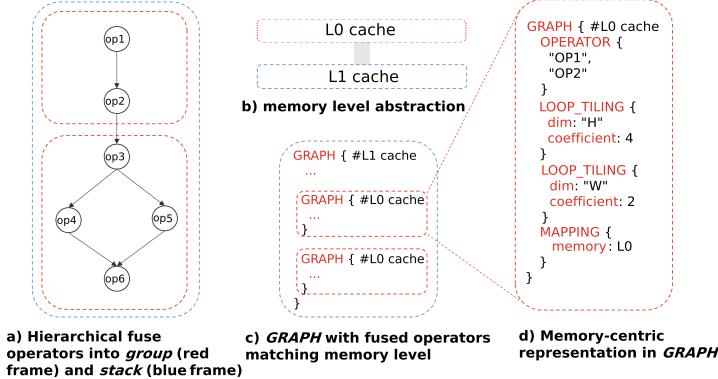


Fig. 1. Illustration of memory-centric representation.

- *Hardware mapping:* The schedule space should support the exploration of diverse hardware architectures with multiple memory levels.

The vast size of this schedule space necessitates a structured approach to represent. In the following subsections, we propose a memory-centric representation for describing the schedule space within FusionFrame. This approach will detail how to represent this complex space effectively and comprehensively for schedule optimization.

3.1 Memory-Centric Representation

Our approach hinges on a critical observation: while mapping tiled for-loops onto various hardware memory levels, nested loops and memory hierarchies are tightly coupled. Each memory level is inherently suited for computations at specific loop nesting levels. To exploit this inherent synergy, we introduce the concept of a *GRAPH* to represent scheduling information specific to a particular memory level. It utilizes a nested structure to accommodate multiple memory levels. For instance, consider hardware with two on-chip memory levels (as illustrated in Fig. 1b). The operators in Fig. 1a can be initially fused onto L0 cache, as highlighted by the red frames. These fused groups (e.g., *op1* and *op2* as one group, and *op3*, *op4*, *op5*, *op6* as the other) are then further fused onto L1 cache, denoted by the blue frames and referred to as a stack of fused operators. We will further elaborate on this memory-centric fusion mechanism in Sect. 4. The distinction between groups and stacks lies in their corresponding memory levels, with each level represented by a separate *GRAPH* (shown in Fig. 1c). This *GRAPH*-based memory-centric representation facilitates hierarchical operator fusion across hardware architectures with any number of memory levels.

Further, the memory-centric representation offers a key advantage: it explicitly assigns memory levels to fused operators, bridging the gap between tiled loops and the hardware’s memory hierarchy. As a result, it enables us to effectively represent not only hierarchical operator fusion (where multiple fused

Table 1. Memory-centric representation primitives.

Primitive		Description
OPERATOR		fused operators in GRAPH
MAPPING	memory	specifies mapped memory level
LOOP_TILING	dim	tile dimension
	coefficient	tile size

groups are combined) but also the mapping of these workloads onto various hardware architectures with multi-level memory systems.

3.2 Schedule Primitives for Memory-Centric Representation

Leveraging the memory-centric representation, FusionFrame defines three fundamental schedule primitives (listed in Table 1) to describe the schedule space comprehensively. These primitives, *OPERATOR*, *LOOP_TILING*, and *MAPPING*, enable the efficient exploration of the space to identify the optimal solution.

OPERATOR: This primitive addresses the fused operators in one *GRAPH*. Each operator is identified using a unique name, and fusion is achieved by elevating intermediate data between these operators. A concrete example is depicted in Fig. 1c, *op1* and *op2* are fused together as a *GRAPH*, this is done by elevating intermediate data between these operators onto L0 cache, same for the next *GRAPH* consisting of *op3*, *op4*, *op5* and *op6*. The hierarchical fusion of these two groups is achieved by elevating the intermediate data in between, i.e., between *op2* and *op3*, to L1 cache.

MAPPING: The *memory* parameter specifies the memory level of the current *GRAPH* it is describing. As illustrated in Fig. 1d, the schedule primitive indicates that the intermediate data between *op1* and *op2* in the current *GRAPH* is mapped to L0 cache.

LOOP_TILING: This primitive defines strategies for reorganizing loops within a specific memory level represented by a *GRAPH*. It operates with two parameters: *dim*, which refers to the tiled dimension (e.g., *H* for height, *W* for width), and *coefficient*, which is the tile size of the selected *dim*.

Figure 1d showcases a detailed schedule representation for fused operators *op1* and *op2*. Here, the tile size in dimension *H* is 4 and in dimension *W* is 2. By allowing the selection of different dimensions for tiling, this primitive enables exploring various schedule space options for various data layouts.

4 Analytical Cost Model

This section describes the FusionFrame analytical cost model, which takes two inputs: the original compute graph and the hardware architecture of the DNN

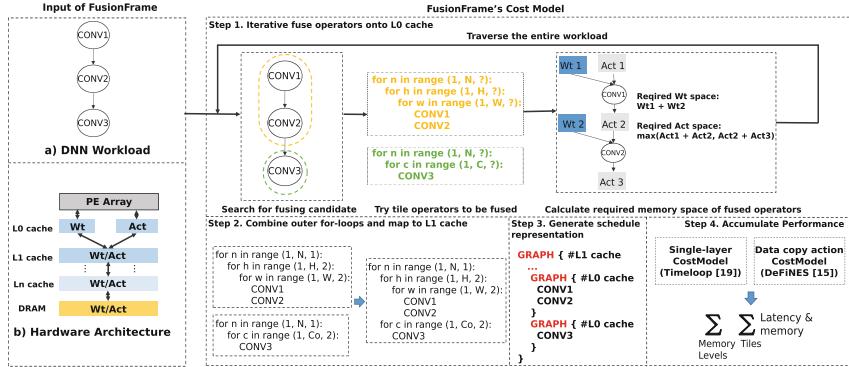


Fig. 2. Illustration of memory-centric fusion mechanism.

accelerator, as illustrated in Fig. 2. The compute graph represents the DNN workload, including sliding-window-based operators (e.g., CONV, POOL) and element-wise operators across various network architectures. The hardware architecture in FusionFrame abstracts the accelerator into PEs and a multi-level memory hierarchy. The PEs support any dimensional array of any size, and the memory hierarchy is abstracted into multiple levels of on-chip memory that store weights (denoted as Wt) and activations (denoted as Act). The cost model explores the schedule space defined in Sect. 3, generates a schedule space representation, and evaluates its performance, including latency and memory accesses on each memory level. This paper focuses on dense workloads, with detailed steps provided below.

4.1 Memory-Centric Fusion Mechanism

FusionFrame employs a memory-centric fusion mechanism, where the compute graph is traversed to fuse operators and map intermediate data between operators onto each level of memory hierarchically. Initially, all operators are fused and tiled to map the inner tiled loops into the innermost, fastest memory (e.g., L0 cache), forming groups of operators. Next, the common outer for-loops of these groups are identified and combined. These combined outer for-loops are then mapped to higher memory levels. Given the capacity of a specified memory level and the compute graph of the entire workload, the operators are fused as shown in Fig. 2.

Starting from the last operator in the compute graph, the fusion process searches for operators to be fused onto L0 cache shown in Fig. 2 step 1. It traverses the entire graph in reverse topological order using a depth-first greedy search algorithm, as described in DeFINES [14]. This approach can be replaced with advanced search algorithms [5, 10, 11] for better performance, and we leave this to future study. Once the operators to be fused are identified, FusionFrame attempts to tile these operators by selecting appropriate tiling dimensions and

sizes. The tiling dimensions are selected in the layout order: first dimension N , then dimensions H and W , and lastly dimension C if large weights are encountered. The selection of tile dimensions and sizes must adhere to the following rules:

- *Operators to be fused must be tiled in the same dimension.* Supported tiling dimensions in FusionFrame are discussed in Sect. 4.2.
- *The total size of tiled intermediate data must fit the specified memory level.* The tiled intermediate data size calculation method is provided in Sect. 4.3.

If FusionFrame cannot find a tile dimension and size that meets the specified rules, the fusion process fails, prompting a new search for other fusible operators. This process continues until the entire compute graph is traversed. Once the fusion on L0 cache is complete, FusionFrame traverses all the fused groups to identify and combine their outer tiled loops, mapping these combined loops onto higher memory levels, such as L1 cache, shown in Fig. 2 step 2. After processing all memory levels, FusionFrame generates a memory-centric representation for the schedule and evaluates the overall performance by summing up the cost of each memory level and each tile for all operators, as shown in Fig. 2 step 3 and step 4, and detailed in Sect. 4.4. The fusion process traverses the entire compute graph once for each memory level, resulting in both time and space complexities of $O(n)$, where n is the number of nodes in the graph. For ResNet-50 [4] with a single batch input, FusionFrame completes the search in under 2 min on a single thread of an Intel Xeon Gold 6142 CPU @ 2.6 GHz.

Figure 3a illustrates the mapping of tiled data across different memory levels. For an output shape of $3 \times 4 \times 4 \times 1$ (NHWC) in DRAM, operators are initially fused onto L0 cache by tiling the N , H , and W dimensions, with tile sizes of 1 for N and 2 for both H and W . These tiles correspond to three nested for-loops. The inner loops over H and W dimensions are mapped to L0 cache with a tile size of $1 \times 2 \times 2 \times 1$ (NHWC). The outer loop of dimension N is mapped to a higher memory level, i.e., L1 cache. Figure 3b illustrates two possible fusion results for a workload of three convolution operators. The left example demonstrates fusing all three operators into a single group, mapping loops over H and W dimensions to L0 cache, and the outermost loop over N dimension to L1 cache. The right example shows the operators fused into two groups, where $CONV1$ and $CONV2$ are tiled in Ho and Wo dimensions and $CONV3$ is tiled in Co dimension. These two groups are mapped to L0 cache, and their outermost loops are combined and mapped to L1 cache.

4.2 Tiling Fused Operators

FusionFrame tiles operators to fit intermediate data into different memory levels, necessitating careful selection of tile dimensions and sizes. It supports tiling all output dimensions of fused operators. Tiling different output dimensions impacts intermediate data differently: tiling in the Ho and Wo dimensions affects intermediate activation size, while tiling in the Co dimension impacts intermediate weight size.

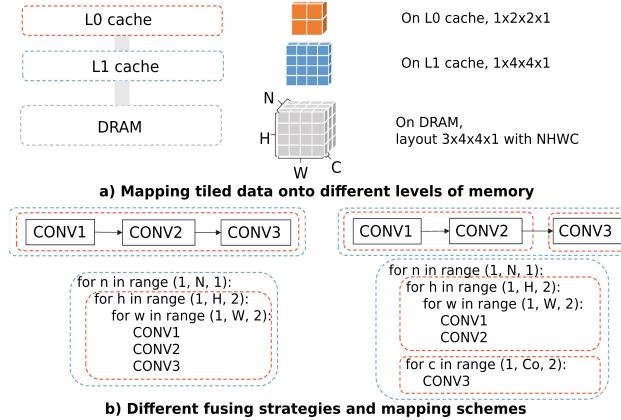


Fig. 3. Illustration of memory-centric fusion mechanism: orange frames represent operators or for-loops mapped to L0 cache, blue frames represent those mapped to L1 cache. (Color figure online)

Figure 4 illustrates tiling along different dimensions. For simplicity, the activation batch size is set to 1. Figure 4a shows a typical convolution layout. Figure 4b presents a typical convolution chain with each operator’s activation (denoted as Act) and weight (denoted as Wt). Figure 4c shows the required data of tiling on the output Ho and Wo dimensions while fusing $CONV1$ and $CONV2$. The activations and weights required to calculate each tile in $Act3$ are highlighted in orange. In this scheme, only activation data are tiled, therefore $Wt1$ and $Wt2$ are residing in L0 cache and tiles of $Act1$, $Act2$, and $Act3$ are loaded from DRAM.

Figure 4d shows another case of tiling along the output channel dimension, i.e., Co . One computation of the innermost loop requires a tile of weight data and the entire input data. This scheme resides input data in L0 cache whereas tiled weight data swaps between on-chip memory and DRAM. This avoids redundant tiled data transfer since there is no overlap between Co tiles but requires gathering all output tiles before proceeding to the next operator. This gathering can be performed on L1 cache in multi-level memory architectures, enabling tiling in the Co dimension and reducing memory pressure for fusing operators with large weights. When tiling is feasible, the largest feasible tile size (T_{max}) fitting into the memory is selected using binary search. The tile size can be further refined between 1 and T_{max} for optimal performance and we leave the fine-grain exploration of tile size as future work.

4.3 Tiled Data Shape Calculation and Memory Allocation

FusionFrame traverses the fused operators along each tiling dimension to calculate the tiled shape of each intermediate data. This traversal starts from the tiled output, establishing the mapping between input and output dimensions for

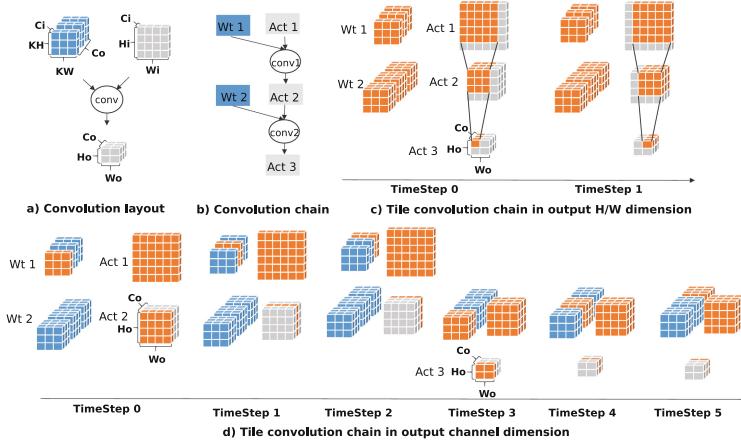


Fig. 4. Illustration of tiling output dimensions, with active tiles marked in orange, denoted as *Act_tile*. (Color figure online)

Table 2. Mapping of input shape and output shape for convolution, with parameters in the equation: S means stride, K means kernel, P means padding, D means dilation.

Dimension	Input(i) vs. Output(o)
N	$N_i = N_o$
H	$H_i = (H_o - 1) \times S_h + K_h - 2 \times P_h + D_h \times (K_h - 1) + 1$
W	$W_i = (W_o - 1) \times S_w + K_w - 2 \times P_w + D_w \times (K_w - 1) + 1$
C	C_i and C_o are not correlated

each operator to deduce the tiled input shape. Supported operators fall into two categories:

- Dimension-Agnostic Operators: These operators, such as element-wise operators, have tiled input shapes identical to their tiled output shapes.
- Dimension-Sensitive Operators: These operators, like sliding-window-based operators (e.g., convolution, pooling), have complex mappings between input and output shapes. Table 2 shows the input-output mapping for convolution, which can be extended to other operators.

Using these mappings, the tiled shape for each intermediate data can be determined. For branches, the data shapes from different branches are merged, as done in DeFiNES [14].

After calculating the tiled shape, the memory required for intermediate data can be allocated. FusionFrame executes fused operators sequentially, simplifying synchronization and optimizing memory usage. Under this execution model, the required activation memory at any point is the sum of the inputs and outputs of

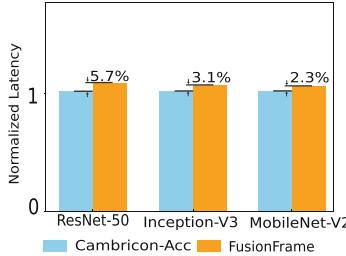


Fig. 5. Validation result against Cambricon-Acc.

the current operator. To compute the maximum memory required, we traverse all fused operators and calculate:

$$M_{required} = \max_{i \in \{1, \dots, n\}} \left(\sum_{j=1}^{m_i} A_{ij} + \sum_{k=1}^{o_i} A_{ik}, \sum_{j=1}^{m_{i+1}} A_{i+1,j} + \sum_{k=1}^{o_{i+1}} A_{i+1,k} \right) \quad (1)$$

where $M_{required}$ represents the total required memory space, n represents the total number of operators in the DNN. m_i represents the number of inputs for operator i , o_i represents the number of outputs for operator i , and A_{ij} and A_{ik} represents the memory footprint of the j -th and k -th inputs for operator i . The function \max ensures that we capture the worst-case scenario for memory usage at any point in execution. The inner expression iterates through each operator and calculates two potential memory requirements. For a simple workload example in Fig. 4b, the total required activation memory space in this convolution chain is $\max(Act1_tile + Act2_tile, Act2_tile + Act3_tile)$.

4.4 Performance Evaluation

FusionFrame calculates the cost of each fused operator for each memory level using a memory-centric approach. For L0 cache, operators perform data movement between L0 cache and PE, and computation in PE, evaluated using the Timeloop [18] cost model. FusionFrame can also integrate other SotA cost models, such as ZigZag [15] and MAESTRO [13]. For higher memory levels, such as L1 cache and DRAM, only data copy actions are considered. FusionFrame employs the data copy action cost model from DeFiNES [14] to measure the latency of moving weights and activations between memory levels, accounting for total data movement size and memory bandwidth, as well as potential non-idealities like memory port conflicts. The total cost of the fusion program is then calculated by summing the costs at each memory level, including both computation and data movement as shown in Fig. 2 step 4.

In summary, FusionFrame uses a memory-centric approach to optimize operator fusion across multiple memory levels. It calculates tile dimensions and sizes

Table 3. Different targeting hardware specifications.

Parameters	Cambricon-Acc	TPU-like	Tesla-NPU-like
Spatial Unrolling (1024 MACs)	Ci 64 Co 64	Ci 32 Co 32	Ci 32 Wo 8 Ho 4
Register per MAC	W&O: 2 B	W: 128 B; O: 1 KB	W: 1B; O: 2 B
L0 Cache Size	I&O: 64 KB; W: 768 KB	I&O&W: 64 KB	W: 64 KB; I&O: 64 KB
L1 Cache Size	I&O&W: 2 MB	I&O&W: 2 MB	W: 1 MB; I&O: 1 MB

to fit within these levels, ensuring efficient use of the on-chip memory hierarchy. The compute and data movement costs are evaluated and accumulated to determine the final performance cost.

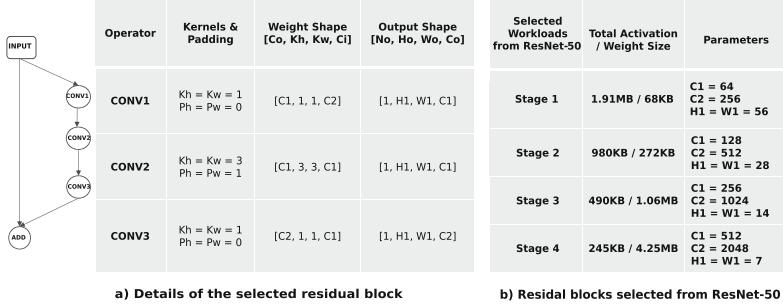
5 Validation

FusionFrame uses a validated cost model Timeloop [18] to evaluate performance, however, to ensure precise modeling of the memory-centric fusion mechanism, FusionFrame is validated against a taped-out chip Cambricon-Acc [3] using the end-to-end performance of several complete DNNs. This chip is taped out using 65nm TSMC technology with an overall area of 56.24 mm^2 . Cambricon-Acc performs DNN computations using SIMD-like instructions on scratchpad memories, which can be modeled as inner-most loops mapped to L0 cache. We set FusionFrame’s hardware parameters to align with Cambricon-Acc’s specifications, as shown in Table 3. The validation results, shown in Fig. 5, demonstrate that FusionFrame’s evaluation match within 6% for three different workloads, which will be used in further case studies as detailed in Sect. 6. The variation arises from DRAM access modeling inaccuracies is mainly caused by non-idealities such as DRAM access granularity and fixed access overhead when accessing small tiled data sizes on DRAM.

6 Case Studies

6.1 Experimental Setup

Benchmarks. We select four identical residual blocks with different parameters from ResNet50 [4] as benchmarks. The total weight and activation size of each selected workload are shown in Fig. 6a. These workloads range from activation-dominant (large activations and small weights) to weight-dominant (small activations and large weights), with activation size decreasing and weight size increasing. Figure 6b provides details of each selected residual block (denoted as *stage*), depicting the residual architecture and corresponding parameters used in the experiments. Different stages share identical network architectures but differ in feature map sizes and output channels for each convolution. To evaluate FusionFrame’s overall network performance and applicability, we selected



a) Details of the selected residual block

b) Residual blocks selected from ResNet-50

Fig. 6. Elaboration of selected residual-block workloads with the same architecture and different parameters in ResNet-50.

three different DNNs with distinct architectures and tested them on three different hardware platforms listed in Table 3. The selected DNNs are ResNet50 [4], MobileNetV2 [6], and InceptionV3 [21].

Hardware Platforms. We target three hardware platforms, i.e., Cambricon-Acc [3], TPU-like [8] and Tesla-NPU-like [22]. These architectures are aligned with the same number of MACs and bandwidth between different memory levels for fair comparison, and their unique on-chip memory capacity, memory levels, and spatial unroll parameters are kept. We set the data width to 8-bit, the DRAM access bandwidth to 64-bit/cycle, and the on-chip memory bandwidth to twice the DRAM access bandwidth to mimic the on-off-chip communication bottleneck. Detailed hardware specifications are listed in Table 3.

6.2 Case Study 1: The Impact of L1 Cache over Different Fusion Dataflows and Tiling Dimensions

Many accelerators are designed with L1 cache [2,8,22] to improve data locality and latency for DNN workloads. The benefits of L1 cache and other memory hierarchies have been explored in single-operator-level experiments [13,15]. However, the impact of L1 cache on different fusion dataflows has not been addressed. In this case study, we explore the impact of L1 cache under different tiling strategies on fusion dataflows. We focus on stage 2, detailed in Fig. 6b, as the target fusion workload, and use Cambricon-Acc as the target hardware platform. To investigate architectures with and without L1 cache, we add a 2MB L1 cache to the original Cambricon-Acc, creating a two-level on-chip memory hierarchy accelerator. We use *Acc-1L* and *Acc-2L* to denote Cambricon-Acc with one and two memory levels, respectively.

Tiling in Ho/Wo Dimension. We first examine tiling in the *Ho* and *Wo* dimensions on *Acc-1L* and *Acc-2L* to study the impact of memory hierarchy

on data movement within fusion dataflows. The fusion strategy for mapping this workload to Acc-2L is shown in Fig. 7a. This strategy fuses all the operators into a single group, so L1 cache only affects the input, weight, and output of the group. Initially, we assess the impact on the input data independently, then extend the analysis to the output and weight. To achieve this, we bypass L1 cache for the output and weight, moving them directly between L0 cache and DRAM.

In this exploration, we sweep the tile size from 1×1 to 6×6 to explore how L1 cache affects input data movement. As shown in Fig. 7, one batch of input is loaded into L1 cache at a time for *Acc-2L*, where input is tiled in H_o and W_o dimensions and mapped to L1 cache. In contrast, for *Acc-1L*, input is tiled on DRAM, resulting in significantly more DRAM accesses due to overlapped data being fetched from DRAM. Additionally, for *Acc-2L*, L1 cache accesses occur because tiled input is fetched from L1 cache to L0 cache. This access amount is the total size of the tiled input, including the overlapped data introduced by tiling in H_o and W_o dimensions. Both DRAM access overheads and L1 cache access decrease as tile size increases due to a reduction in the number of tiles and, consequently, the overlapped data movement between tiles.

Moreover, the total data movement for *Acc-2L* is larger than that for *Acc-1L*, but shows lower latency for tile sizes smaller than 4×4 . This is because the on-chip memory bandwidth is greater than the DRAM access bandwidth. However, as the tile size increases, the overlapped data movement decreases, reducing the benefit of loading input through L1 cache. Therefore, careful consideration is needed to determine whether input should utilize L1 cache based on the available on-chip memory bandwidth. For input passing through L1 cache, the trade-off lies between reducing DRAM accesses and increasing on-chip memory accesses. As for output and weight, they do not suffer from the overlapped data movement issue, so passing them through L1 cache cannot reduce DRAM accesses but increase on-chip memory accesses. Therefore, the better solution for latency is to bypass L1 cache for output and weight.

Tiling in Co Dimension. We further examine the impact of L1 cache when tiling the output channel dimension (Co). The representation is illustrated in Fig. 7d, the workload is divided into three groups, each tiled in the Co dimension due to the weight size exceeding L1 cache capacity. Tiling in the Co dimension requires data concatenation after each convolution operator, as discussed in Section 4.2. However, dimension-agnostic operators like *ADD* can be fused with preceding convolutions tiled in the Co dimension. For the first group containing *CONV1*, the key difference between *Acc-1L* and *Acc-2L* is whether the output is concatenated in L1 cache. Storing the output of *CONV1* in L1 cache reduces DRAM accesses for *Acc-2L* by 21.5%. However, loading the input of *CONV1* through L1 cache increases total data movement and latency, because the input is not tiled in H_o and W_o dimensions, as discussed in case study 1. For groups containing *CONV2* and *CONV3 + ADD*, the total data movement is the same between *Acc-2L* and *Acc-1L*. However, DRAM accesses are reduced by 1.36x and

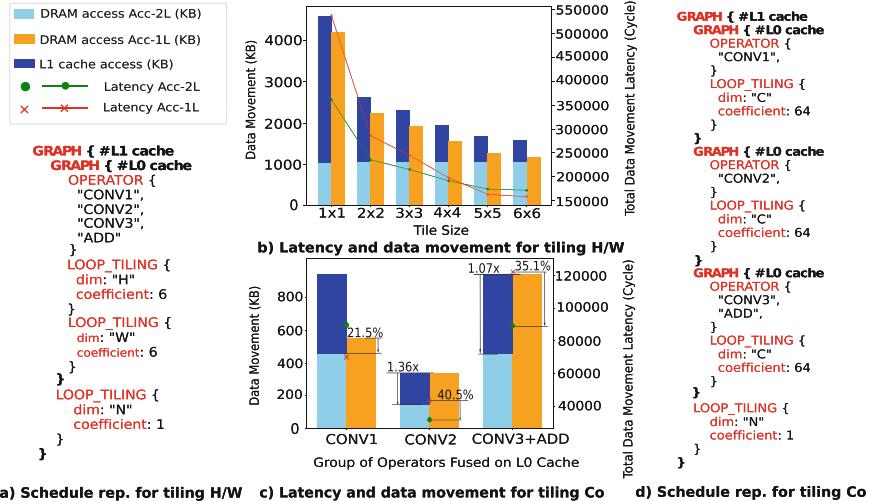


Fig. 7. The memory-centric representation of tiling different dimensions and experimental results on each memory level accesses and total data movement latency under different tile sizes. The representation for tiling H_o and W_o shows that all operators are fused on L0 cache with a largest feasible tile size of 6×6 , and the batch dimension is mapped to L1 cache.

1.07x, and total data movement latency is thus reduced by 40.5% and 35.1% for these two groups respectively, due to intermediate storage of inputs and outputs in L1 cache.

In conclusion, for a stack of fused operators, whether the input should utilize L1 cache should be carefully determined based on the on-chip memory bandwidth and the amount of overlapped data from tiling in H_o and W_o dimensions. Under the output tiling scheme, the output and weight data should bypass L1 cache to avoid detour. Additionally, L1 cache is effective for storing intermediate data between operator groups, reducing DRAM accesses and data movement latency for these intermediates. FusionFrame facilitates analysis of various fusion dataflows and tiling strategies through its flexible memory-centric representation and memory-centric fusion mechanism.

6.3 Case Study 2: SotA Comparison on Different Fusion Dataflows

In this case study, we compare three types of fusion dataflows: layer-by-layer, memory-centric based in FusionFrame and the fusion dataflow used in DeFiNES [14] and ACCO [26]. We implement the fusion dataflow used in DeFiNES [14] within FusionFrame, namely DeFiNES-like, leveraging FusionFrame's memory-centric abstraction. This implementation retains DeFiNES's key features of the fusion dataflow: input and output data are prioritized to use L0 cache, spilling to higher memory levels only when necessary, provided

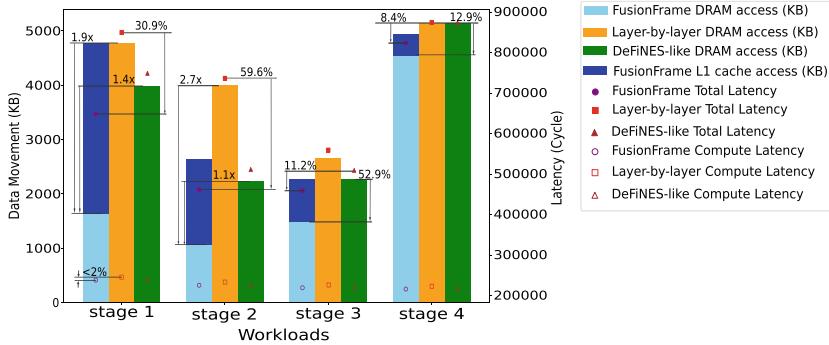


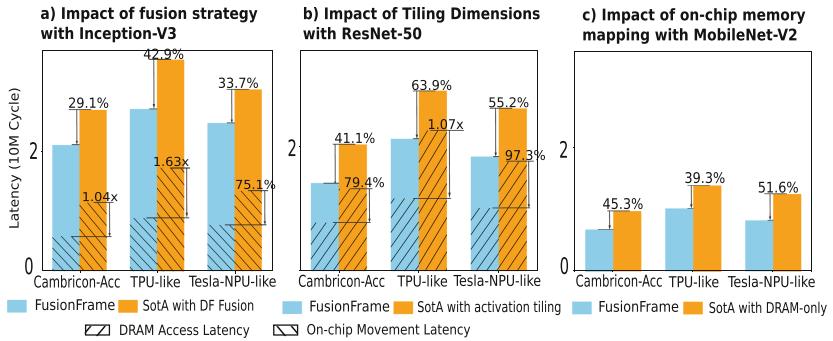
Fig. 8. Comparison of fusion dataflows: layer-by-layer, DeFiNES-like, and FusionFrame on Acc-2L architecture. For stages 1 and 2, Ho and Wo are tiled to 2×2 ; for stages 3 and 4, Co are tiled to 64.

weight fits in L0 cache. The execution process for this dataflow involves loading and storing all types of data directly from and to DRAM, bypassing L1 cache, while weight remains in L0 cache throughout the computation. In contrast, the layer-by-layer dataflow evaluates each operator one at a time completely, with intermediate feature maps passed between operators using the lowest memory level they fit in. FusionFrame adopts the fusion dataflow with better performance identified in case study 1, which involves loading input data of the fused stack through L1 cache for Ho and Wo tiling and other types of data move directly between L0 cache and DRAM. Intermediate feature maps between different groups are passed through L1 cache.

We evaluate these fusion dataflows using the four residual blocks introduced in Fig. 6 on the Acc-2L architecture. In this exploration, we align different fusion mechanisms with the same tile size to illustrate the varying memory access patterns for each fusion mechanism. The results, shown in Fig. 8, detail the different dataflows. The variation of different dataflows' compute latency is less than 2%, as the fusion pattern does not significantly affect computation. Therefore, our analysis primarily focuses on data movement and total latency. For the first two stages, FusionFrame and DeFiNES-like fuse all operators in L0 cache with Ho and Wo tiling, as the total weight fits within L0 cache capacity. FusionFrame dataflows reduce DRAM accesses by $1.9\times$ and $2.7\times$, respectively, and reduce latency by 30.9% and 59.6% compared to the layer-by-layer dataflow. In these cases, FusionFrame and DeFiNES-like fusion dataflows adopt similar strategies, except FusionFrame benefits from loading input through L1 cache, further decreasing DRAM access by $1.4\times$ and $1.1\times$ as the analysis is detailed in case study 1. In stages 3 and 4, the weight size of fused convolutions exceeds L0 cache capacity, leading to different behaviours for FusionFrame and DeFiNES-like. FusionFrame caches the intermediate data between fused groups in L1 cache, whereas DeFiNES-like cannot fuse convolutions when their weight size exceeds L0 cache capacity. Consequently, FusionFrame reduces latency by

Table 4. Related SotA schedule frameworks comparison.

Framework	Complete Tiling Dimensions Support	Multi-level Memory Mapping	Hardware-aware Fusion
DeFiNES [14]	×	✓	×
Genetic-A [12]	×	✓	×
ACCO [26]	×	✓	×
ConvFusion [24]	×	×	×
FusionFrame (ours)	✓	✓	✓

**Fig. 9.** SotA comparison on the performance impact of different scheduling dimensions.

9.8% and DRAM accesses 32.9% in average compared to DeFiNES-like in these two cases. In stage 3, the DeFiNES-like dataflow fuses *ADD* with *CONV3* because the weight of *CONV3* fits within L0 cache capacity. However, in stage 4, *CONV3*'s weight exceeds L0 cache capacity, preventing fusion with *ADD* and resulting in a 12.9% increase in total data movement compared to FusionFrame.

In summary, FusionFrame consistently outperforms DeFiNES-like and layer-by-layer dataflows by leveraging L1 cache for intermediate data storage, significantly reducing DRAM accesses by 80.2% and latency by 14.3% on average under the same fusion depth. The advantages stem from two key factors: loading input through L1 cache to minimize DRAM accesses and caching intermediate data between operator groups in L1 cache. This underscores the importance of flexible fusion dataflows representation and hierarchical fusion of operators on hardware with multi-level memory hierarchies.

6.4 Case Study 3: SotA Comparison on Performance Impact of Scheduling Dimensions

This case study examines the impact of three scheduling dimensions—operator fusion, loop tiling, and hardware mapping—as introduced in Sect. 3 by compar-

ing them against SotA implementations across different workloads on three DNN accelerator platforms shown in Table 3.

For operator fusion, SotA methods such as DeFiNES [14], ACCO [26], and others [12, 24] utilize a depth-first fusion strategy (denoted as *DF*) without optimizing tile sizes, instead relying on user-provided tile sizes. This approach results in sub-optimal fusion behavior and under-utilization of hardware resources. By implementing the *DF* strategy with a 6×6 tile size for each fused stack in the *Ho* and *Wo* dimensions, as suggested by DeFiNES [14], and comparing its latency with FusionFrame using Inception-V3 [21], we found that the *DF* fusion dataflow results in higher on-chip memory accesses, exceeding FusionFrame by an average of 1.14x (Fig. 9a). This is because the user-provided tile size does not align with the hardware specifications, leading to the allocation of intermediate data on L1 cache and causing redundant on-chip data movements. Due to the overhead from on-chip memory access, FusionFrame’s total latency is 35.2% lower on average than the *DF* dataflow, demonstrating the advantage of FusionFrame’s hardware-aware fusion strategy that automatically determines tile sizes based on hardware specifications.

For loop tiling, SotA methods such as Genetic-A [12], ACCO [26], and others [1, 9] focus on activation tiling, where performance is limited by weight size and dependent on large L0 caches. We implemented an activation-tiling-only dataflow and compared it with FusionFrame’s complete output tiling scheme using ResNet-50 [4]. The experiment results (Fig. 9b) show that the DRAM access latency of the activation-tiling-only dataflow exceeds FusionFrame by 52.7% on average. This issue is particularly evident in TPU-like architectures, which exhibit a 1.07x DRAM access latency overhead and 63.9% total latency overhead. This is because the activation-tiling-only dataflow deals with layers in ResNet-50 that exceed L0 weight cache size in a layer-wise manner, making DRAM access dominant in total latency, especially for architectures with small L0 caches such as TPU-like architectures.

For hardware mapping, SotA works such as ConvFusion [24] and Optimus [1] aims to reduce DRAM accesses but overlook on-chip memory accesses. The impact of on-chip memory accesses has been addressed in case study 1, and we evaluate a fusion dataflow optimized for DRAM accesses using MobileNet-V2 [6] and compare it with FusionFrame. The result is shown in Fig. 9c, FusionFrame achieves an average latency improvement of 45.4%, highlighting the importance of balancing on-chip and off-chip accesses.

Furthermore, comparing across different workloads, Cambricon-Acc exhibited the lowest latency, followed by Tesla-NPU-like, and TPU-like architectures. This trend is attributed to Cambricon-Acc’s larger on-chip memory, which better supports fusion dataflow latency optimization.

This case study addresses the limitations of current StoA methods in operator fusion, loop tiling, and hardware mapping. Our evaluation shows that FusionFrame significantly outperforms these methods by optimizing on-chip memory accesses and tile sizes, resulting in lower latency. The comparison of state-of-the-art works is shown in Table 4.

7 Conclusion

In this paper, we introduced FusionFrame, a comprehensive fusion dataflow framework designed to address the challenges in scheduling DNNs on DNN accelerators. FusionFrame characterizes the scheduling space through three dimensions: operator fusion, loop tiling, and hardware mapping, using a memory-centric approach. It provides an accurate analytical cost model to enable flexible scheduling for accelerators with multi-level memory hierarchies, facilitating the exploration of various fusion dataflows. The case studies showed that FusionFrame is able to navigate the trade-offs between different aspects of the schedule space. Our evaluation of three representative DNNs showed that our memory-centric fusion strategy outperforms SotA by up to 63.9% in latency reduction.

Acknowledgments. This research is partially supported by the National Research Foundation, Singapore under its Competitive Research Program Award NRF-CRP23-2019-0003.

References

1. Cai, X., Wang, Y., Zhang, L.: Optimus: an operator fusion framework for deep neural networks. ACM Trans. Embed. Comput. Syst. (2022)
2. Chen, Y., Chen, T., Xu, Z., Sun, N., Temam, O.: Diannao family: energy-efficient hardware accelerators for machine learning. Commun. ACM (2016)
3. Chen, Y., et al.: An instruction set architecture for machine learning. ACM Trans. Comput. Syst. **36**(3) (2019). <https://doi.org/10.1145/3331469>
4. 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)
5. Hegde, K., Tsai, P.A., Huang, S., Chandra, V., Parashar, A., Fletcher, C.W.: Mind mappings: enabling efficient algorithm-accelerator mapping space search. In: Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 943–958 (2021)
6. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
7. Jeong, G., et al.: Union: a unified hw-sw co-design ecosystem in mlir for evaluating tensor operations on spatial accelerators. In: 2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT) (2021)
8. Jouppi, N.P., et al.: In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 1–12 (2017)
9. Kao, S.C., Huang, X., Krishna, T.: Dnnfuser: generative pre-trained transformer as a generalized mapper for layer fusion in dnn accelerators. arXiv preprint [arXiv:2201.11218](https://arxiv.org/abs/2201.11218) (2022)
10. Kao, S.C., Jeong, G., Krishna, T.: Confucius: autonomous hardware resource assignment for dnn accelerators using reinforcement learning. In: 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO) (2020)
11. Kao, S.C., Krishna, T.: Magma: an optimization framework for mapping multiple dnns on multiple accelerator cores. In: 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp. 814–830. IEEE (2022)

12. Karl, S., Symons, A., Fasfous, N., Verhelst, M.: Genetic algorithm-based framework for layer-fused scheduling of multiple dnns on multi-core systems. In: 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2023)
13. Kwon, H., Chatarasi, P., Sarkar, V., Krishna, T., Pellauer, M., Parashar, A.: Maestro: a data-centric approach to understand reuse, performance, and hardware cost of dnn mappings. *IEEE Micro* **40**(3), 20–29 (2020)
14. Mei, L., Goetschalckx, K., Symons, A., Verhelst, M.: Defines: enabling fast exploration of the depth-first scheduling space for dnn accelerators through analytical modeling. In: 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp. 570–583. IEEE (2023)
15. Mei, L., Houshmand, P., Jain, V., Giraldo, S., Verhelst, M.: Zigzag: enlarging joint architecture-mapping design space exploration for DNN accelerators. *IEEE Trans. Comput.* **70**(8), 1160–1174 (2021)
16. Mei, L., Liu, H., Wu, T., Sumbul, H.E., Verhelst, M., Beigne, E.: A uniform latency model for dnn accelerators with diverse architectures and dataflows. In: 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 220–225. IEEE (2022)
17. Moreira, O., Popp, M., Schulz, C.: Graph partitioning with acyclicity constraints. arXiv preprint [arXiv:1704.00705](https://arxiv.org/abs/1704.00705) (2017)
18. Parashar, A., et al.: Timeloop: a systematic approach to dnn accelerator evaluation. In: 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 304–315. IEEE (2019)
19. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning, pp. 28492–28518. PMLR (2023)
20. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015. IEEE Computer Society (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
22. Talpes, E., et al.: Compute solution for tesla’s full self-driving computer. *IEEE Micro* **40**(2), 25–35 (2020)
23. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
24. Waeijen, L., Sioutas, S., Peemen, M., Lindwer, M., Corporaal, H.: Convfusion: a model for layer fusion in convolutional neural networks. *IEEE Access* **9**, 168245–168267 (2021)
25. Xing, Y., et al.: Dnnvm: end-to-end compiler leveraging operation fusion on fpga-based cnn accelerators. In: Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 187–188 (2019)
26. Yin, J., Mei, L., Guntoro, A., Verhelst, M.: ACCO: automated causal CNN scheduling optimizer for real-time edge accelerators. In: IEEE ICCD (2023)
27. Zheng, S., et al.: Tileflow: a framework for modeling fusion dataflow via tree-based analysis. In: Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 1271–1288 (2023)

Author Index

B

- Bang, L. K. 200
Bang, N. H. 200

C

- Cai, Hailang 57
Cao, Ligang 78, 249
Cao, Tuo 179
Cesarano, Antonio 217
Chen, Chao 141
Chen, Chen 231
Chen, Jialuo 57
Chen, Yan 179
Cheng, Guang 57
Chi, Huanhuan 231
Cui, Lei 98

D

- Deng, Yu 12
Du, Yanning 268

F

- Fan, Wenjie 159
Fang, Qi 12
Feng, Xia 159

G

- Gao, Jianhua 47
Gong, Chunye 26
Guo, Jingjing 159
Guo, Shuaizhe 47
Guo, Weihao 78, 249

H

- He, Ying 159
Hung, N. N. 200

J

- Ji, Weixing 47
Jiang, Linshan 315

K

- Khanh, V. H. 200
Khoa, T. D. 200

L

- Lan, Huiying 315
Lee, Ickjai 141
Lee, Kyungmi 141
Li, Qiang 282
Li, Qingru 1
Li, Youmeng 12
Liang, Yaling 249
Liao, Ian 282
Liu, Hengzhu 231
Liu, Jie 78, 249
Liu, Mengxuan 122
Liu, Tao 108, 122
Liu, Xiang 315
Liu, Xiaodong 249
Liu, Zhiquan 159
Luo, Gangyi 179

M

- Ma, Di 26
Ma, Jianfeng 159
Malandrino, Ornella 217
Miquelina, Nuno 298

N

- Nghiem, P. T. 200
Nogueira, Vítor Beires 298

O

- Ong, Kok-Leong 141

P

- Peng, Muchun 249

Q

- Qi, Ji 179
 Qian, Zhuzhong 179
 Qu, Youyang 98
 Quaresma, Paulo 298

R

- Ren, Wei 37
 Ren, Xuhao 122

S

- Santos, Daniel 298
 Schmidt, Daniela 298
 Sessa, Maria Rosaria 217
 Shi, Haoyi 1
 Shi, Yongzhen 249
 Sun, Shouyue 98

T

- Triet, N. M. 200
 Trinh, P. D. 200

W

- Wang, Changguang 1
 Wang, Fangwei 1
 Wang, Libo 159
 Wang, Qinglin 78, 249
 Wang, Yajie 108, 122
 Wang, Yichuan 268
 Wang, Yizhuo 47
 Wang, Zehao 179
 Wang, Zhoukai 268
 Wei, Shengjie 179
 Wen, Sheng 282
 Wu, Feng 98
 Wu, Huishu 108, 122

- Wu, Jiayun 37
 Wu, Liwen 98

X

- Xia, Rui 78
 Xiang, Qiao 57
 Xiang, Yang 282
 Xiao, Tiaojie 26
 Xiong, Ping 231
 Xu, Jingdong 57
 Xu, Jinnan 268
 Xu, Yuwei 57
 Xuan, Haojun 108

Y

- Yang, Jiaxun 98
 Yang, Shun 78, 249
 Yao, Shaowen 98

Z

- Zhang, Bonan 141
 Zhang, Chuan 108, 122
 Zhang, Haofei 12
 Zhang, Jin 26
 Zhang, Xianchao 37
 Zhang, Xiang 26
 Zhang, Yaling 268
 Zhang, Yuxiang 47
 Zhao, Dongmei 1
 Zheng, Liutao 315
 Zheng, Xianghan 37
 Zhou, Jincheng 26
 Zhou, Xuehai 315
 Zhu, Liehuang 108, 122
 Zhu, Yingjie 179