An International ICT Journal
Featuring Industry-University-Institute Cooperation

ISSN 1673-5188
CN 34-1294/TN

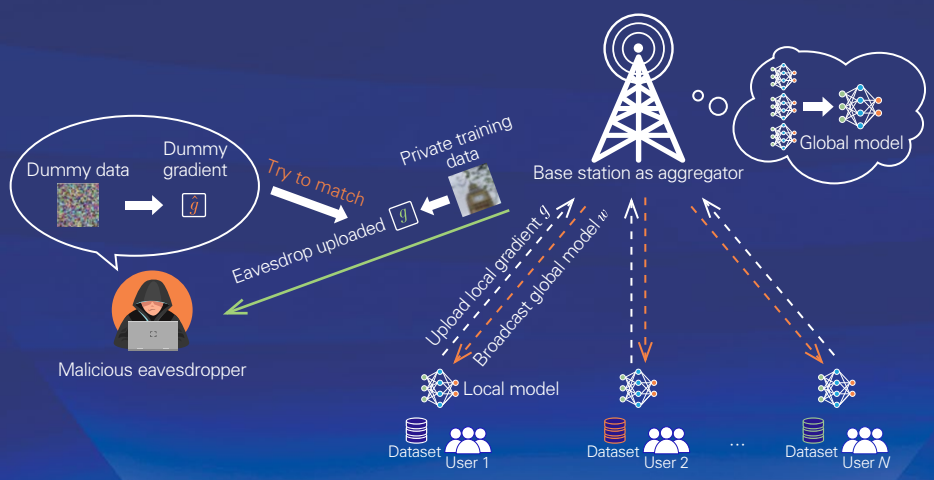
ZTE COMMUNICATIONS

中兴通讯技术(英文版)

http://zte.magtechjournal.com

March 2023, Vol. 21 No. 1

Special Topic:
Federated Learning over Wireless Networks



(See Fig. 1 on P. 48)

ISSN 1673-5188

The 9th Editorial Board of ZTE Communications

CONTENTS

ZTE COMMUNICATIONS
March 2023 Vol. 21 No. 1 (Issue 82)

Special Topic ▶ Federated Learning over Wireless Networks

Research Paper ▶

Serial parameters: CN 34–1294/TN*2003*q*16*94*en*P*¥20.00*2200*11*2023-03

ZTE COMMUNICATIONS

ZTE Communications is a peer-reviewed international ICT journal (CN 34-1294/TN and ISSN 1673-5188) featuring industry-university-institute cooperation. It is published quarterly as a printed publication and can also be freely accessed at https://zte.magtechjournal.com.

ZTE Communications was founded in 2003 and has a readership of more than 100 000. The English version is distributed to universities, colleges, and research institutes in more than 100 countries. The journal covers a wide range of topics in the field of ICT. The Editorial Board members are distinguished domestic and international experts. The journal has become an integrated forum for university academics and industry researchers from around the world.

The journal uses ScholarOne Manuscripts, a professional web-based manuscript submission and peer-review tracking system. Authors must submit manuscripts electronically to https://mc03.manuscriptcentral.com/ztecom.

2023 Special Topics in *ZTE Communications*

Issue 1: Federated Learning over Wireless Networks
CUI Shuguang, The Chinese University of Hong Kong, Shenzhen
YIN Changchuan, Beijing University of Posts and Telecommunications
ZHU Guangxu, Shenzhen Research Institute of Big Data

Issue 2: Evolution of AI Enabled Wireless Network
WANG Ling, Northwestern Polytechnical University
GAO Yin, ZTE Corporation

Issue 3: Reinforcement Learning and Intelligent Decision
GAO Yang, Nanjing University
FENG Yanghe, National University of Defense Technology

Issue 4: 3D Point Cloud Processing and Applications
SUN Huifang, Pengcheng Laboratory
LI Ge, Peking University
CHEN Siheng, Shanghai Jiao Tong University
LI Li, University of Science and Technology of China
GAO Wei, Peking University

Guest Editorial

Special Topic on Federated Learning over Wireless Networks

Guest Editors


CUI Shuguang


YIN Changchuan


ZHU Guangxu

Federated learning has revolutionized the way we approach machine learning by enabling multiple edge devices to collaboratively learn a shared machine learning model without the need for centralized data collection. Such a new machine learning paradigm has gained significant attention in recent years due to its ability to address privacy and security concerns associated with centralized learning, as well as its potential to reduce communication overhead and improve scalability. Deploying cross-device federated learning at the network edge over wireless networks has further extended its potential due to the close proximity to the gigantic number of mobile data and computing power provided by the surging number of Internet of Things (IoT) devices, and is expected to breed new intelligent applications that demand delay-sensitive and mission-critical services, such as smart industry, auto-driving, and metaverse. Despite its great promise, the successful deployment of federated learning over wireless networks has also presented its own unique set of challenges, including network heterogeneity, communication delays, and unreliable connections.

In this special issue, a series of articles are presented to address the aforementioned challenges and propose innovative solutions to enabling federated learning over wireless networks. These articles cover a wide range of topics, including wireless communication protocols, optimization algorithms, security and privacy concerns, network architecture designs, and the application of federated learning in IoT and 5G networks. The call-for-papers of this special issue have brought excellent submissions in both quality and quantity. After two-round reviews, five excellent papers have been selected for publication in this special issue which is organized as follows.

The first paper titled "Adaptive Retransmission Design for Wireless Federated Edge Learning" proposes a novel retransmission scheme for wireless federated edge learning (FEEL). The conventional retransmission schemes for wireless systems, which aim to maximize the system throughput or minimize the packet error rate, are not suitable for the FEEL system. The proposed scheme makes a tradeoff between model training accuracy and retransmission latency, with a retransmission device selection criterion designed based on the channel condition, the number of local data, and the importance of model update. Additionally, the air interface signaling is designed to facilitate the implementation of the proposed scheme in practical scenarios. Simulation experiments validate the effectiveness of the proposed retransmission scheme.

The second paper titled "Reliable and Privacy-Preserving Federated Learning with Anomalous Users" proposes a reliable and privacy-preserving federated learning scheme named RPPFL, based on a single-cloud model. The scheme addresses the issue of anomalous users holding low-quality data, which may reduce the accuracy of trained models. The proposed approach identifies the user's reliability and thereby decreases the impact of anomalous users, based on the truth discovery technique. The additively homomorphic cryptosystem is utilized to provide comprehensive privacy preservation (user's local gradient privacy and reliability privacy). Rigorous theoretical analysis shows the security of RPPFL, and extensive experiments based on open datasets demonstrate that RPPEL compares favorably with existing works in terms of efficiency and accuracy.

The third paper titled "RIS-Assisted Federated Learning in Multi-Cell Wireless Networks" proposes a reconfigurable intelligent surface (RIS) -assisted AirComp-based federated learning (FL) in multi-cell networks. The proposed system enhances the poor user signal caused by channel fading, especially for the device at the cell edge, and reduces inter-cell in-

DOI:10.12142/ZTECOM.202301001
Citation (IEEE Format): S. G. Cui, C. C. Yin, and G. X. Zhu, "Editorial: federated learning over wireless networks," *ZTE Communications*, vol. 21, no. 1, pp. 1–2, Mar. 2023. doi: 10.12142/ZTECOM.202301001.

terference. The convergence of FL in the proposed system is analyzed, and the optimality gap for FL is derived. To minimize the optimality gap, the paper formulates a joint uplink and downlink optimization problem, which is then divided into two separable nonconvex subproblems. Following the successive convex approximation (SCA) method, the paper first approximates the nonconvex term to a linear form, and then alternately optimizes the beamforming vector and phase-shift matrix for each cell. Simulation results demonstrate the advantages of deploying a RIS in multi-cell networks, and the proposed system significantly improves the performance of FL.

The fourth paper titled "Hierarchical Federated Learning: Architecture, Challenges, and Its Implementation in Vehicular Networks" discusses hierarchical federated learning (HFL) and its implementation in vehicular networks. HFL, with a cloud-edge-client hierarchy, can leverage the large coverage of cloud servers and the low transmission latency of the edge server. The limited number of participants in vehicular networks and vehicle mobility degrades the performance of FL training, and HFL is promising in vehicular networks due to its lower latency, wider coverage, and more participants. The paper clarifies new issues in HFL, reviews several existing solutions, introduces some typical use cases in vehicular networks, and discusses the initial efforts on implementing HFL in vehicular networks.

The fifth paper titled "Secure Federated Learning over Wireless Communication Networks with Model Compression" addresses the vulnerability of FL to gradient leakage attacks. A method is proposed to compress the model size to reduce the leakage risk and enhance the efficiency of FL. Specifically, this paper presents a new scheme that applies low-rank matrix approximation to compress the model and uses a secure matrix factorization technique to recover the original model. Experiments showed that the proposed method achieved better accuracy and security compared with the state-of-the-art methods.

To conclude, it is hoped that this special issue will serve as a valuable resource for researchers, practitioners, and students who are interested in federated learning over wireless networks. We also hope that it will inspire further research in this field, leading to new and innovative solutions that will drive the evolution of machine learning. Finally, we would like to express our sincere gratitude to all the authors, reviewers, and editorial staff who have contributed to the success of this special issue. Hopefully, the articles in this special issue are both insightful and informative for prospective readers in the field.

Biographies

CUI Shuguang received his PhD in electrical engineering from Stanford University, USA in 2005. Afterward, he has been working as an assistant, associate, full, Chair Professor in electrical and computer engineering at the University of Arizona (USA), Texas A&M University (USA), UC Davis (USA), and CUHK-Shenzhen (China), respectively. He has also served as the Executive Dean for the School of Science and Engineering at CUHK-Shenzhen and the Executive Vice Director at Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, IoT system design, energy harvesting based communication system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the World's Most Influential Scientific Minds by ScienceWatch in 2014. He was the recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chair for many IEEE conferences. He has also been serving as the editor-in-chief for *IEEE Transactions on Mobile Computing*, area editor for *IEEE Signal Processing Magazine*, and associate editor for *IEEE Transactions on Big Data*, *IEEE Transactions on Signal Processing*, *IEEE JSAC Series on Green Communications and Networking*, and *IEEE Transactions on Wireless Communications*. He has been an elected member of the IEEE Signal Processing Society SPCOM Technical Committee (2009-2014) and the elected chair of IEEE ComSoc Wireless Technical Committee (2017-2018). He is a member of the Steering Committee for *IEEE Transactions on Big Data* and the Chair of the Steering Committee for *IEEE Transactions on Cognitive Communications and Networking*. He was also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE Fellow in 2013, an IEEE ComSoc Distinguished Lecturer in 2014, and an IEEE VT Society Distinguished Lecturer in 2019. He won the IEEE ICC best paper award, ICIP best paper finalist, and the IEEE Globecom best paper award all in 2020.

YIN Changchuan received his PhD degree in telecommunication engineering from Beijing University of Posts and Telecommunications, China in 1998. In 2004, he was a visiting scholar in the Faculty of Science, the University of Sydney, Australia. From 2007 to 2008, he held a visiting position with the Department of Electrical and Computer Engineering, Texas A&M University, USA. He is currently a professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include wireless networks and statistical signal processing. He was the co-recipient of the IEEE International Conference on Wireless Communications and Signal Processing Best Paper Award in 2009 and the IEEE Communications Society Young Author Best Paper Award in 2021. He has served as the symposium co-chair or TPC member for several IEEE flagship conferences (e.g., ICC, Globecom, ISIT, etc).

ZHU Guangxu received his BS and MS degrees from Zhejiang University, China and PhD degree from The University of Hong Kong, China, all in electronic and electrical engineering. He is now a research scientist with the Shenzhen Research Institute of Big Data. His research interests include edge intelligence, federated learning, and 5G/6G technologies. He is a recipient of the 2022 "AI 2000 Most Influential Scholar Award Honorable Mention", Hong Kong Postgraduate Fellowship (HKPF), Outstanding PhD Thesis Award from HKU, and the Best Paper Award from WCSP 2013. He served as a track/symposium/workshop co-chair of several IEEE conferences including IEEE PIMRC 2021, WCSP 2023, and IEEE Globecom 2023.

Adaptive Retransmission Design for Wireless Federated Edge Learning

XU Xinyi, LIU Shengli, YU Guanding
(Zhejiang University, Hangzhou 310027, China)

DOI: 10.12142/ZTECOM.202301002
https://kns.cnki.net/kcms/detail/34.1294.TN.20230220.1702.004.html, published online February 21, 2023
Manuscript received: 2022-10-27

Abstract: As a popular distributed machine learning framework, wireless federated edge learning (FEEL) can keep original data local, while uploading model training updates to protect privacy and prevent data silos. However, since wireless channels are usually unreliable, there is no guarantee that the model updates uploaded by local devices are correct, thus greatly degrading the performance of the wireless FEEL. Conventional retransmission schemes designed for wireless systems generally aim to maximize the system throughput or minimize the packet error rate, which is not suitable for the FEEL system. A novel retransmission scheme is proposed for the FEEL system to make a tradeoff between model training accuracy and retransmission latency. In the proposed scheme, a retransmission device selection criterion is first designed based on the channel condition, the number of local data, and the importance of model updates. In addition, we design the air interface signaling under this retransmission scheme to facilitate the implementation of the proposed scheme in practical scenarios. Finally, the effectiveness of the proposed retransmission scheme is validated through simulation experiments.

Keywords: federated edge learning; retransmission; unreliable communication; convergence rate; retransmission latency

Citation (IEEE Format): X. Y. Xu, S. L. Liu, and G. D. Yu, "Adaptive retransmission design for wireless federated edge learning," *ZTE Communications*, vol. 21, no. 1, pp. 3 – 14, Mar. 2023. doi: 10.12142/ZTECOM.202301002.

1 Introduction

With the construction of smart cities, a large number of Internet of Things devices, smartphones and other mobile devices have emerged from all aspects of our lives. The current society has entered the era of big data, and hundreds of millions of data are generated on mobile terminals every day[1 – 3], which poses novel challenges to both traditional centralized machine learning approaches and wireless communication techniques[4 – 5]. On the one hand, due to a large number of data, uploading all data to the cloud would result in a huge communication burden[6], and on the other hand, since the data contain user privacy, such as medical health and personal preferences, uploading raw data to the cloud would bring about the problem of privacy leakage[7 – 8].

To overcome the abovementioned challenges, a distributed machine learning framework named federated edge learning (FEEL) has been proposed recently[9 – 11]. Under FEEL, multiple distributed mobile devices use their locally dispersed data to jointly train a common machine learning model, rather than transferring raw data to a central node. The original data containing user privacy are stored on mobile devices, and only the intermediate data, such as gradients and parameters, are transmitted so that user privacy can be protected. In addition, FEEL shifts the model training process from the center to the local devices, thus making full use of distributed computing resources. Due to the advantages brought by the special architecture of FEEL, it has been intensively used in the fields of healthcare, computer vision, finance, etc.[12 – 15]

Recently, most research on FEEL assumes that communication links are reliable. For example, Ref. [16] considers the method of minimizing the transmitted energy under the delay constraint to improve the performance of FFEL. However, in practice, especially in wireless FEEL, channel transmission is generally unreliable due to random channel fading, shadowing, and noise. The accuracy of the intermediate data transmission during training cannot be guaranteed[17]. Retransmission is an important means to improve the accuracy of transmission in wireless communication systems, but with the cost of increasing the communication delay[18]. However, with the application of FEEL in medical and autonomous driving, it is more sensitive to the accuracy and delay of transmission[19]. This motivates us to investigate novel retransmission schemes for FEEL in this paper.

1.1 Related Work

There have been several studies considering the channel unreliability of wireless communications in distributed learning systems. In Ref. [20], the wireless channel in the FEEL system is modeled as an erasure channel and a scheme for this situation is proposed, which inherits the previous round of gradient when the packet is lost. Based on this, the authors further analyze the influence of coding rate on wireless

FEEL in Ref. [21]. In Ref. [22], a decentralized stochastic gradient descent method under the user datagram protocol (UDP) is proposed to reduce the impact of unreliable channels on decentralized federated learning. Moreover, an asynchronous decentralized stochastic gradient descent algorithm is proposed in Ref. [23] to reduce the impact of unreliable channels by performing asynchronous learning and reusing outdated gradients in device-to-device (D2D) networks. The authors in Ref. [24] have proposed an unbiased statistical reweighted aggregation scheme from the perspective of gradient aggregation, which comprehensively considers node fairness, unreliable parameter transmission, and resource constraints. In Ref. [25], a sparse federated learning framework is proposed, which compensates for the bias caused by unreliable communication through the similarity between local models, and adds local sparseness to reduce communication cost, which further improves performance. In Ref. [26], a federated learning framework is proposed, where the central server aggregates the global model according to the received parameters and the transmission correct probability, thereby reducing the impact of unreliable transmission. The authors in Ref. [27] further propose a decentralized D2D framework under unreliable channels, which reduces the impact of unreliable channels by jointly optimizing the transmission rate and bandwidth distribution.

From the perspective of wireless communication, retransmission has been applied to many current communication standards, including 5G and WiFi. So far, only a few works have studied the retransmission issue in distributed learning. Retransmission can improve the reliability of data packets, but it also reduces the timeliness of data. In some scenarios, it may even be considered to improve the timeliness of data at the cost of reduced reliability[28]. In Ref. [29], a Hybrid Automatic Repeat reQuest (HARQ) protocol suitable for multilayer cellular networks has been proposed, which can enhance error detection and correction in D2D communications. In Ref. [30], a retransmission scheme based on data importance is proposed for the edge learning system. The specific approach of this scheme is to make a tradeoff between the signal-to-noise ratio (SNR) and the uncertainty of the data, and correspondingly establish a threshold for retransmission.

1.2 Motivations and Contributions

As aforementioned, in wireless FEEL, devices upload gradients to the edge server through wireless channels, which is unreliable. This will affect the performance of model training. The goal of conventional retransmission schemes is to maximize the throughput of correctly transmitted data. However, the performance of FEEL with unreliable channels is limited by traditional retransmission since FEEL has different goals of learning accuracy and learning latency. In particular, the importance of data from different devices is different and generally contributes differently to the model training process. In addition, the communication cost introduced by retransmission of each device is also different due to various channel fading environments. The above factors need to be considered when developing a retransmission scheme for the edge learning system. The main contributions of this paper can be summarized as follows.

• We first propose a FEEL framework with unreliable channels, in which the gradients uploaded by the local devices are split into multiple packets, and the wireless channel exists the packet error rate (PER). Unreliable transmission leads to bias between the actual global gradient and the theoretical one, which is detrimental to model training.

• We mathematically analyze the effect of PER on the convergence rate and communication cost. To mitigate the impact of unreliable communications on learning performance, the retransmission device selection is optimized by making a tradeoff between convergence rate and communication cost.

• We derive the optimal solution to device retransmission selection, which greatly improves the model training performance. We also analyze the performance of the proposed retransmission selection scheme and develop a signaling protocol for retransmission.

• We employ a convolutional neural network (CNN) model of the CIFAR-10 and MNIST datasets to test the learning performance of our proposed retransmission selection scheme. Test results show that our proposed scheme outperforms several existing retransmission schemes.

The rest of the paper is organized as follows. In Section 2, we introduce the system model. In Section 3, the principle of retransmission design is introduced, and the corresponding protocol is proposed. In Section 4, we analyze the retransmission gain and cost and formulate the retransmission selection optimization problem. The retransmission selection is derived in Section 5. Finally, we draw the conclusions in Section 6.

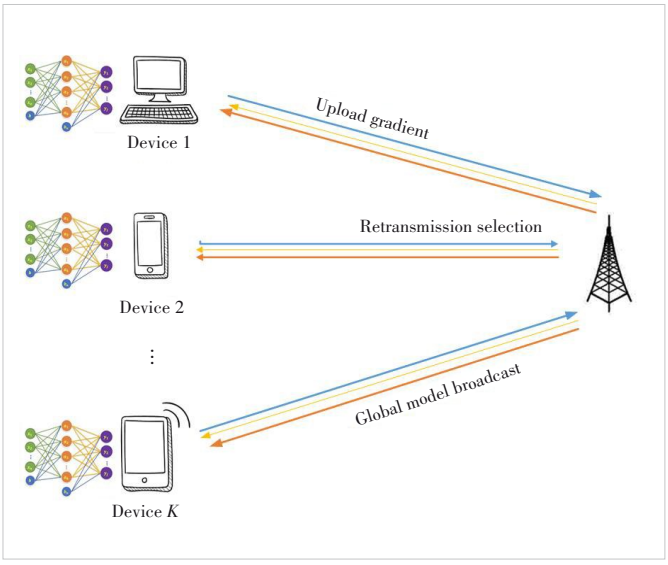
2 System Model

2.1 Machine Learning Model

As depicted in Fig. 1, we consider a FEEL system consisting of one edge server and K devices. Device k has n_k locally labeled data, and the total number of data in the entire system can be represented as $n = \sum_{k=1}^{K} n_k$. All devices only use their own data to jointly train a machine learning model $\boldsymbol{w}$ with the edge server, and the specific method is stochastic gradient descent (SGD). Considering the imbalance of data distribution, the global loss function can be written as:

$$L(\boldsymbol{w}) = \frac{1}{n}\sum_{k=1}^{K} n_k L_k(\boldsymbol{w}), \tag{1}$$

where $L_k(\boldsymbol{w})$ is the loss function of device k, and we have



▲Figure 1. Federated edge learning system

$$L_k(\boldsymbol{w}) = \frac{1}{n_k}\sum_{i=1}^{n_k} f(\boldsymbol{w}, x_{i,k}, y_{i,k}), \tag{2}$$

where $x_{i,k}$ represents the i-th training data of device k, $y_{i,k}$ represents the corresponding label, and $f(\cdot)$ represents the loss function of the training model. Some popular machine learning loss functions are summarized in Table 1.

The purpose of federated training is to find the optimal $\boldsymbol{w}^*$ that minimizes $L(\boldsymbol{w})$. FEEL is different from the traditional centralized machine learning framework. In the FEEL framework, all the original data are kept on local devices, and the training results are uploaded to the edge server. In the t-th round of training, the selected devices use the local data and the global model $\boldsymbol{w}^t$ received from the edge server to obtain the loss function $L_k(\boldsymbol{w}^t)$, and upload the gradient of $L_k(\boldsymbol{w}^t)$ to the edge server, which can be written as:

$$g_k^t = \nabla L_k(\boldsymbol{w}^t). \tag{3}$$

After receiving the uploaded gradients of all selected devices, the edge server decodes the data packets and aggregates the global gradient g^t as:

$$g^t = \frac{1}{n}\sum_{k=1}^{K} n_k g_k^t. \tag{4}$$

▼Table 1. Loss function for popular machine learning models

Learning Model	Loss Function $f(\boldsymbol{w}, x, y)$
Linear regression	$\frac{1}{2}\left\| y - \boldsymbol{w}^T x \right\|^2$
Least-squared support vector machine	$\frac{1}{2}\max\{0, 1 - y\boldsymbol{w}^T x\}^2$
Neural network	$\frac{1}{2}\left\| y - \phi(\boldsymbol{w}, x) \right\|$, where $\phi(\boldsymbol{w}, x)$ is the learning output

Then the edge server uses the global gradient g^t obtained by the aggregation to update the model, that is, $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta g^t$, where η is the learning ratio. After completing the update of the global model, the edge server broadcasts it to each device in the system. In this way, one round of iterative training of FEEL is completed.

2.2 Wireless Communication Model

In this paper, we utilize time division multiple access (TDMA) as the multiple access method. In a TDMA scenario, all devices use the same frequency band in different time slots and upload gradients to the edge server in turn. During one training iteration, it is assumed that the expected channel state information can be obtained by the channel estimation algorithms. Among the training iterations, the channel of the iteration differs from one another. The expected channel state information in each iteration is separately adopted for the performance analysis. Therefore, when a device uploads the gradients, it will occupy the full bandwidth, denoted by B. For ease of analysis, it is assumed that the wireless channel is static at each training gradient upload and changes in different rounds of training iterations. It is further assumed that the distances of all local devices to the edge server are known, and the small-scale fading is modeled as Rayleigh fading. Then, we can express the uploaded data rate of the device k as:

$$R_k = B\log_2\left(1 + \frac{P_k^U |h_k^U|^2}{N_0}\right), \tag{5}$$

where P_k^U is the transmit power of device k, h_k^U is the channel power gain between the device and the edge server, and N_0 is the noise power over the whole bandwidth B. We assume that each device is uploading and retransmitting data at the maximum available power. Note that this assumption fits many scenarios, such as LTE[31].

Since wireless channels are generally unreliable, channel errors need to be considered. It is assumed that the uploaded gradients of each device are divided into several packets, and each packet has redundant encoding for error detection. In this paper, the cyclic redundancy check (CRC) code is used to check for errors. Then the PER of device k can be expressed as:

$$p_k = 1 - \exp\left(-\frac{mBN_0}{P_k^U h_k^U}\right), \tag{6}$$

where m is the PER decision threshold[32].

Since the global model sent by the edge server to all devices is the same, the downlink channel can be modeled as a broadcast channel and a more robust encoding method can be used. In this paper, we consider that the channel error occurs only in the uplink channel, and assume that there is no channel error in the downlink channel. Let the channel bandwidth of the downlink channel be B_D, and denote γ as the smallest

SNR among all devices, and then the achievable downlink data rate is expressed as:

$$R_D = B_D \log_2(1 + \gamma). \tag{7}$$

3 Retransmission Protocol

In this section, we first introduce the principle of retransmission design in FEEL. Then, we propose a novel retransmission protocol and develop the corresponding processing modules for both devices and the edge server.

3.1 Principle of Retransmission Design

In FEEL, the edge server performs global model updates by periodically aggregating local gradients uploaded by devices. Therefore, the performance of the trained model depends on the quality of the gradients received by the edge server. However, unreliable gradient transmission may occur due to wireless channel impairments including interference, noise and shadowing. Therefore, it is predicted that the performance of model training is largely affected by channel impairments.

A common solution to unreliable transmission is retransmission. Conventionally, the purpose of retransmission is to ensure the reliability of data and at the same time maximize the system throughput. However, the main goal of FEEL is to maximize the training accuracy for a given training time. Therefore, a novel retransmission protocol is required for the FEEL system.

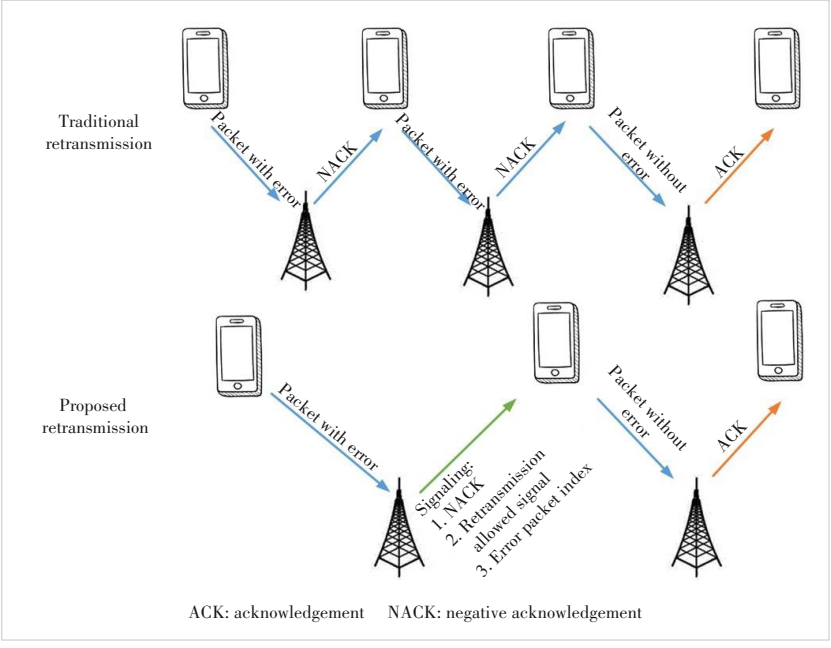
When designing the retransmission protocol for a FEEL system, one should consider both the training accuracy and the additional communication cost brought by retransmission. Retransmission can reduce erroneous packets so that the gradient updates received by the edge server deviate less from the ground-truth gradient, which can improve the convergence speed and the accuracy of model training. However, retransmission also increases the communication latency, resulting in an increase in training time. Therefore, we need to properly select the devices that need to be retransmitted and design appropriate signaling to make a fair tradeoff between learning accuracy and learning latency.

3.2 Retransmission Protocol and Processing Module

In our proposed retransmission protocol, not all devices participate in retransmission, that is, retransmission selection is required. Considering the characteristics of FEEL, the device selection depends on not only the channel conditions but also the local data volume and the importance of the upload gradient. Gradient updates that have a more significant impact on global model training will be retransmitted with a larger probability. Moreover, the latency caused by retransmission should also be accounted for. In our proposed protocol, a device with a higher data rate is also more likely to be retransmitted because it brings less additional communication cost. In addition, the PER between the device and the edge server shall also be taken into account. Due to the robustness of model training, devices with a small PER would bring little performance gain when retransmitting. Also, for a device with a large PER, the reduction of the PER after retransmission is very limited, but it will cause a relatively large communication cost. Therefore, when the PER is too large or too small, the probability of the device being selected for retransmission is both small.

We also consider a new design of retransmission signaling, as shown in Fig. 2. Under the traditional retransmission scheme, after receiving an erroneous packet, the edge server only sends a negative acknowledgement (NACK) signal to the device, requiring the device to retransmit. Until the edge server successfully decodes the data packet, it sends an acknowledgement (ACK) signal to the device, and the device starts to transmit the next data packet. In our protocol, when an edge server receives a packet and detects an error using CRC codes, it sends a signal to the corresponding device that includes the information shown in Fig. 2.

In Fig. 2, NACK indicates that the packet is transmitted with an error, but unlike that in the traditional retransmission schemes, it does not indicate that the device needs to retransmit the packet. Whether to retransmit needs to be judged according to the retransmission selection algorithm. Retransmission allowed signal ν_k indicates whether the device is selected for retransmission, which is related to the channel conditions, the number of local data, and the importance of the gradient.



▲Figure 2. Retransmission signaling

Specifically, $\nu_k = 1$ indicates that the device k is selected for retransmission; otherwise $\nu_k = 0$ indicates no retransmission. When $\nu_k = 1$, it is equivalent to traditional NACK. Error packet index represents the gradient position contained in the transmission data packet. If it is selected for retransmission, the device can retransmit the gradient of the corresponding position.

According to the received signal, the device will determine whether the uploaded packet is transmitted correctly and whether it is allowed to retransmit. After that, it retransmits the particular data corresponding to the erroneous packet, as indicated by the edge server.

4 Retransmission Design

In this section, we first analyze the one-round convergence rate with unreliable channels. Then, we propose a new criterion to evaluate the gain of retransmission on learning performance. The retransmission cost is analyzed as well. Based on this, we formulate a mathematical optimization problem to make a tradeoff between retransmission gain and retransmission cost.

4.1 One-Round Convergence

Due to the PER, during one round of training, the global gradient obtained by the edge server using the received gradient is not equal to the theoretical gradient g^t in Eq. (4). Therefore, we define the actual global gradient obtained by the aggregation under the unreliable channel as $\hat{g}^t$, and we have:

$$\hat{g}^t = \frac{\sum_{k=1}^{K} n_k \hat{g}_k^t}{n}, \tag{8}$$

where $\hat{g}_k^t$ is the actual local gradient of device k received by the edge server. Therefore, when there exists PER, the model update is:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \hat{g}^t = \boldsymbol{w}^t - \eta(g^t - o^t), \tag{9}$$

where o^t is the deviation of the global gradient introduced by unreliable transmission, and we have:

$$o^t = g^t - \frac{\sum_{k=1}^{K} n_k \hat{g}_k^t}{n}. \tag{10}$$

To facilitate mathematical analysis, we make the following assumption.

Assumption 1: (ℓ-smooth loss function) The global loss function is Lipschitz continuous with positive parameter ℓ, shown as:

$$\left\| g^{t+1} - g^t \right\| \leqslant \ell \left\| \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \right\|. \tag{11}$$

Based on the above assumption, we can obtain the convergence rate of one round under an unreliable channel.

Theorem 1: When the learning rate $\eta = \frac{1}{\ell}$, the training loss function in one round can be written as:

$$\mathbb{E}\left\{L(\boldsymbol{w}^{t+1})\right\} \leqslant \mathbb{E}\left\{L(\boldsymbol{w}^t)\right\} - \frac{1}{2\ell}\mathbb{E}\left\{\left\| g^t \right\|^2\right\} + \frac{1}{2\ell}\mathbb{E}\left\{\left\| o^t \right\|^2\right\}. \tag{12}$$

See Appendix A for details.

From Eq. (12), it can be seen that the loss function is constrained by three terms. The first term $\mathbb{E}\left\{L(\boldsymbol{w}^t)\right\}$ represents the loss function of the previous training round, which is independent of unreliable transmissions. The second item $\frac{1}{2\ell}\mathbb{E}\left\{\left\| g^t \right\|^2\right\}$ is related to the theoretical gradient value of this round, which depends on the data in local devices, but is independent of PER and the retransmission scheme. The third term $\frac{1}{2\ell}\mathbb{E}\left\{\left\| o^t \right\|^2\right\}$ is the bias term introduced by channel errors, which will reduce the loss function, thus affecting the convergence speed. In order to reduce the influence of unreliable channels and improve training performance, we need to reduce channel interference. Therefore, we next analyze the impact of PER (p_k^t) on the gradient bias $\mathbb{E}\left\{\left\| o^t \right\|^2\right\}$. Since we focus on the retransmission design of each round, for the convenience of presentation, we ignore the superscript t that represents the number of training rounds in the following.

We first assume that the machine learning model has a total of D layers of neural networks, and the device divides the corresponding gradients into D packets during the uploading process. The d-th packet contains gradient updates for the d-th layer of the neural network, which is denoted as $g_{k,d}$. Let indicator $\rho_{k,d}$ denote whether the transmission of the d-th packet of device k is correct. That is, $\rho_{k,d} = 1$ indicates that there is no error in the transmission, which means that the edge server can decode and obtain the correct gradient $g_{k,d}$, and there is a probability of $P(\rho_{k,d} = 1) = 1 - p_k$. Similarly, we let $\rho_{k,d} = 0$ denote the occurrence of a transmission error with probability of $P(\rho_{k,d} = 0) = p_k$. After the edge server receives the packets, if the error is detected and retransmission is not considered, the corresponding gradient is set to zero, which can be written as:

$$\hat{g}_{k,d} = \begin{cases} g_{k,d}, \rho_{k,d} = 1 \\ 0, \rho_{k,d} = 0 \end{cases}. \tag{13}$$

Lemma 1: The impact of error transmission on learning performance can be expressed as the bias of gradients caused by packet transmission errors, which can be written as:

$$\mathbb{E}\left\{\left\| o \right\|^2\right\} \leqslant \frac{K}{n^2}\sum_{k=1}^{K} n_k^2 p_k^2 \bar{g}_k^2, \tag{14}$$

where $\bar{g}_k = \sum_{d=1}^{D} g_{k,d}$ denotes the sum of the gradient of device k. See Appendix B for details.

First, the gradient bias term is affected by the PER p_k. The larger the PER of the device is, the larger the error term will be, and the smaller the loss function will decrease in one round. Second, the error term is affected by the number of local data on each device. The larger the number is, the more significant the impact of the device's PER on the entire model. Third, the error term is also affected by the gradient obtained from training. The larger the sum of uploaded gradients is, the larger the bias term would be introduced. Finally, since the global gradient is obtained by aggregating the uploaded gradients of selected devices, the bias term can be expressed as the sum of the bias introduced by each device due to unreliable transmission. Through the above analysis, we can obtain the convergence rate of one round in the presence of transmission errors as:

$$\mathbb{E}\left\{L(\boldsymbol{w}^{t+1})\right\} \leqslant \mathbb{E}\left\{L(\boldsymbol{w}^{t})\right\} - \frac{1}{2\ell}\mathbb{E}\left\{\left\| g^{t} \right\|^2\right\} + \frac{K}{2\ell n^2}\sum_{k=1}^{K} n_k^2 p_k^2 \bar{g}_k^2. \tag{15}$$

4.2 Gain of Retransmission

Next, we analyze the learning performance gain brought by retransmission. Define the PER of device k after the retransmission selection as q_k, which can be written as:

$$q_k = p_k\left(1 - \nu_k(1 - p_k)\right), \tag{16}$$

where p_k is the probability that an error occurs in one transmission, and $\nu_k(1 - p_k)$ represents the probability that device k is selected for retransmission and there is no error in the retransmission. Based on Eq. (14), considering the retransmission, the impact of PER on the convergence can be expressed as:

$$\mathbb{E}\left\{\left\| o_r \right\|^2\right\} \leqslant \frac{K}{n^2}\sum_{k=1}^{K} n_k^2 q_k^2 \bar{g}_k^2, \tag{17}$$

where o_r represents the bias between the theoretical gradients and the actual gradients after retransmission.

The PER of the device selected for retransmission will be reduced after retransmission, and its impact on learning performance will also be reduced. Therefore, we can present the following definition to analyze the gain which is achieved by retransmission.

Definition 1: We define the gain of retransmission as the difference between the bias of global gradients before and after retransmission on the learning performance, which can be written as

$$\Omega = \frac{K}{n^2}\sum_{k=1}^{K} n_k^2 p_k^2 \bar{g}_k^2 - \frac{K}{n^2}\sum_{k=1}^{K} n_k^2 q_k^2 \bar{g}_k^2 = \sum_{k=1}^{K} \Omega_k, \tag{18}$$

where Ω_k is the gain of retransmission of device k. Since the whole system can be regarded as a collection of all devices, we have:

$$\Omega_k = \frac{K}{n^2} n_k^2 \bar{g}_k^2\left(p_k^2 - q_k^2\right). \tag{19}$$

Eq. (19) reveals that the retransmission gain of the device is related to the number of local data, the value of the gradient update, and the reduction of the PER before and after retransmission. A larger data volume and gradient value of the device will bring a larger gain of retransmission to the learning performance. This solution can also be applied to dynamic wireless channels, just changing the retransmission PER to the actual PER.

4.3 Cost of Retransmission

Although device retransmission will bring gains to the learning performance, retransmission will also increase communication latency due to the additional resource required by retransmission. Therefore, we give the definition of the cost of retransmission as follows.

Definition 2: The cost of retransmission of device k is defined as the increase in latency introduced by retransmission, which can be expressed as

$$C_k = \frac{qNp_k}{R_k}\nu_k, \tag{20}$$

where q is the number of quantization bits and N is the total number of parameters.

4.4 Problem Formulation

Until now we have analyzed the gain and cost of retransmission. Retransmission will bring a gain in learning performance but increase additional communication costs. Therefore, we need to consider the tradeoff between cost and gain when developing a retransmission scheme. Our goal is to maximize retransmission gain while minimizing retransmission cost. We define $\beta \in [0,1]$ as a factor for the tradeoff between retransmission gain and retransmission cost, and the following retransmission gain-cost tradeoff problem can be established.

$$\text{P1: } \min_{\nu_k} \sum_{k=1}^{K}\left(-\beta\Omega_k + (1-\beta)C_k\right), \tag{21}$$

subject to

$$\nu_k \in \{0,1\}, \forall k. \tag{21a}$$

Eq. (21a) represents the retransmission indicator limitation. When β is close to 0, it means that the main goal is to reduce

the latency when retransmission is selected. When β is close to 1, it means that improving the convergence rate is the main goal.

5 Retransmission Optimization and Theoretical Analysis

In this section, we first give a retransmission selection strategy based on P1. Then, we analyze the effect of PER on retransmission selection.

5.1 Optimal Solution

By inserting Eqs. (16), (19), and (2) into Eq. (21), and relaxing the $\{0,1\}$ variable ν_k to $[0,1]$, P1 can be formulated as:

$$\text{P2: } \min_{\nu_k} \sum_{k=1}^{K} -\beta \frac{K}{n^2} n_k^2 \bar{g}_k^2 \left(p_k^2 - \left(p_k - \nu_k p_k (1-p_k) \right)^2 \right) + (1-\beta) \frac{qNp_k}{R_k} \nu_k, \tag{22}$$

subject to

$$\nu_k \in [0,1], \forall k. \tag{22a}$$

Eq. (22) consists of two parts: the first part is related to federated learning (FL) training loss, and the second part is related to FL one-round training latency. This is a classical convex optimization problem, and the optimal solution can be obtained through the Karush-Kuhn-Tucker (KKT) condition.

Theorem 2: The retransmission selection policy can be expressed as:

$$\nu_k^* = \left[\frac{1}{1-p_k} - \frac{(1-\beta)qNn^2}{2\beta K n_k^2 \bar{g}_k^2 p_k^2 (1-p_k)^2 R_k} \right]_0^1, \forall k, \tag{23}$$

where $[X]_0^1 = \min\{1, \max\{X, 0\}\}$. See Appendix C for further details.

Theorem 2 reveals that the retransmission indicator is a value bounded by 0 and 1, which is related to the local data volume, gradient value, data rate, and the PER of the device. Specifically, the probability of being selected for retransmission ν_k^* increases with the data number n_k and the gradient value $\bar{g}_k$ in the order of $-\frac{1}{2}$. This is because with a large number of device data and gradient values, the learning performance gain obtained by retransmission is also large. Also, ν_k^* increases with the data rate R_k in the order of -1. Since the data rate is large, the communication cost of retransmission will be small, and the probability of the device being selected for retransmission will increase. The impact of the device PER on the retransmission selection will be analyzed in the next section.

Since the obtained ν_k^* is the optimal solution after relaxation, we need to consider how to convert it into a $\{0,1\}$ variable for retransmission selection. We give two strategies. The first is to perform threshold processing on ν_k^*, with 0.5 as the limit. If $\nu_k^* \geqslant 0.5$, it means retransmission, and if $\nu_k^* < 0.5$, it will not be retransmitted. The second is to sort all devices from large to small according to the value of ν_k^*, and select the largest proportion $M\%$ of devices of ν_k^* for retransmission. The choice of M reflects the tradeoff between model accuracy and training latency.

5.2 Theoretical Analysis

In this section, we will analyze the impact of PER on the retransmission indicator. We first define:

$$m_k = \frac{(1-\beta)qNn^2}{2\beta K n_k^2 \bar{g}_k^2 R_k}. \tag{24}$$

From Eq. (24), m_k is related to the number of local data, gradient value and data rate, but is irrelevant to the PER. When the local data volume, the gradient value, and the uploaded data rate of device k are large, device k is more important in the retransmission design, and m_k is correspondingly small. Therefore, m_k reflects the contribution of the gradient of device k to the global model training, as well as the state of its channel. And m_k is always greater than 0. Moreover, the importance of device decreases as m_k increases. Then, in order to analyze the influence of p_k on the retransmission indicator ν_k^*, we define the following function:

$$f(p_k) = \frac{1}{1-p_k} - \frac{(1-\beta)qNn^2}{2\beta K n_k^2 \bar{g}_k^2 p_k^2 (1-p_k)^2 R_k} = \frac{1}{1-p_k} - \frac{m_k}{p_k^2 (1-p_k)^2}, \tag{25}$$

where $f(p_k)$ is a strictly unimodal function with $p_k \in [0,1]$. See Appendix D for details.

Theorem 2 reveals that the optimal retransmission indicator first increases and then decreases with p_k. Therefore, there exists an optimal p_k^* that maximizes $f(p_k)$. This result is rather intuitive, which shows that there is a tradeoff between retransmission gain and cost. For the device with a low PER, due to the robustness of neural networks, retransmission has little gain in learning performance, but will increase communication cost. Therefore, its probability of being selected for retransmission is relatively low. For the device with a relatively high PER, there will still be a high PER after retransmission. Thus, the gain in model training performance is not large. Also, the retransmission cost is large, and the probability of being selected is low. Note that devices with intermediate PER can improve the accuracy of gradient data after retransmission, and will not bring reused data or additional deviation.

6 Numerical Result

In this section, we conduct extensive experiments to verify the effectiveness of the proposed retransmission scheme.

6.1 Simulation Settings

Assume that the coverage area of the edge server is 1.5 km, and there are K (K=10) mobile devices that are randomly distributed across the cellular network. The transmit power of each device is 28 dBm, and the transmit power of the edge server is 33 dBm. Then, the noise power spectral density is −174 dBm/Hz and the PER decision threshold m = 0.2 dB. Since in the TDMA scenario, all devices occupy one channel to upload gradients. The uplink channel takes into account large-scale fading, given by $128.1 + 37.6\log(d)$, where d represents the distance between the device and the edge server in kilometers. We also consider small-scale fading of the channel, specifically represented by Rayleigh fading. All devices and the edge server jointly train a CNN model. We choose CIFAR-10 and MNIST as datasets. CIFAR-10 consists of 50 000 training images and 10 000 testing images. And MNIST consists of 55 000 training images and 5 000 testing images. The datasets are both non-identically and independently distributed (non-IID) and divided into 10 categories. Also, we choose the learning ratio η = 0.05. We quantize each element of the uploaded gradient with 16 bits. All elements of each layer are treated as one packet, and a 32-bit CRC code is added. Other major parameters are listed in Table 2.

6.2 Performance of Proposed Retransmission Scheme

Based on the previous theoretical analysis, the proposed algorithm can make a tradeoff between reducing the gradient aggregation bias caused by unreliable transmission and controlling the transmission delay, thereby accelerating the model convergence. We use the global training loss and global test accuracy to evaluate the learning performance of the whole learning system. In the simulation of this section, the discretization method for the retransmission factor ν_k^* is to take 0.5 as the threshold. That is, the selection indicator is set to 0 if ν_k^* is less than 0.5 and set to 1 if it is larger than 0.5.

The comparison algorithms in Fig. 3 are shown as follows.

• Without PER: The wireless channel is ideal and PER-free, meaning that all gradients can be transmitted correctly.

• Without retransmission: There is PER in the uplink channel, but retransmission is not considered. If the uploaded data packet is judged to be incorrect, it will be set to zero and the packet will be discarded.

• Existing retransmission schemes: Using the existing retransmission scheme based on the transmission result. The devices retransmit the erroneous data packets after receiving the NACK.

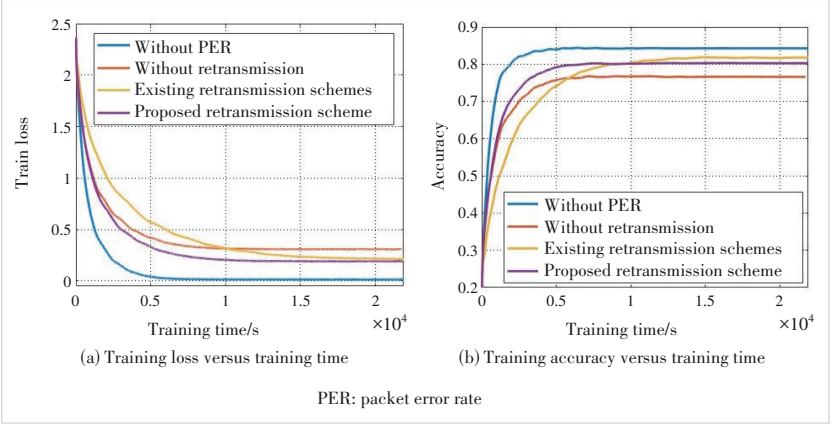
• The proposed retransmission scheme: Using the scheme proposed in this paper, we made the retransmission selection according to the device's local data, gradient data, and PER.

We first perform simulations under the CIFAR-10 dataset. The curves of training loss and test accuracy versus training time under different retransmission schemes are shown in Fig. 3. As can be seen from the figure, when transmitting on a reliable channel, no retransmission is required. At this time, the model training can reach convergence in a very short time with a high model accuracy. When the channels are unreliable and retransmission is not performed, the performance of model training will be greatly degraded. When retransmission is not performed, model training can reach convergence very fast, but the accuracy of the final model is pretty low. As a result, when there is no retransmission, the communication cost is relatively small. Although multiple rounds of training are required, one round of training latency is short, so the overall latency is short. However, due to the large bias between the received gradient and the local gradient, the performance of the final trained model is not satisfactory, which also confirms the necessity of retransmission. It can also be seen that, in the existing retransmission scheme, although the accuracy of the final model is high, it takes much longer time to converge. This is because the existing retransmission scheme aims to maximize the throughput, without considering selecting retransmission devices, or the importance of uploading gradients for model training. Due to a large number of transmitted gradient data and participating training de-

▼**Table 2. Simulation parameters**

Parameters	Values
Path loss model	$128.1 + 37.6\log(d)$
Transmission power of the edge server	33 dBm
Transmission power of device	28 dBm
Additive white Gaussian noise power	−174 dBm/Hz
Bandwidth of downlink	10 MHz
Quantization bit of each element	16
Number of devices	10
Bandwidth of uplink	10 MHz
CRC code	32

CRC: cyclic redundancy check



(a) Training loss versus training time (b) Training accuracy versus training time

PER: packet error rate

▲**Figure 3. Performance comparison between transmission schemes under CIFAR-10**

vices, the wireless FEEL system needs to spend a lot of time to achieve model convergence without retransmission selection. Therefore, the existing retransmission schemes cannot exhibit good performance under the FEEL system. As shown in Fig. 3, the retransmission scheme proposed in this paper can make the model training converge in a short time, and achieve high accuracy at the same time. The reason is that the influence of different gradients has been considered in the retransmission. This scheme can maximize the retransmission gain, reduce the influence of channel errors, and improve the performance of model training by selecting proper retransmission devices. In order to further illustrate the effectiveness of our proposed scheme, we increase the number of devices to 20 for simulation, and the results are shown in Fig. 4.

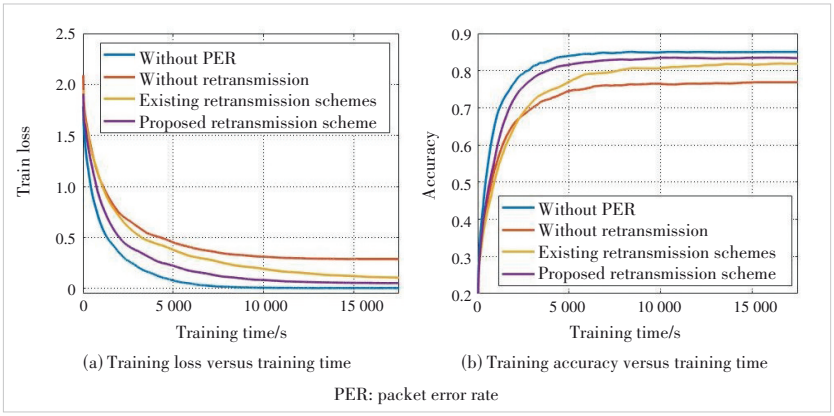
6.3 Performance with Difference Retransmission Ratios

When selecting $M\%$ of devices for retransmission in each round of transmission, the choice of parameter M may reflect the tradeoff between model accuracy and training latency in our proposed retransmission scheme.

From Fig. 4, when M is too small, e.g., 20% or 40%, both the convergence rate and final model accuracy become low. This is because the impact of channel error is strong when the number of selected retransmission devices is small. When M is too big, e.g., 80%, the convergence speed is low and the final accuracy has no significant advantage. This is because retransmission will increase the latency, and some devices are not of high importance, resulting in limited retransmission gain.

6.4 Performance Comparison Under Other Datasets

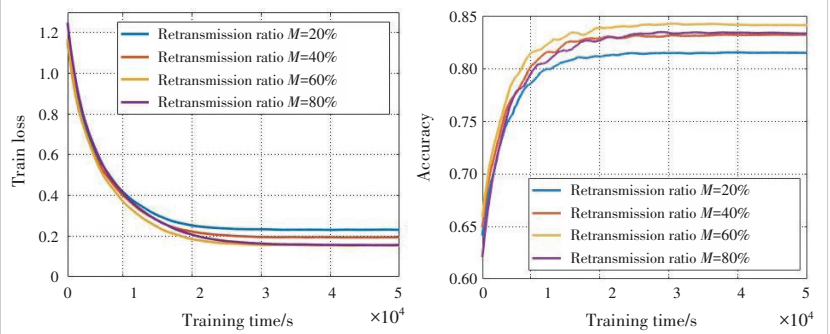
To verify the broad effectiveness of our proposed scheme, we change the training dataset to MNIST for further simulations. MNIST consists of 0 – 9 numbers handwritten by different people. The curves of training loss and test accuracy are shown in Fig. 6. After the dataset is changed, the effect of channel unreliability on model training and the performance improvement of our proposed scheme can still be seen. From Fig. 7, the proportion M of retransmission devices still affects performance, which further proves the necessity of retransmission device selection.



(a) Training loss versus training time (b) Training accuracy versus training time

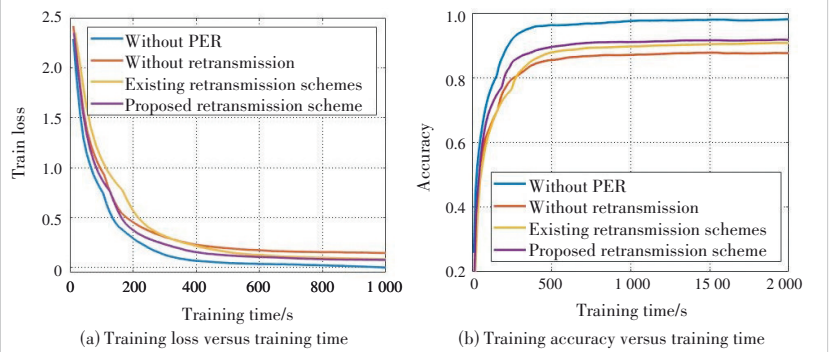
PER: packet error rate

▲ **Figure 4. Performance comparison between different retransmission schemes under CIFAR-10 with device number K=20**



(a) Training loss versus training time (b) Training accuracy versus training time

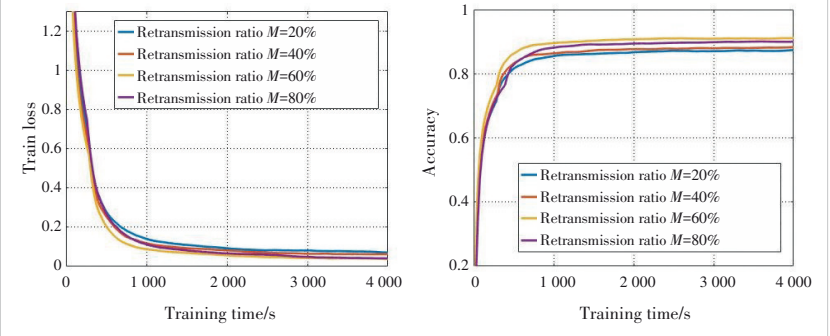
▲ **Figure 5. Performance comparison between different M under CIFAR-10**



(a) Training loss versus training time (b) Training accuracy versus training time

PER: packet error rate

▲ **Figure 6. Performance comparison between transmission schemes under MNIST**



(a) Training loss versus training time (b) Training accuracy versus training time

▲ **Figure 7. Performance comparison between different M under MNIST**

7 Conclusions

In this paper, we mainly study the retransmission design for FEEL under unreliable channels. We first analyze the impact of unreliable transmission on the training performance of the FEEL model, and derive the relation between the loss function and the channel PER in one round. Based on this, we analyze the gain to the convergence rate brought by device retransmission, as well as the communication cost introduced. Then, we propose a retransmission selection scheme for FEEL with unreliable channels, which can make a tradeoff between the training accuracy and the transmission latency. It comprehensively considers the channel conditions, the number of local data, and the importance of updates. We also present the air interface signaling and retransmission protocol design under the proposed retransmission selection scheme. Finally, the effectiveness of the proposed retransmission scheme is verified by extensive simulation experiments. The results show that our proposal can effectively reduce the impact of unreliable wireless channels on the training of the FEEL model, and is superior to the existing retransmission schemes.

Appendix A

Proof of Theorem 1

We first use the second-order Taylor expansion of $L\left(\boldsymbol{w}^{t+1}\right)$ to get

$$L\left(\boldsymbol{w}^{t+1}\right)=L\left(\boldsymbol{w}^{t}\right)+\left(\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\right)\nabla L\left(\boldsymbol{w}^{t}\right)+\frac{1}{2}\left(\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\right)^{T}\nabla^{2}L\left(\boldsymbol{w}^{t}\right)\left(\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\right). \tag{26}$$

Based on Eq. (11) in Assumption 1, we can get

$$L\left(\mathbf{w}^{t+1}\right)\leqslant L\left(\mathbf{w}^{t}\right)+\left(\mathbf{w}^{t+1}-\mathbf{w}^{t}\right)g^{t}+\frac{1}{2}\beta\left\|\mathbf{w}^{t+1}-\mathbf{w}^{t}\right\|^{2}. \tag{27}$$

By taking expectation over both sides, it follows

$$\mathbb{E}\left\{L\left(\mathbf{w}^{t+1}\right)\right\}\leqslant\mathbb{E}\left\{L\left(\mathbf{w}^{t}\right)\right\}+\mathbb{E}\left\{-\eta\left(g^{t}-o^{t}\right)^{T}g^{t}\right\}+\frac{1}{2}\beta\eta^{2}\mathbb{E}\left\{\left\|g^{t}-o^{t}\right\|^{2}\right\}. \tag{28}$$

To remove the cross-term, we fix $\eta=\frac{1}{\beta}$. Then it follows

$$\begin{aligned}&\mathbb{E}\left\{L\left(\mathbf{w}^{t+1}\right)\right\}\leqslant\mathbb{E}\left\{L\left(\mathbf{w}^{t}\right)\right\}-\frac{1}{\beta}\mathbb{E}\left\{\left(g^{t}-o^{t}\right)^{T}g^{t}\right\}+\frac{1}{2\beta}\mathbb{E}\left\{\left\|g^{t}-o^{t}\right\|^{2}\right\}=\\&\mathbb{E}\left\{L\left(\mathbf{w}^{t}\right)\right\}-\frac{1}{2\beta}\mathbb{E}\left\{\left(g^{t}-o^{t}\right)^{T}g^{t}\right\}+\frac{1}{2\beta}\mathbb{E}\left\{\left(g^{t}-o^{t}\right)^{T}o^{t}\right\}=\\&\mathbb{E}\left\{L\left(\mathbf{w}^{t}\right)\right\}-\frac{1}{2\beta}\mathbb{E}\left\{\left(g^{t}-o^{t}\right)^{T}\left(g^{t}+o^{t}\right)\right\}=\mathbb{E}\left\{L\left(\mathbf{w}^{t}\right)\right\}-\frac{1}{2\beta}\mathbb{E}\left\{\left\|g^{t}\right\|^{2}\right\}+\frac{1}{2\beta}\mathbb{E}\left\{\left\|o^{t}\right\|^{2}\right\}.\end{aligned} \tag{29}$$

Thus, we have completed the proof of Theorem 1.

Appendix B

Proof of Lemma 1

First, the bias term can be expressed as the difference between the ground-truth gradient and the aggregated gradient, which can be expressed as

$$\begin{aligned}&\mathbb{E}\left\{\left\|o\right\|^{2}\right\}=\mathbb{E}\left\{\left\|g^{t}-\hat{g}^{t}\right\|^{2}\right\}=\\&\mathbb{E}\left\{\left\|\frac{\sum_{k=1}^{K}n_{k}\sum_{d=1}^{D}g_{k,i}}{n}-\frac{\sum_{k=1}^{K}n_{k}\sum_{d=1}^{D}\hat{g}_{k,i}}{n}\right\|^{2}\right\}=\\&\mathbb{E}\left\{\left\|\frac{\sum_{k=1}^{K}n_{k}\sum_{d=1}^{D}\left(1-\rho_{k,d}\right)g_{k,d}}{n}\right\|^{2}\right\}.\end{aligned} \tag{30}$$

By opening it with the sum of squares formula and substituting the probability of the indicator $\rho_{k,d}$, $P(\rho_{k,d}=0)=p_k$ and $P(\rho_{k,d}=1)=1-p_k$, we can get

$$\begin{aligned}&\mathbb{E}\left\{\left\|o\right\|^{2}\right\}=\frac{1}{n^{2}}\mathbb{E}\left\{\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\sum_{d_1=1}^{D}\sum_{d_2=1}^{D}n_{k_1}\left(1-\rho_{k_1,d_1}\right)g_{k_1,d_1}n_{k_2}\left(1-\rho_{k_2,d_2}\right)g_{k_2,d_2}\right\}=\\&\frac{1}{n^{2}}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\sum_{d_1=1}^{D}\sum_{d_2=1}^{D}n_{k_1}p_{k_1}g_{k_1,d_1}n_{k_2}p_{k_2}g_{k_2,d_2}=\\&\frac{1}{n^{2}}\left(\sum_{k=1}^{K}n_{k}p_{k}\sum_{d=1}^{D}g_{k,d}\right)^{2}\leqslant\frac{K}{n^{2}}\sum_{k=1}^{K}n_{k}^{2}p_{k}^{2}\left(\sum_{d=1}^{D}g_{k,d}\right)^{2}.\end{aligned} \tag{31}$$

Denoting $\bar{g}_k=\sum_{d=1}^{D}g_{k,d}$, we can obtain the solution in Lemma 1.

Appendix C

Proof of Theorem 2

First, we take the first-order and second-order differentials of the objective function, and get

$$\begin{aligned}&\frac{\partial\sum_{k=1}^{K}\left(-\beta\Omega_{k}+\left(1-\beta\right)C_{k}\right)}{\partial\nu_{k}}=\\&-\frac{2\beta K}{n^{2}}n_{k}^{2}\bar{g}_{k}^{2}p_{k}^{2}\left(1-p_{k}-\nu_{k}\left(1-p_{k}\right)^{2}\right)+\left(1-\beta\right)\frac{qNp_{k}}{R_{k}},\end{aligned} \tag{32}$$

$$\frac{\partial^{2}\sum_{k=1}^{K}\left(-\beta\Omega_{k}+\left(1-\beta\right)C_{k}\right)}{\partial\nu_{k}^{2}}=\frac{2\beta K}{n^{2}}n_{k}^{2}\bar{g}_{k}^{2}p_{k}^{2}\left(1-p_{k}\right)^{2}\geqslant 0. \tag{33}$$

So the objective function of P2 is convex. In addition, Eq. (22a) is a linear constraint. Therefore, we can conclude that P2 is convex and we can use the KKT condition to find the optimal solution. We define the Lagrangian function $\mathcal{L}$ under the inequality constraints, as

$$\mathcal{L} = \sum_{k=1}^{K} \beta \frac{K}{n^2} n_k^2 \bar{g}_k^2 \left(p_k^2 - \left(p_k - \nu_k p_k (1 - p_k) \right)^2 \right) + (1-\beta) \frac{qNp_k}{R_k} \nu_k + \sum_{k=1}^{K} \mu_k (-\nu_k) + \sum_{k=1}^{K} \lambda_k (\nu_k - 1), \tag{34}$$

where $\mu_k \geqslant 0$ and $\lambda_k \geqslant 0$, which are both constraint coefficients of Eq. (22a). Let ν_k^* represent the optimal solution of P2. Then using the KKT condition, we can get

$$\frac{\partial \mathcal{L}}{\partial \nu_k^*} = -\frac{2\beta K}{n^2} n_k^2 \bar{g}_k^2 p_k^2 \left(1 - p_k - \nu_k^* (1 - p_k)^2 \right) + (1-\beta) \frac{qNp_k}{R_k} - \mu_k + \lambda_k, \forall k, \tag{35}$$

$$\mu_k (-\nu_k^*) = 0, \forall k, \tag{36}$$

$$\lambda_k (\nu_k^* - 1) = 0, \forall k. \tag{37}$$

By solving the above equations, we can get the optimal solution, as shown in Theorem 2.

Appendix D

Proof of Theorem 3

Taking the partial derivative of $f(p_k)$ over p_k, it follows

$$\frac{\partial f(p_k)}{\partial p_k} = \frac{p_k^3 (1 - p_k) + 2m_k (1 - 2p_k)}{p_k^3 (1 - p_k)}. \tag{38}$$

Then we define $h(p_k) = p_k^3 (1 - p_k) + 2m_k (1 - 2p_k)$. Taking the first-order and second-order differentials of $h(p_k)$, we have:

$$\frac{\partial h(p_k)}{\partial p_k} = 3p_k^2 - 4p_k^3 - 4m_k, \frac{\partial^2 h(p_k)}{\partial p_k^2} = 6p_k (1 - 2p_k). \tag{39}$$

Let $\frac{\partial^2 h(p_k)}{\partial p_k^2} = 0$, we have $\frac{\partial h(p_k)}{\partial p_k}$ that increases on (0, 0. 5) and decreases on (0. 5, 1). There is a unique p_k^* so that

$$h(p_k) \begin{cases} < 0, p_k \in (p_k^*, 0) \\ = 0, p_k = p_k^* \\ > 0, p_k \in (p_k^*, 1). \end{cases} \tag{40}$$

where p_k^* is related to m_k. And since $m_k > 0$, $p_k^* \in (0,1)$.

Therefore, we can prove that $f(p_k)$ increases on $(0, p_k^*)$ and decreases on $(p_k^*,1)$.

References

[1] ZHANG T, GAO L, HE C Y, et al. Federated learning for the Internet of Things: applications, challenges, and opportunities [J]. IEEE Internet of Things magazine, 2022, 5(1): 24 – 29. DOI: 10.1109/IOTM.004.2100182
[2] GUO F X, YU F R, ZHANG H L, et al. Enabling massive IoT toward 6G: a comprehensive survey [J]. IEEE Internet of Things journal, 2021, 8(15): 11891 – 11915. DOI: 10.1109/JIOT.2021.3063686
[3] MOHAMMADI F G, SHENAVARMASOULEH F, ARABNIA H R. Applications of machine learning in healthcare and Internet of Things (IOT): a comprehensive review [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2202.02868
[4] VERBRAEKEN J, WOLTING M, KATZY J, et al. A survey on distributed machine learning [J]. ACM computing surveys, 2021, 53(2): 1 – 33. DOI: 10.1145/3377454
[5] MAJEED I A, KAUSHIK S, BARDHAN A, et al. Comparative assessment of federated and centralized machine learning [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2202.01529
[6] GUPTA R, ALAM T. Survey on federated-learning approaches in distributed environment [J]. Wireless personal communications, 2022, 125(2): 1631 – 1652. DOI: 10.1007/s11277-022-09624-y
[7] JIANG Y L, ZHANG K, QIAN Y, et al. Anonymous and efficient authentication scheme for privacy-preserving distributed learning [J]. IEEE transactions on information forensics and security, 2022, 17: 2227 – 2240. DOI: 10.1109/TIFS.2022.3181848
[8] TRELEAVEN P, SMIETANKA M, PITHADIA H. Federated learning: the pioneering distributed machine learning and privacy-preserving data technology [J]. Computer, 2022, 55(4): 20 – 29. DOI: 10.1109/MC.2021.3052390
[9] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions [J]. IEEE signal processing magazine, 2020, 37 (3): 50 – 60. DOI: 10.1109/MSP.2020.2975749
[10] LIU J, HUANG J Z, ZHOU Y, et al. From distributed machine learning to federated learning: A survey [J]. Knowledge and information systems, 2022, 64(4): 885 – 917. DOI: 10.1007/s10115-022-01664-x
[11] ALEDHARI M, RAZZAK R, PARIZI R M, et al. Federated learning: a survey on enabling technologies, protocols, and applications [J]. IEEE access: practical innovations, open solutions, 2020, 8: 140699 – 140725. DOI: 10.1109/access.2020.3013541
[12] ABREHA H G, HAYAJNEH M, SERHANI M A. Federated learning in edge computing: a systematic survey [J]. Sensor, 2022, 22(2): 450. DOI: 10.3390/s22020450
[13] LIM W Y B, LUONG N C, HOANG D T, et al. Federated learning in mobile edge networks: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2020, 22(3): 2031 – 2063. DOI: 10.1109/COMST.2020.2986024
[14] NGUYEN D C, PHAM Q V, PATHIRANA P N, et al. Federated learning for smart healthcare: a survey [J]. ACM computing surveys, 2023, 55(3): 1 – 37. DOI: 10.1145/3501296
[15] ZHENG Z H, ZHOU Y Z, SUN Y L, et al. Applications of federated learning in smart cities: Recent advances, taxonomy, and open challenges [J]. Connection science, 2022, 34(1): 1 – 28. DOI: 10.1080/09540091.2021.1936455
[16] YANG Z H, CHEN M Z, SAAD W, et al. Energy efficient federated learning over wireless communication networks [J]. IEEE transactions on wireless communications, 2021, 20(3): 1935 – 1949. DOI: 10.1109/TWC.2020.3037554
[17] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks [J]. IEEE transactions on wireless communications, 2021, 20(1): 269 – 283. DOI: 10.1109/TWC.2020.3024629
[18] NADEEM F, LI Y H, VUCETIC B, et al. Analysis and optimization of HARQ

for URLLC [C]//IEEE Globecom Workshops. IEEE, 2022: 1 – 6. DOI: 10.1109/GCWkshps52748.2021.9682028
[19] JIANG P W, WEN C K, JIN S, et al. Deep source-channel coding for sentence semantic transmission with HARQ [J]. IEEE transactions on communications, 2022, 70(8): 5225 – 5240. DOI: 10.1109/TCOMM.2022.3180997
[20] SHIRVANIMOGHADDAM M, SALARI A, GAO Y F, et al. Federated learning with erroneous communication links [J]. IEEE communications letters, 2022, 26(6): 1293 – 1297. DOI: 10.1109/LCOMM.2022.3167094
[21] SALARI A, SHIRVANIMOGHADDAM M, VUCETIC B, et al. Rate-convergence tradeoff of federated learning over wireless channel [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2205.04672
[22] YE H, LIANG L, LI G Y. Decentralized federated learning with unreliable communications [J]. IEEE journal of selected topics in signal processing, 2022, 16(3): 487 – 500. DOI: 10.1109/JSTSP.2022.3152445
[23] JEONG E, ZECCHIN M, KOUNTOURIS M. Asynchronous decentralized learning over unreliable wireless networks [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2202.00955
[24] LI Z D, ZHOU Y J, WU D P, et al. Fairness-aware federated learning with unreliable links in resource-constrained Internet of Things [J]. IEEE Internet of Things journal, 2022, 9(18): 17359 – 17371. DOI: 10.1109/JIOT.2022.3156046
[25] MAO Y Z, ZHAO Z H, YANG M L, et al. SAFARI: sparsity enabled federated learning with limited and unreliable communications [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2204.02321
[26] SALEHI M, HOSSAIN E. Federated learning in unreliable and resource-constrained cellular wireless networks [J]. IEEE transactions on communications, 2021, 69(8): 5136 – 5151. DOI: 10.1109/TCOMM.2021.3081746
[27] JIANG Z H, YU G D, CAI Y L, et al. Decentralized edge learning via unreliable device-to-device communications [J]. IEEE transactions on wireless communications, 2022, 21(11): 9041 – 9055. DOI: 10.1109/TWC.2022.3172147
[28] NADEEM F, LI Y H, VUCETIC B, et al. HARQ optimization for real-time remote estimation in wireless networked control [EB/OL]. [2022-10-10]. https://arxiv.org/abs/2201.05838
[29] SHAH S W H, RAHMAN M M U, MIAN A N, et al. Effective capacity analysis of HARQ-enabled D2D communication in multi-tier cellular networks [J]. IEEE transactions on vehicular technology, 2021, 70(9): 9144 – 9159. DOI: 10.1109/TVT.2021.3100675
[30] LIU D Z, ZHU G X, ZENG Q S, et al. Wireless data acquisition for edge learning: data-importance aware retransmission [J]. IEEE transactions on wireless communications, 2021, 20(1): 406 – 420. DOI: 10.1109/TWC.2020.3024980
[31] SIMONSSON A, FURUSKAR A. Uplink power control in LTE - overview and performance, subtitle: principles and benefits of utilizing rather than compensating for SINR variations [C]//IEEE 68th Vehicular Technology Conference. IEEE, 2008: 1 – 5. DOI: 10.1109/VETECF.2008.317
[32] XI Y, BURR A, WEI J B, et al. A general upper bound to evaluate packet error rate over quasi-static fading channels [J]. IEEE transactions on wireless communications, 2011, 10(5): 1373 – 1377. DOI: 10.1109/TWC.2011.012411.100787

Biographies

XU Xinyi received her BE degree in communication engineering from Zhejiang University, China in 2021. Now she is working towards her MS degree with the College of Information Science and Electronic Engineering, Zhejiang University. Her research interest focuses on federated learning.

LIU Shengli received his BS degree in information engineering from Soochow University, China in 2017, and his PhD degree from the College of Information Science and Electronic Engineering, Zhejiang University, China in 2022. He currently holds a post-doctoral position at the College of Information Science and Electronic Engineering, Zhejiang University. In 2021, he was a Visiting Research Scholar with the Centre for Wireless Communication, University of Oulu, Finland and the VTT Technical Research Centre of Finland. His current research interests mainly include machine learning and federated learning.

YU Guanding (yuguanding@zju.edu.cn) received his BE and PhD degrees in communication engineering from Zhejiang University, China in 2001 and 2006, respectively. He joined Zhejiang University in 2006 and is now a professor with the College of Information and Electronic Engineering. From 2013 to 2015, he was also a visiting professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology, USA. His research interests include 5G communications and networks, mobile edge computing, and machine learning for wireless networks.

Reliable and Privacy-Preserving Federated Learning with Anomalous Users

ZHANG Weiting[1], LIANG Haotian[2], XU Yuhua[2], ZHANG Chuan[2]

(1. Beijing Jiaotong University, Beijing 100091, China;
2. Beijing Institute of Technology, Beijing 100081, China)

DOI: 10.12142/ZTECOM.202301003

https://kns.cnki.net/kcms/detail/34.1294.TN.20230210.1505.002.html, published online February 10, 2023

Manuscript received: 2022-11-01

Abstract: Recently, various privacy-preserving schemes have been proposed to resolve privacy issues in federated learning (FL). However, most of them ignore the fact that anomalous users holding low-quality data may reduce the accuracy of trained models. Although some existing works manage to solve this problem, they either lack privacy protection for users' sensitive information or introduce a two-cloud model that is difficult to find in reality. A reliable and privacy-preserving FL scheme named reliable and privacy-preserving federated learning (RPPFL) based on a single-cloud model is proposed. Specifically, inspired by the truth discovery technique, we design an approach to identify the user's reliability and thereby decrease the impact of anomalous users. In addition, an additively homomorphic cryptosystem is utilized to provide comprehensive privacy preservation (user's local gradient privacy and reliability privacy). We give rigorous theoretical analysis to show the security of RPPFL. Based on open datasets, we conduct extensive experiments to demonstrate that RPPEL compares favorably with existing works in terms of efficiency and accuracy.

Keywords: federated learning; anomalous user; privacy preservation; reliability; homomorphic cryptosystem

Citation (IEEE Format): W. T. Zhang, H. T. Liang, Y. H. Xu, et al., "Reliable and privacy-preserving federated learning with anomalous users," *ZTE Communications*, vol. 21, no. 1, pp. 15–24, Mar. 2023. doi: 10.12142/ZTECOM.202301003.

1 Introduction

With the popularity of big data techniques, machine learning has promoted wide applications in artificial intelligence fields, such as the smart IoT[1-2], smart industry[3-4], and autonomous driving[5-6]. Nowadays, due to the emergence of data protection regulations, like General Data Protection Regulation (GDPR)[7] and California Consumer Privacy Act (CCPA)[8], users pay increasing attention to data privacy. Data privacy significantly hinders training data collection, which limits the development of machine learning. Federated learning (FL), as a collaborative machine learning paradigm, is considered a promising solution to the challenges and has attracted tremendous attention from industry and academia. Specifically, a typical framework of FL consists of a server and some users (i.e., data owners). In FL, to preserve data privacy, users only share the trained local models' parameters instead of sharing raw data.

In spite of the benefits, there are two challenges in designing such an FL scheme. The first one is that the gradient attack may lead to privacy leakage. Specifically, in the gradient attack, adversaries utilize user-shared model parameters to infer sensitive information from training data. Thus far, some works[9-10] have been proposed to utilize the gradient leak attack to compromise user privacy. For instance, ZHU et al.[10] introduced a gradient inversion attack scheme to reconstruct sensitive information from public shared gradients, where adversaries launch attacks by iteratively optimizing the dummy inputs and the corresponding labels. Followed by Ref. [10], some gradient attack schemes have been proposed[11-12]. For instance, to enhance the performance of gradient inversion attacks, ZHAO et al.[11] proposed a simple and effective gradient inversion attack. Their scheme improves the effectiveness of recovering label information by combining the mathematical analysis of the gradients. Subsequently, YIN et al.[12] extended the gradient inversion attack into FL applications that are more practical, e.g., high-resolution images with large batch-size. If gradient attacks are not considered well in designing FL schemes, user privacy will incur serious threats. Therefore, users will be reluctant to participate in these applications, which significantly hinders the development of FL. The sec-

This work was supported in part by the Fundamental Research Funds for Central Universities under Grant No.2022RC006, in part by the National Natural Science Foundation of China under Grant Nos.62201029 and 62202051, in part by the BIT Research and Innovation Promoting Project under Grant No. 2022YCXZ031, in part by the Shandong Provincial Key Research and Development Program under Grant No. 2021CXGC010106, and in part by the China Postdoctoral Science Foundation under Grant Nos.2021M700435, 2021TQ0042, 2021TQ0041, BX20220029 and 2022M710007.
ZHANG Weiting and LIANG Haotian contribute equally in this work.
Corresponding author: ZHANG Chuan

ond challenge is that users with low-quality data decrease the performance of FL. In practical applications, the data quality of different users is usually uneven due to various reasons (e.g., device quality and education level)[13]. For example, users with high-quality devices usually own superior data, while users with low-quality devices have poorer data. If anomalous users are not identified in the training process, they will impair the performance of FL and even lead to the unavailability of FL models. Thus, it is also crucial to identify anomalous users and reduce their negative influence on the FL training process.

In recent years, to deal with the gradient attacks and preserve user privacy in FL, some solutions[14-16] have been proposed. Particularly, based on their cryptographic tools, these schemes can be categorized into three classes, i.e., secure multi-party computation (SMC) based schemes, homomorphic encryption (HE) based schemes, and differential privacy (DP) based schemes. DP-based FL schemes address the privacy leakage issues by adding noise[14]. However, the introduction of noise unavoidably reduces the model accuracy, hindering the applications of FL. To preserve user privacy, some SMC-based schemes[15] are proposed without compromising model accuracy. However, frequent user interaction introduces tremendous resource overhead to users and the server. To make a trade-off among the model's accuracy, user privacy, and resource overhead, some HE-based FL schemes are proposed[16].

Unfortunately, most existing privacy-preserving FL schemes ignore anomalous users. To address the challenge, several works[17-18] have been proposed to identify anomalous users and reduce their impacts. Specifically, ZHAO et al.[17] utilized the differential privacy technique and function mechanism to enable privacy-preserving FL. In their scheme, the server is allowed to access each user's data quality for identifying anomalous users. However, in practice, the user's data quality should be private. Once the data quality is disclosed to the server, it will lead to discrimination in the training process, which significantly reduces the users' enthusiasm to participate in FL. To preserve data quality information when identifying anomalous users, XU et al.[18] designed a framework to support privacy-preserving FL by introducing a non-colluding two-cloud model. In their scheme, additively homomorphic cryptosystem and YAO's garbled circuits are utilized to evaluate user data quality without compromising user privacy. It is hard to find two non-colluding clouds in practice, thereby limiting its implementation in real-world applications. Moreover, it also ignores the problem of user collusion. In FL, users may collude with each other to infer others' sensitive information. Therefore, a privacy-preserving FL scheme with anomalous user identification and user collusion resistance deserves to be investigated.

To solve the challenges, we propose a reliable and privacy-preserving FL (RPPFL) scheme based on the single-cloud model. The comparison results of RPPFL and other existing works are shown in Table 1. To identify anomalous users, RPPFL evaluates data quality without compromising user privacy. Particularly, we epitomize the contributions as follows:

▼Table 1. Comparison of RPPFL and other existing works

	User Privacy Preservation	Robust to User Instability	Support for Anomalous Users	Collusion Resistance	Server Setting
PPDL[16]	√	×	×	×	Single-cloud
PPML[19]	√	√	×	√	Single-cloud
SecProbe[17]	√	×	√	√	Single-cloud
PPFDL[18]	√	√	√	×	Two non-colluding clouds
RPPFL	√	√	√	√	Single-cloud

PPDL: privacy-preserving deep learning
PPFDL: privacy-preserving federated deep learning
PPML: privacy-preserving machine learning
RPPFL: reliable and privacy-preserving federated learning

- We first discover the challenges in designing a privacy-preserving FL scheme that supports anomalous identification. Then, to resolve these challenges, we design a reliable and privacy-preserving FL scheme named RPPFL, which is also resilient to user collusion attacks.
- We adopt the truth discovery technique to evaluate data quality. Subsequently, we utilize the (p, t) threshold Paillier cryptosystem to strengthen RPPFL to protect user privacy from being exposed and defend against user collusion attacks.
- Formal analysis proves the security of RPPFL. Then, based on the open datasets MNIST and CIFAR-10, extensive experiments are conducted to demonstrate that RPPFL is practically efficient and effective.

In this paper, the remainder is established as follows. In the next section, we illustrate the related models and security requirements of our construction. The preliminaries are reviewed in Section 3, and the detailed construction is presented in Section 4. Section 5 provides the security analysis. The experiments are given in Section 6, and Section 7 discusses the related works. Section 8 concludes the paper.

2 Models and Security Requirements

We first present the system model and threat model of RPPFL. After that, based on the threat model, we give the security requirements. To have a better understanding, we list some frequently used notations that appear in RPPFL, which is shown in Table 2.

2.1 System Model

As we can see in Fig. 1, the system model of RPPFL consists of an aggregation server and several users.

- The aggregation server is an entity with strong computing and storage capabilities. To reduce the anomalous users' negative impacts on the accuracy of the model, the aggregation server is allowed to identify users' data quality (i.e., user reliability). Then, with the user's reliability and local gradients, the aggregation server aggregates the global gradients in a privacy-preserving manner. Subsequently, global gradients

will be sent to the users.

• The users are entities holding different datasets that can be utilized to train FL models. To get models with better performance, they cooperate in training models with the help of an aggregation server. Instead of sharing datasets directly, they share the gradients of local models. To protect gradient privacy, users first encrypt local gradients with an additively homomorphic cryptosystem. Then, users send them to the aggregation server and update local models after receiving global gradients from the aggregation server.

2.2 Threat Model

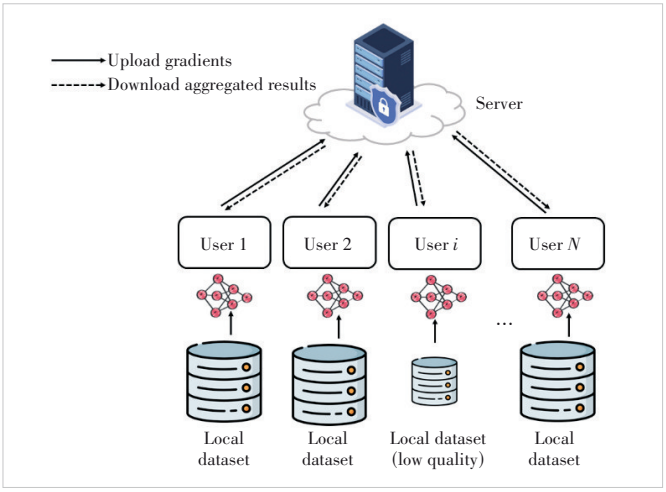
In our scenario, like previous works[20-21], we presume that the aggregation server and all users are honest-but-curious. That is, the server will faithfully obey the designed procedures to accomplish its task. However, it may try to retrieve others' sensitive information using prior acquired knowledge. Besides, we presume that the aggregation server will not collude with users and there are at most $t-1$ users colluding. Then, we mainly consider the following two adversaries.

1) The aggregation server may try to deduce users' local gradients and reliability according to the information it acquired.

2) The user may try to infer the information of his/her reliabilities according to the information he/she acquired.

▼Table 2. Frequently used notations

Notation	Meaning
n	A large positive integer
$\mathbb{Z}_n$	The set of integers modulo n
$\mathbb{Z}_n^*$	The multiplicative group of reversible elements of $\mathbb{Z}_n$
N	The number of users
K	The number of the selected users
M	The number of gradient types
M_f	A big integer of the magnitude of 10
x_m^k	The m-th gradient of the k-th user
$\widetilde{x_m^k}$	The integer corresponding to the enlargement of x_m^k
x_m^*	The aggregated result of the m-th gradient
R_k	The reliability (indicates the data quality) of the user k
$\mathbb{C}$	The coefficient used to amplify users' reliability
sk_k	The secret key of the selected user k
sk_{N+1}	The secret key of the aggregation server
$\mathrm{Enc}_{\mathrm{pk}}(\cdot)$	The ciphertext encrypted by a public key
r_k	The random value selected by the user k



▲Figure 1. System model of reliable and privacy-preserving federated learning (RPPFL)

2.3 Security Requirements

On the basis of system and threat models, we have developed the following security requirements.

1) User's local gradient privacy. To effectively preserve user privacy, the user's local gradients should be sent to the aggregation server in the ciphertext, which prevents the adversary (e.g., the server) from recovering the user's sensitive information from the shared gradients and global parameters.

2) Privacy protection of reliability for users. To ensure the fairness of the FL process, all information related to the reliability of the user should be kept secret and unavailable to any participant, even to the user itself.

3 Preliminaries

In this section, we will illustrate the preliminaries about truth discovery, FL, and the additively homomorphic cryptosystem.

3.1 Truth Discovery

Truth discovery aims at estimating ground truth data from numerous heterogeneous data. In general, it is composed of two main steps: weight update and truth update.

1) Weight update

In this step, the weight of each user is computed based on the distance between their provided data and the ground truths. Without losing generality, we here assume the ground truths are fixed. Typically, each user's weight w_k can be computed as $w_k = f\left(\sum_{m=1}^{M} d(x_m^k, \boldsymbol{x}_m^*)\right)$, where f denotes a monotonically decreasing function, and $d(x_m^k, \boldsymbol{x}_m^*)$ is a distance function (i.e., the Euclidean distance). Therefore, if the provided data from a specific user are close to the ground truth, the user's weight will be assigned to a higher value.

2) Truth update

In this step, on the basis of each user's weight, the ground truth is estimated according to Eq. (1):

$$\boldsymbol{x}_m^* = \frac{\sum_{k=1}^{K} x_m^k \cdot w_k}{\sum_{k=1}^{K} w_k}. \tag{1}$$

In the case of continuous data, $\boldsymbol{x}_m^*$ means the estimated

ground truth. As for the categorical data, $\boldsymbol{x}_m^*$ represents a probability vector. Each element in the vector means the probability of a specific answer being the truth[22].

3.2 Federated Learning

As a collaborative learning paradigm, FL intends to train models based on data from distributed users. The basic training process of FL is shown below.

1) Selecting users

Assume there exist N users, each holding a local dataset $\mathcal{D}_j, j \in [1,N]$, which is derived from the whole training dataset $\mathcal{D}=\{(u_i,v_i); i = 1,2,\cdots,M\}$, where $\mathcal{D} = \bigcup_{j \in [1,N]} \mathcal{D}_j$. For each epoch $t \in \{1,2,\cdots\}$ in FL, the aggregation server chooses K users at random, where $K < N$.

2) Local training

Each selected user $k, k \in [1, K]$, randomly chooses a small batch of dataset B^k. Then, they leverage stochastic gradient descent (SGD), a commonly used optimization algorithm, to calculate gradients over their local datasets. Specifically, we let $\boldsymbol{u}_i^k$ and $\boldsymbol{v}_i^k$ denote the feature vector and its corresponding label in B^k, respectively, and θ_t^k denotes the parameters of the model in the current epoch. The loss function, indicating the distance between prediction results and real labels, can be denoted as $L(\theta_t^k,\boldsymbol{u}_i^k,\boldsymbol{v}_i^k)$. Then, the gradient can be calculated as Eq. (2):

$$\nabla_{\theta_t^k} = \nabla L\left(B^k,\theta_t^k\right) = \frac{\sum_{<u_i,v_i> \in B^k} \nabla L\left(\theta_t^k,\boldsymbol{u}_i^k,\boldsymbol{v}_i^k\right)}{\left|B^k\right|}. \tag{2}$$

After that, $\nabla_{\theta_t^k}$ will be transmitted to the aggregation server.

3) Global aggregation

After receiving local gradients from all selected users, the aggregation server will aggregate the global gradients as Eq. (3):

$$\text{Global} = \frac{\sum_{k=1}^{K} \nabla_{\theta_t^k}}{K}. \tag{3}$$

Finally, the global gradients will be transmitted to the users to update their local model as:

$$\theta_{t+1}^k = \theta_t^k - \eta \cdot \text{Global}, \tag{4}$$

where η denotes the learning rate.

3.3 Additively Homomorphic Cryptosystem

The cryptosystem in RPPFL is on the basis of the (p,t)-threshold Paillier cryptosystem[22]. As a typical asymmetric cryptosystem, it utilizes the public key (pk) to encrypt the plaintexts and secret key (sk) to recover the plaintexts. Note that (p,t)-threshold Paillier cryptosystem splits the secret key into p parts, i.e., $(\text{sk}_1,\text{sk}_2,\cdots,\text{sk}_p)$, and sends them to p different parties. In (p,t)-threshold Paillier cryptosystem-based applications, any entity cannot decrypt the ciphertexts alone. That is, the ciphertext can only be decrypted if at least t entities cooperate together. Moreover, even if some users are dropped off during the process because of the insatiability, the ciphertext can still be recovered.

We use $\text{Enc}_{\text{pk}}(\cdot)$ to denote the ciphertexts encrypted by the public key. Then, assuming $m \in \mathbb{Z}_n$ denotes a plaintext, its corresponding ciphertext can be calculated as follows:

$$C = \text{Enc}_{\text{pk}}(m) = g^m r^n \bmod n^2, \tag{5}$$

where $r \in \mathbb{Z}_n^*$ is a randomly selected value and should be kept secret. For decryption, each party l, $l \in [1,p]$, requires to compute the partial decryption c_l according to Eq. (6) with the secret key sk_l,

$$c_l = c^{2\Delta sk_l}, \tag{6}$$

where we denote $\Delta = p!$. Based on the algorithm in Ref. [23], these partial decryptions can be composed together for decrypting the ciphertext C in order to recover the plaintext m.

Then, we further present additively homomorphic properties of our adapted cryptosystem. Specifically, given the ciphertexts of two plaintexts, $m_1,m_2 \in \mathbb{Z}_n$ are encrypted with the same public key:

$$\begin{aligned} C_1 &= \text{Enc}_{\text{pk}}(m_1) = g^{m_1} r_1^{\ n} \bmod n^2, \\ C_2 &= \text{Enc}_{\text{pk}}(m_2) = g^{m_2} r_2^{\ n} \bmod n^2. \end{aligned} \tag{7}$$

We have

$$\begin{aligned} \text{Enc}_{\text{pk}}(m_1 + m_2) &= \text{Enc}_{\text{pk}}(m_1) \cdot \text{Enc}_{\text{pk}}(m_2) \\ &= g^{m_1 + m_2}(r_1 r_2)^n \bmod n^2, \end{aligned} \tag{8}$$

$$\text{Enc}_{\text{pk}}(b \cdot m_1) = \text{Enc}_{\text{pk}}(m_1)^b = g^{bm_1} r_1^{bn} \bmod n^2, \tag{9}$$

where b denotes a constant.

4 Scheme Design and Details

In this section, we first illustrate the approach that we utilize to handle anomalous users. Then, we give the details of our proposed RPPFL.

4.1 Approach to Handling Anomalous Users

To decrease the negative influence of anomalous users on the trained model in federation learning, here we describe the mechanism Me_{AU}, which is inspired by the truth discovery[24]. In RPPFL, we assume that the data from different users are independently and equally distributed. We assume that each user holds M categories of gradients (in Section 3.2) after train-

ing on their local dataset. The m-th gradient of the k-th user can be represented as x_m^k, where $m \in [1, M], k \in [1, K]$. We use $\boldsymbol{x}_m^*$ to denote the global m-th gradient of K selected users. Additionally, we let R_k represent the reliability (indicates the data quality) of the user k. Me_{AU} mainly includes two phases: updating the user's reliability and updating global gradients.

1) Update user's reliability

The user's reliability will be given a high value when the calculated gradient is close to the global gradient from the server. Specifically, given the global gradient $\boldsymbol{x}_m^*$, the reliability of user k is calculated as follows:

$$R_k = f\left(\sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)\right), \tag{10}$$

where f denotes a monotonically decreasing function, and $d(\cdot)$ denotes a function that measures the distance between the local gradients and global gradients. In RPPFL, we use the same method as in Ref. [18], and formulate Eq. (10) as:

$$R_k = \frac{\mathbb{C}}{\sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)}, \tag{11}$$

where $\mathbb{C}$ is used to amplify users' reliability, which is calculated according to Eq. (12):

$$\mathrm{C} = \chi^2_{\left(1-\frac{\alpha}{2}, |M|\right)}, \tag{12}$$

where χ denotes the Chi-squared distribution, and α represents its corresponding significance level. It is noteworthy that when the value of α and M (the number of gradients) is determined, the coefficient $\mathbb{C}$ can be regarded as a constant. On the basis of some proposed works[18, 25–26], for users with high-quality data for training, the obtained gradients are always consistent in the direction of the vector with high probability. To guarantee the convergence of training, the direction of the local gradient x_m^k is always required as the same with the global gradient x_m^*. Thus, we compute $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^2$ if x_m^k and x_m^* are both positive or negative. If not, we set $d(x_m^k, x_m^*)$ to a large positive integer (illustrated in Section 4.2).

2) Update global gradients

With the reliability of each user given, the aggregated result of m-gradient is calculated as

$$x_m^* = \frac{\sum_{k=1}^{K} R_k x_m^k}{\sum_{k=1}^{K} R_k}. \tag{13}$$

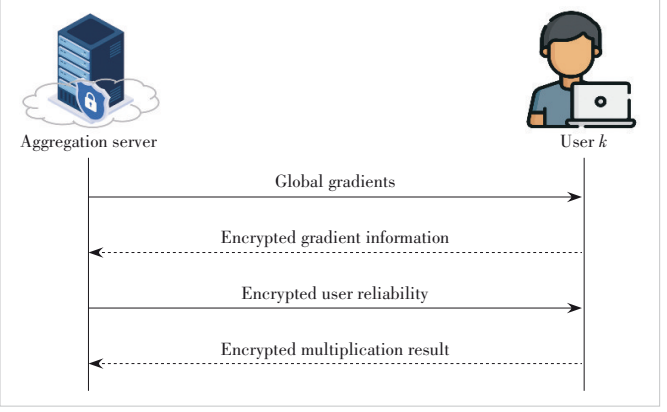
Note that we do not directly remove these anomalous users. The reason is that the reliability information is kept secret from all participants, even the users themselves, to prevent discrimination during the training phase. The existence of low-quality data is inevitable. In some rare cases where all users are normal, there is still the possibility that the trained model will be overfitted in the actual prediction. Based on the above facts, RPPFL tolerates gradients from anomalous users but ensures that the global gradients are mainly contributed by normal users. However, ensuring that each participant in federated learning is unaware of users' reliability will inevitably increase the difficulty of reducing the impacts of low-quality data.

4.2 Reliable and Privacy-Preserving Federated Learning

As shown below, we first briefly summarize the main process of RPPFL, i.e., reliability identification and gradient aggregation, and then give its details. The workflow of RPPFL is displayed in Fig. 2, and the protocol framework is shown as Protocol 1. We assume that a trusted third party (TTP) has executed the (p,t)-threshold Paillier cryptosystem before running the reliable and privacy-preserving federated learning protocol, where $p = N + 1$ and $t = K + 1$. The secret keys $(\mathrm{sk}_1, \mathrm{sk}_2, \cdots, \mathrm{sk}_N)$ are sent to N different users, respectively, and sk_{N+1} is sent to the aggregation server. Besides, the public key is distributed to all entities.

• Reliability identification. In this step, each selected user first calculates the Euclidean distance between its local gradients and the global gradients from the aggregation server. These calculation results will be encrypted using the public key and then transmitted to the aggregation server. With these ciphertexts, the aggregation server calculates the reliability of each user while protecting data privacy. Ultimately, the encrypted reliability will be sent to the corresponding user for the following procedure.

• Gradient aggregation. In this phase, each user calculates the product of their gradient and reliability in the encryption domain. These ciphertexts are transmitted to the server. With the help of K selected users, the server decrypts these received ciphertexts and subsequently updates



▲ **Figure 2. Workflow of reliable and privacy-preserving federated learning (RPPFL)**

the global models.

Note that the additively homomorphic cryptosystem is defined over the integer ring. However, the gradient often consists of many floating-point numbers in real-world federated learning. We define a big integer M_f, which is a magnitude of 10. Before utilizing homomorphic encryption on the gradient x_m^k, we calculate $\lfloor M_f \cdot x_m^k \rfloor$, which we denote as $\widetilde{x_m^k}$. $\widetilde{x_m^k}$ is the rounded version of the gradient for encryption, and the original approximated result can be easily recovered by simply dividing $\widetilde{x_m^k}$ with M_f. Unless otherwise mentioned, we also use this format to represent other rounded values in the remaining parts of the paper. Then, for each negative integer x_m^k, we use the trick adopted in Ref. [27] by simply replacing it with its inverse in the cryptosystem.

The update of the global models in federated learning lasts for several iterations. Here, we give the calculation procedure in one of the iterations.

1) Reliability identification

Step 1: The aggregation server first selects K users and sends the global gradient $\{x_m^*\}_{m=1}^M$ to them. If it is in the first iteration, $\{x_m^*\}_{m=1}^M$ is the random value initialized by the aggregation server; otherwise, $\{x_m^*\}_{m=1}^M$ is derived in the previous iteration. Upon receiving $\{x_m^*\}_{m=1}^M$, the user $k, k \in [1,K]$, calculates:

$$\mathrm{D} = \sum_{m=1}^{M} d\left(x_m^k, x_m^*\right), \tag{14}$$

and obtains its reciprocal, i.e., $\mathbb{D}^{-1}$. Then, to preserve the privacy of $\mathbb{D}^{-1}$, the user $k, k \in [1,K]$, chooses a random value $r_k \in \mathbb{Z}_n^*$ and encrypts it as follows:

$$\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right) = g^{\widetilde{\mathbb{D}^{-1}}} r_k^n \bmod n^2. \tag{15}$$

When the encryption is completed, each user sends $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$ to the aggregation server.

Step 2: After receiving $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$ from all selected K users, the aggregation server calculates the reliability of each user in ciphertexts as

$$\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left\lfloor M_f \cdot \frac{1}{\mathbb{D}} \right\rfloor \cdot \left\lfloor M_f \cdot \mathbb{C} \right\rfloor\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left\lfloor M_f \cdot \frac{1}{\mathbb{D}} \right\rfloor\right)^{\left\lfloor M_f \cdot \mathbb{C} \right\rfloor} = g^{\tilde{\mathbb{C}} \widetilde{\frac{1}{\mathbb{D}}}} r_k^{\tilde{\mathbb{C}} n} \bmod n^2, \tag{16}$$

where the aggregation server calculates $\mathbb{C}$ and keeps it secretly. $\left[\widetilde{\cdot}\right]$ denotes the product of two rounded values. After that, the aggregation server transmits the encrypted reliability $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$ to user $k, k \in [1,K]$.

Protocol 1. Reliable and privacy-preserving federated learning

Input:

K selected users, M types of gradients, local gradients $\{x_m^k\}_{m,k=1}^{M,K}$, initialized global gradients $\{x_m^*\}_{m=1}^M$, and coefficient $\mathbb{C}$

Output:

Global gradients $\{x_m^*\}_{m=1}^M$

1. The aggregation server sends $\{x_m^*\}_{m=1}^M$ to each user k.
2. Each user k computes the local gradients.
3. Each user k computes $\mathrm{Enc}_{\mathrm{pk}}(\mathbb{D}^{-1})$, where $\mathbb{D}^{-1} = 1/\sum_{m=1}^{M} d(x_m^k, x_m^*)$.
4. Each user k sends $\mathrm{Enc}_{\mathrm{pk}}(\mathbb{D}^{-1})$ to the aggregation server.
5. The aggregation server computes $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}])$ for each user k.
6. The aggregation server sends $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}])$ back to each user k.
7. Each user k computes the product of local gradients and their reliability $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}] \cdot \widetilde{x_m^k})$, $m \in [1,M]$.
8. Each user k sends $\mathrm{Enc}_{\mathrm{pk}}([\widetilde{R_k}] \cdot \widetilde{x_m^k})$, $m \in [1,M]$ to the aggregation server.
9. The aggregation server computes $\mathrm{Enc}_{\mathrm{Global}}$ and $\mathrm{Enc}_{\mathrm{pk}}\left([\sum_{k=1}^{K} \widetilde{R_k}]\right)$.
10. The aggregation server computes $\{x_m^*\}_{m=1}^M$ according to Eqs. (19) and (20).
11. Repeat steps 3 – 7 until the convergence criteria in FL is reached.

2) Gradient aggregation

Once the reliability of each user has been obtained, the next step is to update the global gradients according to the reliability and local gradients of all selected users.

Step 1: After receiving $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$ from the aggregation server, the user k calculates the product of local gradients and their reliability in ciphertexts

$$\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right) = \mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)^{\widetilde{x_m^k}} = g^{\widetilde{x_m^k}\left[\widetilde{R_k}\right]} r_k^{\widetilde{x_m^k} n} \bmod n^2. \tag{17}$$

Then, $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$ will be transmitted to the aggregation server.

Step 2: When the aggregation server receives the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$, $k \in [1,K]$, from all selected users, it aggregates them in ciphertexts according to the homomorphic property of the (p,t)-threshold Paillier cryptosystem.

$$\mathrm{Enc}_{\mathrm{Global}} = \prod_{k=1}^{K} \mathrm{Enc}_{pk}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right) = g^{\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)}\left(\prod_{k=1}^{K} r_k\right)^n \bmod n^2 = \mathrm{Enc}_{pk}\left(\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)\right). \tag{18}$$

After that, $\mathrm{Enc}_{\mathrm{Global}}$ is sent to K selected users. Each user k uses their secret key sk_k to partially decrypts $\mathrm{Enc}_{\mathrm{Global}}$ and then sends them to the aggregation server. The aggregation server first obtains the partial decryption with its secret key sk_{N+1}. Then, based on $K+1$ partially decrypted ciphertexts, the aggregation server recovers the plaintexts $\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)$. Similarly, the aggregation server can also calculate the summation of each user's reliability, i. e., $\sum_{k=1}^{K}\left[\widetilde{R_k}\right]$. Therefore, the global gradients can be updated as:

$$\widetilde{x_m^*} = \frac{\sum_{k=1}^{K}\left(\left[\widetilde{R_k}\right] \cdot \widetilde{x_m^k}\right)}{\sum_{k=1}^{K}\left[\widetilde{R_k}\right]}, \tag{19}$$

which will be sent to K users to update their local models. Note that x_m^* can be recovered by calculating

$$x_m^* = \left\lfloor \widetilde{x_m^*} / \left(M_f\right) \right\rfloor. \tag{20}$$

Reliability identification and gradient aggregation are performed iteratively until the convergence criteria are fulfilled.

5 Security Analysis

Based on the threat model in Section 2.2, the potential threats mainly come from the entities (i.e., users and the aggregation server). Thus, the objective of RPPFL is to protect the user's local gradient and the user's reliability from being exposed to any entity in RPPFL. Furthermore, it should also be resilient to the user collusion attack. Here, we prove the security of RPPFL by giving Theorem 1, followed by the corresponding proof.

Theorem 1. Assuming that the aggregation server is non-colluding with users and there are at most $t-1$ users colluding, neither the user's local gradient nor the user's reliability will be leaked to any entity in RPPFL.

Proof. First, we prove that each user cannot infer their own reliability from the information they have acquired and the ciphertexts returned by the aggregation server. Next, we show that the aggregation server cannot infer each user's local gradient and reliability from the information it holds and the ciphertexts returned by the user.

The user knows the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right]\right)$, $\mathrm{Enc}_{\mathrm{Global}}$, and plaintexts $\{x_m^*\}_{m=1}^{M}$, $\mathbb{D} = \sum_{m=1}^{M} d\left(x_m^k, x_m^*\right)$. Since there are at most $t-1$ users colluding, the user cannot recover the secret key (sk), from sk_k. Additionally, the (p,t)-threshold Paillier cryptosystem has already been demonstrated to defend against chosen-plaintext attacks[22]. Therefore, the user cannot decrypt these ciphertexts. With the global gradient $\{x_m^*\}_{m=1}^{M}$, the user calculates $\mathbb{D}$ locally. However, since $\mathbb{C}$ is only known by the aggregation server. Without knowing $\mathbb{C}$, it is impossible for the user to acquire its reliability.

For the aggregation server, it knows the ciphertexts $\mathrm{Enc}_{\mathrm{pk}}\left(\widetilde{\mathbb{D}^{-1}}\right)$, $\mathrm{Enc}_{\mathrm{pk}}\left(\left[\widetilde{R_k}\right] \cdot x_m^k\right)$, and plaintexts $\mathbb{C}$, $\sum_{k=1}^{K}\left(R_k \cdot x_m^k\right)$, $\sum_{k=1}^{K} R_k$. Since the (p,t)-threshold Paillier cryptosystem has been demonstrated to defend against chosen-plaintext attacks, the aggregation server cannot recover the secret key, and thus cannot decrypts these ciphertexts. As for $\mathbb{C}$, without the plaintexts $\mathbb{D}$, the aggregation server cannot obtain the users' reliabilities. Although the aggregation server knows the sum of K users' reliabilities, i. e., $\sum_{k=1}^{K} R_k$, it is impossible to identify the individual reliability of each user without knowing other information. Similarly, it is also impossible to separate the individual reliability and model weight from $\sum_{k=1}^{K}\left(R_k \cdot x_m^k\right)$.

Therefore, RPPFL can prevent the user's local gradient and reliabilities from disclosing to other entities. Moreover, for the user collusion attack, the properties of the Paillier cryptosystem ensure the safety of the scheme when there are no more than $t-1$ users colluding.

6 Experiments

In this section, we perform experiments to observe the performance of RPPFL. The FL framework is built via PyTorch with Cuda 10.2, which runs on the server with two Nvidia Tesla-P40 GPUs for hardware and RedHat for the operating system. For the cryptosystem, we utilize the Paillier library for implementation, and the running environment is Java 18.0. Moreover, we choose MNIST and CIFAR-10 as the datasets in FL, which are commonly used in many scenarios. As for the users in FL, they are all equipped with the same convolutional neural network (CNN) to calculate local gradients with the use of their local data. The model in the experiments is inspired by LeNet widely used in various situations. Finally, as for the hyper-parameters, the learning rate is set to 0.001, while the batch size is 128.

6.1 Accuracy Performance

In this part, we observe the accuracy performance of RPPFL. As mentioned before, many attributes influence the model's accuracy. Here, we mainly focus on the impact of the number of users and the number of gradients per user. With-

out losing generality, we set the dataset $\mathcal{D}_i$ for each user k in the same size. Meanwhile, to construct low-quality data for anomalous users, we replace a fixed proportion of their original data with random noises $\epsilon \in [0,1]$. The ratio of the replaced data is set to 20% in our experiments.

1) Number of users

We first illustrate the influence of the number of users that take part in the training process. To better demonstrate the performance of RPPFL, we take two related works[18, 28] for comparison.

Fig. 3 displays the comparison of accuracy based on a different number of users, where the number of gradients for each user is set to 2 500. The figure demonstrates that the increment in the number of users in RPPFL does improve the model accuracy because more data from corresponding users contribute to the trained model. Moreover, for both the MNIST dataset in Fig. 3(a) and CIFAR-10 dataset in Fig. 3(b) , the accuracy of RPPFL is about the same as PPFDL in Ref. [18] and outperforms that in Ref. [28]. Therefore, we can reach the conclusion that RPPFL can ensure the aggregation gradients are mainly contributed by users with data of high quality.

2) Number of gradients per user

We then discuss the influence of the number of gradients for each user on accuracy performance.

Fig. 4 demonstrates that the model accuracy will also improve when the number of gradients increases. It is evident that more involved gradients in the FL training procedure will boost the convergence rate and make the model more accurate. From Figs. 4(a) and 4(b), the performance of RPPFL is still better than the schemes in Refs. [28] and [18]. In conclusion, RPPFL ensures that the user with high-quality data is rewarded with high reliability and guarantees that the aggregation result is mainly contributed by these users.

6.2 Efficiency

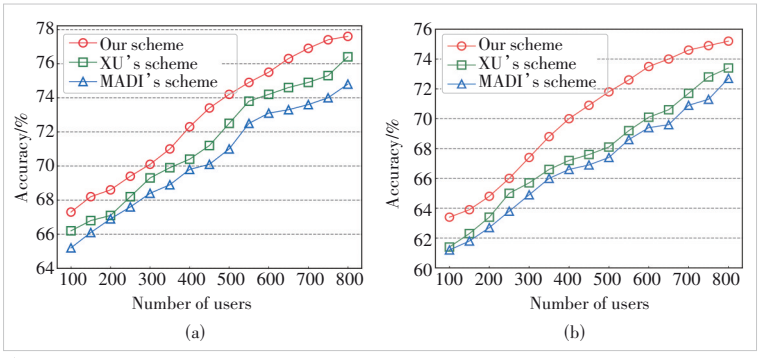
In this part, we observe the efficiency performance of RPPFL. For simplicity, we here only discuss and visualize the efficiency in the aggregation phase of FL. To keep fairness, we test the schemes in Refs. [28] and [18] on the same platform (hardware and software) for RPPFL. Specifically, the CNN network is the same for every user, and other hyper-parameters remain the same.

Fig. 5(a) demonstrates the computational cost for different user numbers, while Fig. 5(b) presents the one for different gradient numbers per user. It can be observed that with the growth of the number of users and the number of gradients per user, the aggregation time increases for all the schemes. Moreover, RPPFL has better efficiency than the one in Ref. [28]. As we can see, the RPPFL is moderately inferior to the one in Ref. [18]. It is because the PPFDL in Ref. [18] adopts a two-cloud model, where the computational costs are shared between the two cloud servers, while RPPFL is established on a single cloud model. However, PPFDL requires two non-colluding cloud servers, which is not practical in real-world scenarios compared with RPPFL.

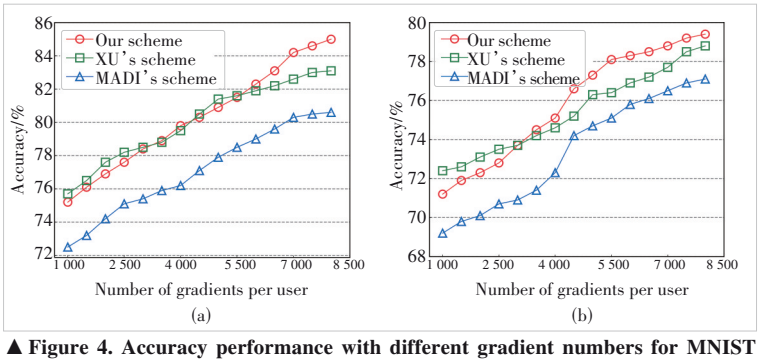
7 Related Works

In this section, we illustrate some related works of privacy-preserving federated learning.

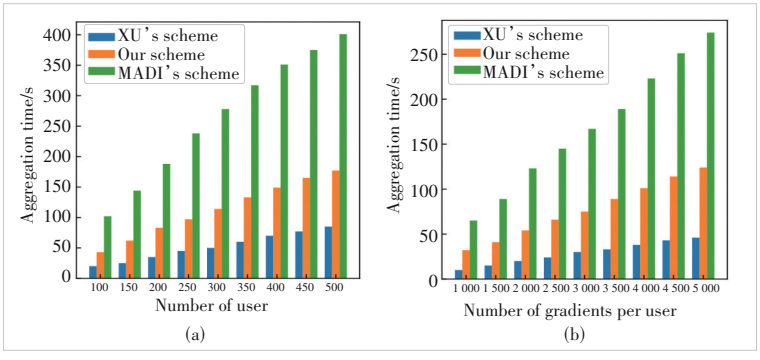
Since the proposal of the original FL, many schemes have been designed to preserve data privacy in FL based on privacy-preserving techniques. These techniques can be mainly divided into three categories: differential privacy, secure multi-party computation, and homomorphic encryption. As for the differential privacy, the authors in Ref. [29] proposed a



▲ Figure 3. Accuracy performance with different user numbers for MNIST and CIFAR-10 datasets



▲ Figure 4. Accuracy performance with different gradient numbers for MNIST and CIFAR-10 datasets



▲ Figure 5. Computational costs for different schemes

mechanism that set different proportions of selected parameters to preserve data privacy while preserving training accuracy. In 2016, ABADI et al.[30] leveraged differential privacy with a moderate privacy budget to learn models of deep neural networks. When it comes to secure multi-party computation, the authors in Ref. [19] proposed a safe and practical aggregation protocol in the FL training process. SMC was adopted to ensure the privacy of the users' gradients shared with the aggregation server. In 2018, JAYARAMAN et al.[31] introduced a distributed learning method that combines DP with SMC. Moreover, because the users' access to power and network bandwidth is always under a particular constraint in real-world scenarios, secret sharing and key exchange protocols are also considered to enhance the robustness of FL. Authors in Ref. [32] proposed a scheme leveraging the secret key-sharing technique to protect privacy in FL while verifying the integrity of aggregation results. For homomorphic encryption, in 2018, PHONE et al.[16] presented a system for privacy-preserving collaborative deep learning. It utilizes Learning with Errors (LWE)-based homomorphic encryption to secure the privacy of publicly shared model parameters among the participants. Furthermore, the authors in Ref. [20] designed high-efficiency protocols by adopting secure two-party computation, which was established on the two-server model (non-collusion). In 2021, MADI et al.[28] presented a scheme with a combination of homomorphic encryption and verifiable computing. The aim was to execute a federated averaging operator directly in the ciphertext and prove that the operator is correctly executed.

In conclusion, homomorphic encryption can be applied for privacy-preserving federated learning according to its property of addition and multiplication in the ciphertext domain. However, the enormous computational burden is unacceptable in scenarios that exist plenty of users or training data with large dimensions. Although SMC is better that HE in terms of computational costs, it always needs many interactions among entities. This brings a high communication burden and a lack of robustness. Compared with the other two techniques, differential privacy performs better in cost. But a balance between privacy and accuracy should always be considered. Ref. [33] demonstrated that if the model accuracy was acceptable, adversaries could still reconstruct the user's private data. Authors in Ref. [34] successfully leveraged a generative adversarial network (GAN) to violate data privacy even if all shared parameters were protected by differential privacy. Therefore, combining the advantages of different privacy-preserving mechanisms while overcoming their drawback has raised much concern for researchers.

Moreover, all these solutions mentioned above fail to consider the problem of anomalous users. To tackle this problem, SecProbe was proposed[17] as the first solution to handling anomalous users in collaborative deep learning while protecting data privacy. It utilized techniques based on DP to perturb the objective function of the target network. However, Ref. [34] showed that the current mechanism of DP can hardly reach an acceptable balance between security and accuracy. XU et al.[18] designed PPFDL with the leverage of additively homomorphic cryptosystem and garbled circuits. However, their system structure is based on the two-cloud model, and it requires two non-colluding cloud servers. Therefore, such limitation makes their scheme impractical in many real-world situations like edge computing. Moreover, their PPFDL is also vulnerable to user collusion attacks.

8 Conclusions

In this paper, we propose RPPFL, a reliable and privacy-preserving federated learning scheme. RPPFL uses a truth discovery technique to identify each user's reliability according to their data quality and thereby reduce the contribution of anomalous users on the global models. Specifically, we leverage an additively homomorphic cryptosystem to enrich the truth discovery technique to provide comprehensive privacy protection (e.g., model privacy and data quality privacy) and user collusion resistance. Security analysis demonstrates the security of RPPFL. Experimental results of two different real-world datasets indicate that RPPFL has acceptable performance on both accuracy and efficiency. For future work, considering that the user may infer data information of others with the global gradients, we will focus on designing a reliable and privacy-preserving federated learning scheme that can protect the privacy of gradients on both the aggregation server side and the user side.

References

[1] WANG J S, LIU Y, ZHANG W T, et al. ReLFA: resist link flooding attacks via renyi entropy and deep reinforcement learning in SDN-IoT [J]. China communications, 2022, 19(7): 157 – 171. DOI: 10.23919/JCC.2022.07.013
[2] KANG J W, LI X D, NIE J T, et al. Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things [J]. IEEE transactions on network science and engineering, 2022, 9(5): 2966 – 2977. DOI: 10.1109/TNSE.2022.3178970
[3] ZHANG W T, YANG D, WU W, et al. Spectrum and computing resource management for federated learning in distributed industrial IoT [C]//Proceedings of 2021 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2021: 1 – 6. DOI: 10.1109/ICCWorkshops50388.2021.9473515
[4] ZHANG W T, YANG D, WU W, et al. Optimizing federated learning in distributed industrial IoT: A multi-agent approach [J]. IEEE journal on selected areas in communications, 2021, 39(12): 3688 – 3703. DOI: 10.1109/JSAC.2021.3118352
[5] PENG H X, SHEN X M. Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks [J]. IEEE journal on selected areas in communications, 2021, 39(1): 131 – 141. DOI: 10.1109/JSAC.2020.3036962
[6] PENG H X, WU H Q, SHEN X S. Edge intelligence for multi-dimensional resource management in aerial-assisted vehicular networks [J]. IEEE wireless communications, 2021, 28(5): 59 – 65. DOI: 10.1109/MWC.101.2100056
[7] European Union. General data protection regulation [EB/OL]. [2022-10-28]. https://gdpr-info.eu/
[8] State of California Department of Justice. California consumer privacy act [EB/OL]. [2022-10-28]. https://oag.ca.gov/privacy/ccpa
[9] SONG C Z, RISTENPART T, SHMATIKOV V. Machine learning models that

remember too much [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 587 – 601. DOI: 10.1145/3133956.3134077
[10] ZHU L G, LIU Z J, HAN S. Deep leakage from gradients Advances [EB/OL]. [2022-10-28]. https://doi.org/10.1007/978-3-030-63076-8_2
[11] ZHAO B, K R MOPURI, H BILEN. iDLG: improved deep leakage from gradients [EB/OL]. [2022-10-28]. https://arxiv.org/abs/2001.02610
[12] YIN H X, MALLYA A, VAHDAT A, et al. See through gradients: image batch recovery via gradinversion [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2021: 16332 – 16341
[13] ZHANG C, ZHAO M Y, ZHU L H, et al. FRUIT: a blockchain-based efficient and privacy-preserving quality-aware incentive scheme [J]. IEEE journal on selected areas in communications, 2022, 40(12): 3343 – 3357. DOI: 10.1109/JSAC.2022.3213341
[14] OUADRHIRI A E, ABDELHADI A. Differential privacy for deep and federated learning: a survey [J]. IEEE access, 2022, 10: 22359 – 22380. DOI: 10.1109/ACCESS.2022.3151670
[15] PEYVANDI A, MAJIDI B, PEYVANDI S, et al. Privacy-preserving federated learning for scalable and high data quality computational-intelligence-as-a-service in Society 5.0 [J]. Multimedia tools and applications, 2022, 81(18): 25029 – 25050. DOI: 10.1007/s11042-022-12900-5
[16] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning via additively homomorphic encryption [J]. IEEE transactions on information forensics and security, 2018, 13(5): 1333 – 1345. DOI: 10.1109/TIFS.2017.2787987
[17] ZHAO L C, WANG Q, ZOU Q, et al. Privacy-preserving collaborative deep learning with unreliable participants [J]. IEEE transactions on information forensics and security, 2020, 15: 1486 – 1500. DOI: 10.1109/TIFS.2019.2939713
[18] XU G W, LI H W, ZHANG Y, et al. Privacy-preserving federated deep learning with irregular users [J]. IEEE transactions on dependable and secure computing, 2022, 19(2): 1364 – 1381. DOI: 10.1109/TDSC.2020.3005909
[19] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 1175 – 1191. DOI: 10.1145/3133956.3133982
[20] MOHASSEL P, ZHANG Y P. SecureML: a system for scalable privacy-preserving machine learning [C]//Proceedings of 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 19 – 38. DOI: 10.1109/SP.2017.12
[21] ZHENG Y F, DUAN H Y, WANG C. Learning the truth privately and confidently: encrypted confidence-aware truth discovery in mobile crowdsensing [J]. IEEE transactions on information forensics and security, 2018, 13(10): 2475 – 2489. DOI: 10.1109/TIFS.2018.2819134
[22] MIAO C L, JIANG W J, SU L, et al. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems [C]//Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ACM, 2015: 183 – 196. DOI: 10.1145/2809695.2809719
[23] DAMGARD I, JURIK M. A generalisation, a simplification and some applications of paillier's probabilistic public-key system [M]. Public Key Cryptography. Berlin, Heidelberg: Springer Berlin, 2001: 119 – 136. DOI: 10.1007/3-540-44586-2_9
[24] LI Y L, GAO J, MENG C S, et al. A survey on truth discovery [J]. ACM SIGKDD explorations newsletter, 2016, 17(2): 1 – 16. DOI: 10.1145/2897350.2897352
[25] SMITH V, CHIANG C K, SANJABI M, et al. Federated multi-task learning [EB/OL]. [2022-10-28]. https://arxiv.org/abs/1705.10467
[26] WANG L P, WANG W, LI B. CMFL: mitigating communication overhead for federated learning [C]//Proceedings of 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 954 – 964. DOI: 10.1109/ICDCS.2019.00099
[27] XU G W, LI H W, TAN C, et al. Achieving efficient and privacy-preserving truth discovery in crowd sensing systems [J]. Computers & security, 2017, 69: 114 – 126. DOI: 10.1016/j.cose.2016.11.014
[28] MADI A, STAN O, MAYOUE A, et al. A Secure Federated Learning framework using Homomorphic Encryption and Verifiable Computing [C]//Proceedings of 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS). IEEE, 2021: 1 – 8. DOI: 10.1109/RDAAPS48126.2021.9452005
[29] SHOKRI R, SHMATIKOV V. Privacy-preserving deep learning [C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 1310 – 1321. DOI: 10.1145/2810103.2813687
[30] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy [C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 308 – 318. DOI: 10.1145/2976749.2978318
[31] JAYARAMAN B, WANG L X, EVANS D, et al. Distributed learning without distress: privacy-preserving empirical risk minimization [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. ACM, 2018: 6346 – 6357. DOI: 10.5555/3327345.3327531
[32] XU G W, LI H W, LIU S, et al. VerifyNet: secure and verifiable federated learning [J]. IEEE transactions on information forensics and security, 2020, 15: 911 – 926. DOI: 10.1109/TIFS.2019.2929409
[33] JAYARAMAN B, EVANS D. Evaluating differentially private machine learning in practice [EB/OL]. [2022-10-28]. https://arxiv.org/abs/1902.08874
[34] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep models under the GAN: information leakage from collaborative deep learning [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 603 – 618. DOI: 10.1145/3133956.3134012

Biographies

ZHANG Weiting received his PhD degree in communication and information systems from Beijing Jiaotong University, China in 2021. From Nov. 2019 to Nov. 2020, he was a visiting PhD student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an associate professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, edge intelligence, and machine learning for wireless networks.

LIANG Haotian received his BS degree from Lanzhou University, China in 2022. He is currently working towards his master's degree in the School of Cyberspace Science and Technology, Beijing Institute of Technology, China. His research interests include machine learning security, Internet of Things security, and cloud security.

XU Yuhua is currently an undergraduate student in School of Computer Science and Technology, Beijing Institute of Technology, China. She is currently working at the research laboratory of advanced network and data security at the School of Cyberspace Science and Technology, Beijing Institute of Technology. Her research interests include applied cryptography and blockchain.

ZHANG Chuan (chuanz@bit.edu.cn) received his PhD degree in computer science from Beijing Institute of Technology, China in 2021. From Sept. 2019 to Sept. 2020, he worked as a visiting PhD student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an assistant professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology, China. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

RIS-Assisted Federated Learning in Multi-Cell Wireless Networks

WANG Yiji, WEN Dingzhu, MAO Yijie, SHI Yuanming
(ShanghaiTech University, Shanghai 201210, China)

DOI: 10.12142/ZTECOM.202301004
https://kns.cnki.net/kcms/detail/34.1294.TN.20230227.1850.002.html, published online February 28, 2023
Manuscript received: 2022-12-04

Abstract: Over-the-air computation (AirComp) based federated learning (FL) has been a promising technique for distilling artificial intelligence (AI) at the network edge. However, the performance of AirComp-based FL is decided by the device with the lowest channel gain due to the signal alignment property. More importantly, most existing work focuses on a single-cell scenario, where inter-cell interference is ignored. To overcome these shortages, a reconfigurable intelligent surface (RIS)-assisted AirComp-based FL system is proposed for multi-cell networks, where a RIS is used for enhancing the poor user signal caused by channel fading, especially for the device at the cell edge, and reducing inter-cell interference. The convergence of FL in the proposed system is first analyzed and the optimality gap for FL is derived. To minimize the optimality gap, we formulate a joint uplink and downlink optimization problem. The formulated problem is then divided into two separable nonconvex subproblems. Following the successive convex approximation (SCA) method, we first approximate the nonconvex term to a linear form, and then alternately optimize the beamforming vector and phase-shift matrix for each cell. Simulation results demonstrate the advantages of deploying a RIS in multi-cell networks and our proposed system significantly improves the performance of FL.

Keywords: federated learning (FL); reconfigurable intelligent surface (RIS); over-the-air computation (AirComp); multi-cell networks

Citation (IEEE Format): Y. J. Wang, D. Z. Wen, Y. J. Mao, et al., "RIS-assisted federated learning in multi-cell wireless networks," *ZTE Communications*, vol. 21, no. 1, pp. 25 – 37, Mar. 2022. doi: 10.12142/ZTECOM.202301004.

1 Introduction

With the development of the Internet of Things (IoT) and wireless technologies, recent years have witnessed an explosion of IoT devices and mobile data, which is of great significance for training AI models to enable various kinds of intelligent applications, such as auto-driving vehicles, equipment condition monitoring, and smart cities[1 – 2]. However, conventional methods that upload massive distributed data to a cloud encounter huge communication overhead and violate data privacy. To overcome these problems, federated learning (FL) emerges as a promising solution, where a shared AI model is trained among multiple devices without raw data transmission[3 – 6]. Specifically, there are three steps in each training iteration of FL. First, a central server generates an initial global model and then broadcasts the global model to the edge devices covered by it. Then, each edge device performs one or more steps of local training based on the received global model and local dataset to calculate a local model or gradient vector and uploads it to the central server. Finally, the central server aggregates all local information and updates the global model for the next communication round.

One main research direction of FL is to overcome the communication bottleneck caused by frequent transmission of the high dimensional model and gradient vectors. To combat the influence of wireless communications, the authors in Ref. [7] proposed a joint learning and communication framework to minimize the FL loss function. Partial device participation approaches, such as random scheduling and proportional fairness, have been proposed for the rational allocation of limited communication resources in FL[8]. To improve the communication efficiency of the FL uplink model aggregation, an over-the-air computation (AirComp) technique based on the waveform superposition characteristics of the multiple access channels (MACs) was proposed in Refs. [9 – 13], which realizes the summation calculation of the receiver function during information transmission. To overcome the bottleneck of limited communication bandwidth in the aggregation process, the authors in Ref. [14] presented a fast model aggregation method to improve the performance of FL by jointly optimizing beamforming vectors and device selection. In Ref. [15], a federated zeroth-order optimization (FedZO) algorithm based on AirComp was proposed to enable communication-efficient transmission by performing multiple local updates and partial device participation. Compared with the orthogonal multiple access (OMA) method, where the information of other users is regarded as interference, and the summation of all signals is

then calculated, i. e., computing after communication, AirComp greatly improves communication efficiency. The benefits of AirComp-based FL have motivated its application in the unmanned aerial vehicle (UAV)[16 - 17] and reconfigurable intelligent surface (RIS)-enabled networks[18 - 23].

The schemes mentioned above cannot solve the essential problem that wireless channel fading leads to poor signal strength of many devices, especially for AirComp-based FL, whose performance generally depends on the worst device in the network. To mitigate the effects of wireless channel fading, RIS is recognized as a revolutionary technology that achieves high spectrum and energy efficiency by reconfiguring the wireless channel environment at a low cost[24 - 27]. The authors in Ref. [25] designed a RIS-assisted AirComp system to increase the performance of AirComp by optimizing the transceivers and RIS phase-shift. It was shown in Refs. [19 - 20] that configuring RISs in AirComp-based FL further reduced the error of model aggregation, thereby improving the learning performance. Considering the low latency and privacy-secure nature of FL, a differentially private FL system via RIS was proposed in Ref. [12] to achieve a better tradeoff between the learning performance and privacy under the constraints of privacy and power. In order to further reduce the aggregation error, a multi-RIS scenario was presented in Ref. [28], where both the base station and the user used one dedicated RIS to mitigate the effects of poor channels. However, all the aforementioned works are limited to a single-cell setting. In fact, considering a multi-cell scenario is more in line with practical large-scale network design[29 - 31]. Due to the serious fading of the signal received by users at the cell edge, deploying RISs can relay the intended signal to enhance signal strengths for edge users and expand network coverage in multi-cell scenarios[31 - 33]. Besides, the authors in Refs. [30] and [34] proved that deploying a RIS at the cell edge can achieve the highest performance gain compared with other RIS deployments. Most of the existing RIS-assisted multi-cell networks focus on communication-only system models, ignoring the application of FL. Although the multi-cell FL interference management was considered in Ref. [29], RIS was not considered to enhance the performance of FL. To the best of our knowledge, this is the first work that investigates AirComp-based FL in RIS-assisted multi-cell networks.

In this paper, we investigate a RIS-assisted AirComp-based FL system in multi-cell networks, where a RIS is deployed at the cell edge to help each cell complete different FL tasks. In the process of FL, we consider both the impact of downlink and uplink communications. For the fast aggregation of uplink gradients, we adopt AirComp to improve communication efficiency. However, the performance of AirComp-based FL is dependent on the device with the worst link gain (e.g., the cell-edge device with a large path loss). Besides, the inter-cell interference also degrades its performance. To address these issues, we further deploy a RIS at the cell edge to enhance signal strength and mitigate inter-cell interference, thereby improving the FL performance. In our proposed system, there are some difficulties that we need to highlight. First, we consider both the impact of downlink model dissemination and that of uplink gradient aggregation, both are inevitably affected by channel fading, noise and inter-cell interference. It is different from most FL works, i. e., only uplink aggregation errors are considered. Second, considering the downlink influence makes the convergence analysis of our system more complicated. This derivation result is related to noise and inter-cell interference. Third, the optimization problems are non-convex and complex. We have to jointly optimize the beamforming vector and phase shift to improve the performance of our proposed system. The main contributions of this paper are summarized as follows:

• We propose a RIS-assisted AirComp-based FL system in two-cell networks, where a RIS is used for enhancing the signal of cell-edge devices during the process of both downlink and uplink transmission as well as for canceling the inter-cell interference. Then, we derive the convergence analysis of the proposed framework. The optimal gap of FL is determined by the uplink error and the downlink error of two cells, and each error contains channel fading, inter-cell interference and received noise.

• To maximize the learning performance for all cells, it is necessary to minimize the optimal gap. To this end, we decouple this optimization problem into two separate subproblems, respectively for the downlink and uplink optimization. Each subproblem requires a joint alternating optimization of beamforming vectors and phase-shift matrices. Since the optimization subproblems remain nonconvex, we first make a variable conversion and then utilize the successive convex approximation (SCA) method to approximate the problem. An alternative optimization algorithm is then proposed to solve each subproblem.

• Extensive simulations are performed to verify the performance of the proposed RIS-assisted FL system in two-cell networks. It shows that the proposed scheme can enhance the performance of the AirComp-based FL system by enhancing the signal strength and suppressing the inter-cell interference. In addition, the proposed algorithm guarantees fairness among cells.

The rest of this paper is organized as follows. Section 2 introduces the system model of RIS-assisted AirComp-based FL in a two-cell scenario. Section 3 provides the convergence analysis and the problem formulation. In Section 4, we propose an SCA-based joint alternating beamforming and phase-shift matrix optimization to minimize the upper bound of all cells. Simulation results are provided in Section 5 to support the advantages of the proposed system. Finally, we conclude this work in Section 6.

2 System Model

2.1 Network Model

As shown in Fig. 1, we mainly develop a RIS-assisted AirComp-based FL system in a two-cell network, where each cell has K single-antenna edge devices and one access point (AP), where each AP is equipped with N antennas. At the edge of two cells, we deploy a RIS to enhance the signal strength of edge devices, where the RIS has S passive reflecting elements. Edge device $k \in \mathcal{K}_l = \{1,2,\cdots,K\}$ is associated with AP $l \in \mathcal{L} = \{1,2\}$ to complete information exchange under both downlink and uplink communications, where $\mathcal{K}_l \cap \mathcal{K}_j = \varnothing$, $\forall l \neq j$ and $l,j \in \mathcal{L}$. During the process of transmission, we assume that each AP knows the channel state information for all edge devices.

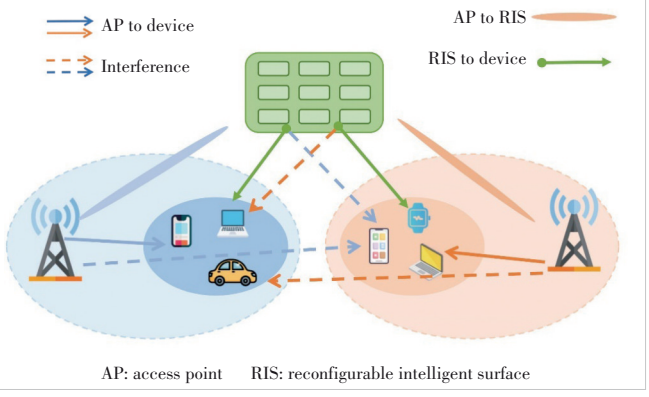
2.2 Federated Learning Model

In the two-cell FL system, each edge device $k \in \mathcal{K}_l$ has its own local dataset $\mathcal{D}_k = \left\{\left(x_i, y_i\right)\right\}_{i=1}^{D_k}$ with $D_k = \left|\mathcal{D}_k\right|$ data samples and each cell trains an individual FL model. The goal of FL is to collaboratively train a shared model $\boldsymbol{w}_l \in R^d$ of dimension d without making any local dataset public. The local loss function for edge device $k \in \mathcal{K}_l$ is defined as

$$F_{l,k}\left(\boldsymbol{w}_l\right) = \frac{1}{D_k} \sum_{\left(x_i,y_i\right) \in \mathcal{D}_k} f_l\left(\boldsymbol{w}_l; x_i, y_i\right), \tag{1}$$

where $f_l(\cdot)$ is the sample-wise loss function defined by the learning task for cell l. In this work, we consider a general model where learning tasks of the two cells are different. Without loss of generality, all local datasets for users in the same cell are assumed to have the same size, i. e., $\left|\mathcal{D}_{k_1}\right| = \left|\mathcal{D}_{k_2}\right|, \forall k_1, k_2 \in \mathcal{K}_l$. As a result, the global loss function for the learning task in cell l can be expressed as

$$F_l\left(\boldsymbol{w}_l\right) = \frac{1}{K} \sum\nolimits_{k \in \mathcal{K}_l} F_{l,k}\left(\boldsymbol{w}_l\right). \tag{2}$$



▲**Figure 1. RIS-assisted AirComp-based FL system in a two-cell network**

Then the global model for the l-th cell is obtained by

$$\boldsymbol{w}_l^* = \arg\min\nolimits_{\boldsymbol{w}_l \in R^d} F_l\left(\boldsymbol{w}_l\right). \tag{3}$$

To achieve the above FL purpose, we utilize the federated stochastic gradient descent (FedSGD) algorithm to perform local updates, which means only part of the datasets participates in training. Specifically, at the t-th communication round, AP l and the edge devices perform the following three procedures:

1) Broadcasting: AP l broadcasts the current global model $\boldsymbol{w}_l^t$ to the edge devices belonging to this cell l.

2) Local model update: Based on the received global model $\boldsymbol{w}_l^t$, each edge device $k \in \mathcal{K}_l$ performs a one-step local model update via the local mini-batch SGD algorithm, which is given by

$$\boldsymbol{w}_{l,k}^t = \boldsymbol{w}_l^t - \frac{\zeta^t}{B} \sum_{\left(\boldsymbol{x}_i,y_i\right) \in \mathcal{B}_k^t} \nabla f_l\left(\boldsymbol{w}_l^t; \boldsymbol{x}_i, y_i, \mathcal{B}_k^t\right) = \boldsymbol{w}_l^t - \zeta^t \nabla F_{l,k}\left(\boldsymbol{w}_l^t\right), \tag{4}$$

where ζ^t denotes the learning rate and $\mathcal{B}_k^t$ is the mini-batch dataset with size $\mathcal{B}_k^t$. Besides, we let $\boldsymbol{p}_k^t = \nabla F_{l,k}(\boldsymbol{w}_l^t)$ denote the trained gradient information. Then all edge devices upload the computed gradient information $\boldsymbol{p}_k^t$.

3) Model aggregation and update: The AP aggregates the received local gradient information and then generates a new global model as:

$$\boldsymbol{w}_l^{t+1} = \boldsymbol{w}_l^t - \frac{\zeta^t}{K_l} \sum_{k \in \mathcal{K}_l} \boldsymbol{p}_k^t. \tag{5}$$

Algorithm 1 summarizes the above steps of FedSGD.

Algorithm 1: FedSGD

Input: Initialize the global model $\boldsymbol{w}^0$, communication round T, local iteration epoch E, mini-batch dataset $\mathcal{B}$, and learning rate ζ.

for communication round $t = 1, 2, \cdots, T$ **do**

AP broadcasts the global model $\boldsymbol{w}^t$ to the edge devices;

Edge devices initial local model $\boldsymbol{w}_k^{t,0} = \boldsymbol{w}^t$ and make E local training;

for local iteration epoch $e = 1, 2, \cdots, E$ **do**

$\boldsymbol{w}_k^{t,e} \leftarrow \text{LocalSGD}(\boldsymbol{w}_k^{t,e-1}, \mathcal{B}_k^t)$

end

Compute the cumulative gradient information $\nabla f_{l,k} \leftarrow \boldsymbol{w}_k^{t,e} - \boldsymbol{w}_k^{t,0}$;

Upload the gradient information and update the global model

$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \zeta^t \nabla F_{l,k}(\boldsymbol{w}_l^t)$;

end

In the proposed two-cell system, we assume that these steps are synchronous in both cells and their gradient information is

uploaded to the AP. The synchronization can be enabled by AirShare[35], which transmits the clock over the air and provides a distributed protocol. In the next section, we elaborate on the communication process of the proposed system following the procedure of FL.

2.3 Downlink Communication for RIS-Assisted FL System

From the perspective of communication, we utilize the universal frequency reuse technique to improve spectral efficiency. In other words, the two cells share the same frequency during both downlink and uplink communications, inevitably causing inter-cell interference.

Considering a round of downlink communications in cell l, AP l shares the global model with each edge device in cell l. However, in most of the existing works on FL, the process of broadcast is error-free, which indicates the edge device $k \in \mathcal{K}_l$ can accurately receive signals from AP l. In this subsection, we consider the effects of noise and inter-cell interference in downlink communications. Here, we omit the time index and denote the downlink transmitted signal from AP l to the edge device k as w_l. In addition, we assume w_l follows the standard Gaussian distribution, i.e., $w_l \sim \mathcal{CN}(0,1)$. However, the transmitted signals may go through poor channel conditions in the communication process, which results in a larger receive error at edge device k. To lift the accuracy of the received signal, we deploy a RIS to mitigate the distortion of signals.

Specifically, we let $\boldsymbol{\theta}_d = \left[\beta e^{j\theta_1^{dl}}, \cdots, \beta e^{j\theta_S^{dl}}\right]$ represent the diagonal phase-shift matrix of the RIS in the downlink communication and $\boldsymbol{\Theta}_d = \mathrm{diag}(\boldsymbol{\theta}_d)$ with $\theta_s^{dl} \in [0, 2\pi]$ and $\beta \in [0, 1]$ is the amplitude reflection coefficient on the incident signal. To be specific, we set $\beta = 1$ in this paper and mainly consider the first reflected signal[24], because the signal reflected by multiple times appears insignificant due to propagation loss. Subsequently, we let $\boldsymbol{h}_{l,k}^l \in \mathbb{C}^N$, $\boldsymbol{h}_{l,k}^r \in \mathbb{C}^S$, and $\boldsymbol{G}_l \in \mathbb{C}^{N \times S}$ denote the equivalent channels from edge device k in cell l to AP l, from edge device k in cell l to the RIS, and from the RIS to AP l, respectively. We define the k-th edge device in $\mathcal{K}_l$ as the (l,k)-th edge device. And then, the received signal at edge device k in $\mathcal{K}_l$ from AP to the device and that from AP to RIS and to the device are given by

$$y_{l,k} = \left(\boldsymbol{h}_{l,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H + \boldsymbol{h}_{l,k}^{l,H}\right)\boldsymbol{t}_l w_l + \sum_{j \neq l}\left(\boldsymbol{h}_{j,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H + \boldsymbol{h}_{j,k}^{l,H}\right)\boldsymbol{t}_j w_j + n_k, \quad (6)$$

where $\boldsymbol{t}_l \in C^N$ denotes the transmit beamforming vector at AP l, and $n_k \sim \mathcal{CN}(0, \sigma_d^2)$ is the additive white Gaussian noise with zero mean and variance σ_d^2 at the (l,k)-th edge device. The transmit power constraint at AP l satisfies $E\left[|\boldsymbol{t}_l w_l|^2\right] = |\boldsymbol{t}_l|^2 \leqslant P_d$, where $P_d \geqslant 0$ denotes the maximum transmit power at AP l. Supposing perfect channel state information (CSI) is available, each edge device k in cell l can estimate the received global model by scaling a designed receive scalar $r_{l,k}$ which is set to $r_{l,k} = \left(\left(\boldsymbol{h}_{l,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H + \boldsymbol{h}_{l,k}^{l,H}\right)\boldsymbol{t}_l\right)^{-1}$. The received global model at edge device k is given by

$$w_{l,k} = r_{l,k}y_{l,k} = w_l + e_k^{dl}, \quad (7)$$

where $e_k^{\mathrm{dl}} = \left(\sum_{j \neq l}\left(\boldsymbol{h}_{j,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H + \boldsymbol{h}_{j,k}^{l,H}\right)\boldsymbol{t}_j w_j + n_k\right) \Big/ \left(\left(\boldsymbol{h}_{l,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H + \boldsymbol{h}_{l,k}^{l,H}\right)\boldsymbol{t}_l\right)$ consists of the inter-cell interference and noise. Repeating d times, the global model is

$$\boldsymbol{w}_{l,k} = \boldsymbol{w}_l + \mathrm{Re}\{\boldsymbol{e}_k^{\mathrm{dl}}\}, \quad (8)$$

where $\boldsymbol{w}_{l,k}$, $\boldsymbol{w}_l$ and $\boldsymbol{e}_k^{\mathrm{dl}}$ are all vectors of dimension d. After receiving the global model $\boldsymbol{w}_{l,k}$, all edge devices start training based on the local data and then generate new local model parameters. The gradient information is the difference between the global model and the local model as in Eq. (4). After that, all edge devices upload their gradient information to AP l through the uplink communication.

2.4 Uplink AirComp Aggregation for RIS-Assisted FL System

In uplink communications, since the average sum in Eq. (5) for gradient aggregation is included in the category of nomographic functions, AirComp, as a promising technique, takes advantage of the waveform superposition properties of MACs in wireless networks to improve transmission efficiency. Fig. 2 shows the process of AirComp. For the sake of briefness, we also omit the time index in the following presentation. The transmitted signal and pre-processing function of the (l,k)-th edge device are denoted by $x_{l,k} \in C$ and $\psi_{l,k}(\cdot): C \to C$, respectively. The target function processed at the l-th AP is given by

$$f = \phi\left(\sum_{k \in \mathcal{K}_l}\psi_{l,k}\left(x_{l,k}\right)\right), \quad (9)$$

where $\phi(\cdot)$ is the post-processing function at the AP. After pre-processing, the symbol transmitted at the (l,k)-th edge device $s_{l,k}$ is assumed to be independent and has the nature of zero mean and unit variance, i.e., $E\left[s_{l,k}\right] = 0$, $E\left[s_{l,k}s_{l,k}^H\right] = 1$. In this case, the aggregation at the l-th AP is expressed as

$$g_l = \sum_{k \in \mathcal{K}_l} s_{l,k}. \quad (10)$$

Similar to the downlink communication, we let $\boldsymbol{\theta}_u = \left[\beta e^{j\theta_1^{ul}}, \cdots, \beta e^{j\theta_S^{ul}}\right]$ represent the diagonal phase-shift matrix of the RIS in the uplink communication and $\boldsymbol{\Theta}_u = \mathrm{diag}(\boldsymbol{\theta}_u)$ with $\theta_s^{ul} \in [0, 2\pi]$. AP l mainly aggregates three types of signals, namely, the signal of cell l, the interference signal of other cells, and noise, where the first two items both contain the signal from the edge devices to AP l and the signal from the edge devices to

RIS and to AP l. Thus, the received signal at AP l is given by

$$\boldsymbol{y}_l = \sum_{k \in \mathcal{K}_l} \left(\boldsymbol{G}_l \boldsymbol{\Theta}_u \boldsymbol{h}_{l,k}^r + \boldsymbol{h}_{l,k}^l \right) z_{l,k} s_{l,k} + \sum_{i \in \mathcal{K}_j, j \neq l} \left(\boldsymbol{G}_j \boldsymbol{\Theta}_u \boldsymbol{h}_{j,i}^r + \boldsymbol{h}_{j,i}^l \right) z_{j,i} s_{j,i} + \boldsymbol{n}_l, \tag{11}$$

where $z_{l,k} \in C$ is the transmit scalar at the (l,k)-th edge device and it satisfies the maximum power constraint $\left| z_{l,k} \right|^2 \leqslant P$, and $\boldsymbol{n}_l \in C^N \sim \mathcal{CN}(0, \sigma^2 I)$ denotes the additive white Gaussian noise with zero mean and variance σ^2. Then, the estimated function at AP l after post-processing is marked as

$$\widehat{g}_l = \frac{1}{\sqrt{\eta_l}} \boldsymbol{m}_l^H \boldsymbol{y}_l = \frac{1}{\sqrt{\eta_l}} \boldsymbol{m}_l^H \sum_{k \in \mathcal{K}_l} \left(\boldsymbol{G}_l \boldsymbol{\Theta}_u \boldsymbol{h}_{l,k}^r + \boldsymbol{h}_{l,k}^l \right) z_{l,k} s_{l,k} + \frac{1}{\sqrt{\eta_l}} \boldsymbol{m}_l^H \sum_{i \in \mathcal{K}_j, j \neq l} \left(\boldsymbol{G}_j \boldsymbol{\Theta}_u \boldsymbol{h}_{j,i}^r + \boldsymbol{h}_{j,i}^l \right) z_{j,i} s_{j,i} + \frac{\boldsymbol{m}_l^H \boldsymbol{n}_l}{\sqrt{\eta_l}}, \tag{12}$$

where $\boldsymbol{m}_l$ denotes the received beamforming vector at AP l and η_l denotes the denoising factor to suppress noise. Following Ref. [34], each transmit scalar can be designed as

$$z_{l,k} = \sqrt{\eta_l} \left(\boldsymbol{m}_l^H \left(\boldsymbol{G}_l \boldsymbol{\Theta}_u \boldsymbol{h}_{l,k}^r + \boldsymbol{h}_{l,k}^l \right) \right)^{-1}. \tag{13}$$

Therefore, the estimated function at AP l can further be expressed as

$$\widehat{g}_l = g_l + e_l^{\mathrm{ul}}, \tag{14}$$

where $e_l^{\mathrm{ul}} = \left(\boldsymbol{m}_l^H / \sqrt{\eta_l} \right) \sum_{i \in \mathcal{K}_j, j \neq l} \left(\boldsymbol{G}_j \boldsymbol{\Theta}_u \boldsymbol{h}_{j,i}^r + \boldsymbol{h}_{j,i}^l \right) z_{j,i} s_{j,i} + \left(\boldsymbol{m}_l^H \boldsymbol{n}_l / \sqrt{\eta_l} \right)$ denotes the total uplink error, which includes the inter-cell interference and noise. When AP l completes the aggregation process, a new round of global model updates is generated according to Eq. (5), i.e., $\boldsymbol{w}_l^{t+1} = \boldsymbol{w}_l^t - \frac{\zeta^t}{K_l} \widehat{g}_l$.

3 Convergence Analysis and Problem Formulation

In this section, we provide the convergence analysis of the proposed RIS-Assisted AirComp-based two-cell FL system. Based on the convergence results, we get an optimality gap bound that is influenced by both the downlink and uplink errors. In addition, we formulate the optimization problem to improve the performance of the proposed system.

3.1 Convergence Results

Assumption 1: M-Smoothness. All local loss functions $(F_1, \cdots, F_k)$ are M-Smoothness. For all $\boldsymbol{x}$ and $\boldsymbol{y}$, we have

$$F_k(\boldsymbol{x}) \leqslant F_k(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^T \nabla F_k(\boldsymbol{x}) + \frac{M}{2} \left|\left| \boldsymbol{x} - \boldsymbol{y} \right|\right|^2. \tag{15}$$

Assumption 2: μ-strongly convexity. All local loss functions $F_1, \cdots, F_k$ are μ-strongly convex. For all $\boldsymbol{x}$ and $\boldsymbol{y}$, we have

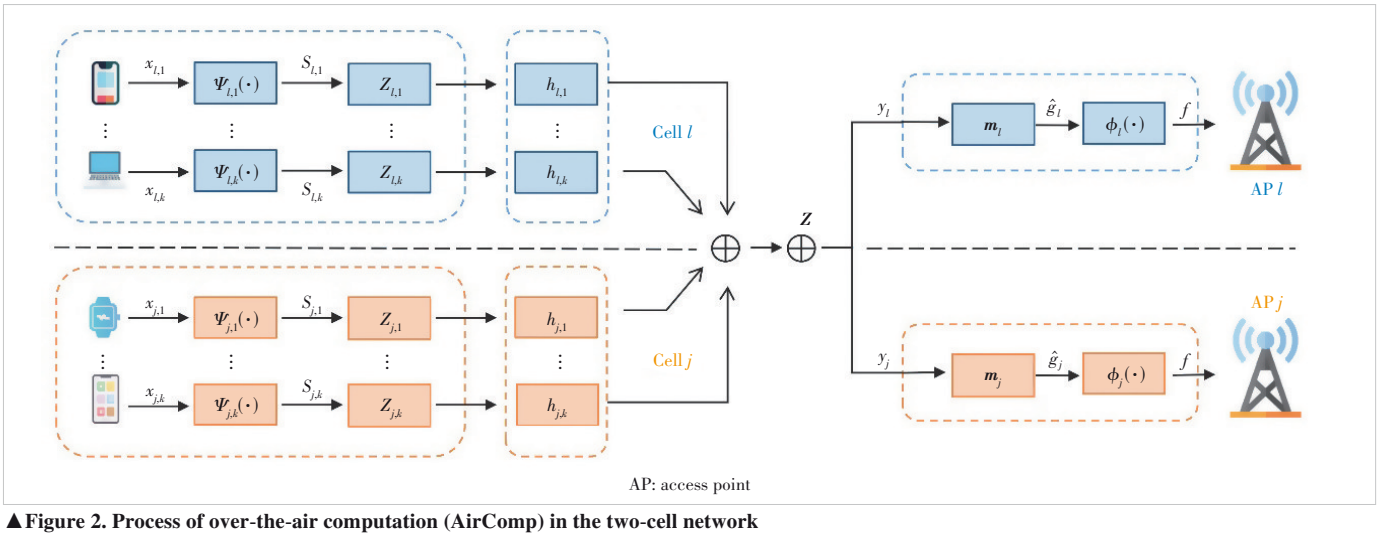
$$F_k(\boldsymbol{x}) \geqslant F_k(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^T \nabla F_k(\boldsymbol{x}) + \frac{\mu}{2} \left|\left| \boldsymbol{x} - \boldsymbol{y} \right|\right|^2. \tag{16}$$

Theorem 1: Let Assumptions 1 and 2 be hold. In cell l, the learning rate satisfies $0 \leqslant \zeta_t \equiv \zeta = 1/M$. After T communication rounds, the expected optimality gap in the RIS-Assisted FL system is upper bounded by

$$\mathbb{E}\left[F_l\left(\boldsymbol{w}_l^T \right) - F_l\left(\boldsymbol{w}_l^\star \right) \right] \leqslant \rho^T \mathbb{E}\left[F_l\left(\boldsymbol{w}_l^0 \right) - F_l\left(\boldsymbol{w}_l^* \right) \right] + \sum_{t=0}^{T-1} \rho^{T-t-1} \left(\frac{M}{2K} \sum_{k \in \mathcal{K}} \mathbb{E}\left[\left|\left| \boldsymbol{e}_{k,t}^{\mathrm{dl}} \right|\right|^2 \right] + \frac{1}{2MK^2} \mathbb{E}\left[\left|\left| \boldsymbol{e}_{l,t}^{\mathrm{ul}} \right|\right|^2 \right] \right), \tag{17}$$

where $\rho = 1 - \mu/M$.

Proof: Please refer to Appendix for details.



▲Figure 2. Process of over-the-air computation (AirComp) in the two-cell network

3.2 Problem Formulation

According to Theorem 1, the first term to the right of the inequality gradually tends to zero as the number of T increases. Thus, the upper bound is dominated by the last term, which includes the inter-cell interference and noise error in the downlink and uplink communications. we aim to minimize the upper bound in each time slot for transmitting the gradient information in all cells, given by

$$\sum_{l=1}^{L}\left(\frac{M}{2K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left|\left|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right|\right|^2\right]+\frac{1}{2MK^2}\mathbb{E}\left[\left|\left|\boldsymbol{e}_{l,t}^{\mathrm{ul}}\right|\right|^2\right]\right),\forall t\in T,\forall l\in L, \tag{18}$$

We denote the optimization objective in Eq. (18) by the symbol $\mathcal{E}$. Specially, the denoising factor η_l in Eq. (14) is designed as

$$\eta_l = P\min_k\left\|\boldsymbol{m}_l^H\left(\boldsymbol{G}_l\boldsymbol{\Theta}_u\boldsymbol{h}_{l,k}^r+\boldsymbol{h}_{l,k}^l\right)\right\|^2. \tag{19}$$

Then, the corresponding optimization problem can be formulated as

$$\begin{aligned}&\underset{\boldsymbol{m}_l,\boldsymbol{\Theta}_u,\boldsymbol{t}_l,\boldsymbol{\Theta}_d}{\text{minimize}}\ \mathcal{E}\\&\text{subject to}\left|\theta_s^{\mathrm{ul}}\right|=1,\forall s=1,\cdots,S,\\&\qquad\left|\theta_s^{\mathrm{dl}}\right|=1,\forall s=1,\cdots,S,\\&\qquad\left\|\boldsymbol{t}_l\right\|^2\leqslant P_d,\end{aligned} \tag{20}$$

where θ_s^{ul} and θ_s^{dl} mean the phase-shift constraints, and $\boldsymbol{t}_l$ is the transmit beamforming constraint.

$$\mathbb{E}\left[\left|\left|\boldsymbol{e}_{l,t}^{\mathrm{dl}}\right|\right|^2\right]=\frac{\sum_{j\neq l}\left\|\left(\boldsymbol{h}_{j,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H+\boldsymbol{h}_{j,k}^{l,H}\right)\boldsymbol{t}_j\right\|^2+\sigma_d^2}{\left\|\left(\boldsymbol{h}_{l,k}^{rH}\boldsymbol{\Theta}_d\boldsymbol{G}_l^H+\boldsymbol{h}_{l,k}^{l,H}\right)\boldsymbol{t}_l\right\|^2}, \tag{21}$$

$$\mathbb{E}\left[\left|\left|\boldsymbol{e}_{l,t}^{\mathrm{ul}}\right|\right|^2\right]=\sum_{\substack{i\in\mathcal{K}_j\\j\neq l}}\frac{\eta_j\left\|\boldsymbol{m}_l^H\left(\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{j,i}^r+\boldsymbol{h}_{j,i}^l\right)\right\|^2}{\eta_l\left\|\boldsymbol{m}_j^H\left(\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{j,i}^r+\boldsymbol{h}_{j,i}^j\right)\right\|^2}+\frac{\left\|\boldsymbol{m}_l^H\right\|^2\sigma^2}{\eta_l}. \tag{22}$$

For Problem (20), the optimization variables are the received beamforming vector $\boldsymbol{m}$, uplink phase-shift matrix $\boldsymbol{\Theta}_u$, transmit beamforming vector $\boldsymbol{t}$, and downlink phase-shift matrix $\boldsymbol{\Theta}_d$. The first two correspond to variables in the uplink process, and the last two are variables in the downlink process. We observe that the variables in these two processes are not coupled and their corresponding constraints are independent. Therefore, we can decompose the optimization objective into two sub-problems, i. e., downlink and uplink optimizations. Then, we can further solve Problem (20) by minimizing the following two sub-problems in Eqs. (23) and (24) simultaneously.

$$\begin{aligned}&\underset{\boldsymbol{m}_l,\boldsymbol{\Theta}_u}{\text{minimize}}\ \sum_{l=1}^{L}\mathbb{E}\left[\left|\left|\boldsymbol{e}_{l,t}^{\mathrm{ul}}\right|\right|^2\right],\\&\text{subject to}\ \left|\theta_s^{\mathrm{ul}}\right|=1,\forall s=1,\cdots,S.\end{aligned} \tag{23}$$

$$\begin{aligned}&\underset{\boldsymbol{t}_l,\boldsymbol{\Theta}_d}{\text{minimize}}\ \sum_{l=1}^{L}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left|\left|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right|\right|^2\right],\\&\text{subject to}\ \left|\theta_s^{\mathrm{dl}}\right|=1,\forall s=1,\cdots,S,\\&\qquad\left\|\boldsymbol{t}_l\right\|^2\leqslant P_d.\end{aligned} \tag{24}$$

4 Optimization Framework

In this section, we specify the optimization framework for solving the uplink and downlink optimization problems, respectively. Each optimization problem also includes both beamforming optimization and phase-shift optimization.

4.1 Uplink Optimization

To simplify Eq. (19), we introduce an auxiliary variable vector $\boldsymbol{\gamma}_l=\min_k\|\boldsymbol{m}_l^H\left(\boldsymbol{G}_l\boldsymbol{\Theta}_u\boldsymbol{h}_{l,k}^r+\boldsymbol{h}_{l,k}^l\right)\|^2$ for cell l. By taking Eq. (19) to Problem (23) and introducing a new optimizing variable $\boldsymbol{v}_l=\boldsymbol{m}_l/\sqrt{\gamma_l}$, the minimum problem in Eq. (23) can be adapted as

$$\begin{aligned}&\underset{\boldsymbol{v}_l,\boldsymbol{\Theta}_u}{\text{minimize}}\ \sum_{l=1}^{L}\sum_{\substack{i\in\mathcal{K}_j\\j\neq l}}\frac{\left\|\boldsymbol{v}_l^H\left(\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{j,i}^r+\boldsymbol{h}_{j,i}^l\right)\right\|^2}{\left\|\boldsymbol{v}_j^H\left(\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{j,i}^r+\boldsymbol{h}_{j,i}^j\right)\right\|^2}+\sum_{l=1}^{L}q\left\|\boldsymbol{v}_l\right\|^2,\\&\text{subject to}\ \left\|\boldsymbol{v}_l^H\left(\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{l,k}^r+\boldsymbol{h}_{l,k}^l\right)\right\|^2\geqslant 1,\forall k\in\mathcal{K}_l,\forall l,\\&\qquad\left|\theta_s^{\mathrm{ul}}\right|=1,\forall s=1,\cdots,S.\end{aligned} \tag{25}$$

where $q=\sigma^2/P$ is a constant. We observe that the above problem turns out to be highly intractable due to the nonconvexity of the objective function and nonconvex quadratic constraints for $\boldsymbol{v}$ and $\boldsymbol{\Theta}$. First, we decompose the above optimization problem into $L+1$ subproblems, i.e., L beamforming problems and a phase-shift problem. Then, we propose an alternative optimization algorithm to solve the uplink optimization problem.

1) Received beamforming optimization: We fix the diagonal phase-shift matrix $\boldsymbol{\Theta}_u$, and the l-th optimization sub-problem can be written as

$$\begin{aligned}&\underset{\boldsymbol{v}_l}{\text{minimize}}\ \sum_{\substack{i\in\mathcal{K}_j\\j\neq l}}\frac{\left\|\boldsymbol{v}_l^H\boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u}\right\|^2}{\left\|\boldsymbol{v}_j^H\boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u}\right\|^2}+q\left\|\boldsymbol{v}_l\right\|^2,\\&\text{subject to}\ \left\|\boldsymbol{v}_l^H\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u}\right\|^2\geqslant 1,\forall k\in\mathcal{K}_l,\forall l,\end{aligned} \tag{26}$$

where $\boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u}=\boldsymbol{G}_j\boldsymbol{\Theta}_u\boldsymbol{h}_{j,i}^r+\boldsymbol{h}_{j,i}^l$, and $\boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u}$ and $\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u}$ are also followed by this representation. Then we introduce an auxiliary

variable $b_{j,i}$ which satisfies $\left\| \boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right\|^2 \Big/ \left\| \boldsymbol{v}_j^H \boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u} \right\|^2 \leqslant b_{j,i}$. Subsequently, Problem (26) can be equivalent to

$$\begin{aligned} \underset{\boldsymbol{v}_l, \boldsymbol{b}}{\text{minimize}} \quad & \sum_{\substack{i \in \mathcal{K}_j \\ j \neq l}} b_{j,i} + q \left\| \boldsymbol{v}_l \right\|^2, \\ \text{subject to} \quad & \left\| \boldsymbol{v}_l^H \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u} \right\|^2 \geqslant 1, \forall k \in \mathcal{K}_l, \forall l, \\ & \frac{\left\| \boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right\|^2}{\left\| \boldsymbol{v}_j^H \boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u} \right\|^2} \leqslant b_{j,i}, \forall j \neq l. \end{aligned} \tag{27}$$

However, the constraints in Eq. (27) are nonconvex for the optimization variable $\boldsymbol{v}_l$. To address the nonconvexity of the constraints, we use the SCA method to transform the quadratic form into a linear constraint[11]. We let $\boldsymbol{a}_{l,k} = \left[\mathrm{Re}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u} \right), \mathrm{Im}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u} \right) \right]$ and $\boldsymbol{a}_{j,i} = \left[\mathrm{Re}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right), \mathrm{Im}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right) \right]$, and the corresponding linear constraints are

$$\begin{aligned} & \left\| \boldsymbol{a}_{l,k}^{(t)} \right\|^2 + 2(\boldsymbol{a}_{l,k}^{(t)})^T (\boldsymbol{a}_{l,k} - \boldsymbol{a}_{l,k}^{(t)}) \geqslant 1, \forall k, \forall l, \\ & \left\| \boldsymbol{a}_{j,i}^{(t)} \right\|^2 + 2(\boldsymbol{a}_{j,i}^{(t)})^T (\boldsymbol{a}_{j,i} - \boldsymbol{a}_{j,i}^{(t)}) \leqslant b_{j,i} \left\| \boldsymbol{v}_j^H \boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u} \right\|^2, \forall i, \forall j, \end{aligned} \tag{28}$$

where $\boldsymbol{a}_{l,k}^{(t)}$ and $\boldsymbol{a}_{j,i}^{(t)}$ are the t-th iteration solution. At the beginning of the iteration, the initial $\boldsymbol{a}_{l,k}^{(0)}$ and $\boldsymbol{a}_{j,i}^{(0)}$ can be randomly generated. By substituting Eq. (28) into Eq. (27), we have the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{v}_l, \boldsymbol{b}, \boldsymbol{a}}{\text{minimize}} \quad & \sum_{\substack{i \in \mathcal{K}_j \\ j \neq l}} b_{j,i} + q \left\| \boldsymbol{v}_l \right\|^2, \\ \text{subject to} \quad & \left\| \boldsymbol{a}_{l,k}^{(t)} \right\|^2 + 2(\boldsymbol{a}_{l,k}^{(t)})^T (\boldsymbol{a}_{l,k} - \boldsymbol{a}_{l,k}^{(t)}) \geqslant 1, \forall k, \forall l, \\ & \left\| \boldsymbol{a}_{j,i}^{(t)} \right\|^2 + 2(\boldsymbol{a}_{j,i}^{(t)})^T (\boldsymbol{a}_{j,i} - \boldsymbol{a}_{j,i}^{(t)}) \leqslant B_{j,i}, \forall i, \forall j, \\ & \boldsymbol{a}_{l,k} = \left[\mathrm{Re}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u} \right), \mathrm{Im}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_u} \right) \right], \\ & \boldsymbol{a}_{j,i} = \left[\mathrm{Re}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right), \mathrm{Im}\left(\boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{l,\boldsymbol{\Theta}_u} \right) \right], \end{aligned} \tag{29}$$

where $B_{j,i} = b_{j,i} |\boldsymbol{v}_l^H \boldsymbol{h}_{j,i}^{j,\boldsymbol{\Theta}_u}|^2$. Then, we find that the objective function and constraints are convex for any optimization variable, which means we can adopt the CVX tools to obtain the optimal beamforming vector $\boldsymbol{v}_l$. When the beamforming vectors of all cells are obtained, we start optimizing the phase shift.

2) Uplink phase-shift optimization: With the given beamforming vector $\boldsymbol{v}$, we transform the channel as $\boldsymbol{G}_j \boldsymbol{\Theta}_u \boldsymbol{h}_{j,i}^r = \boldsymbol{R}_{j,i}^r \boldsymbol{\theta}_u$, where $\boldsymbol{\theta}_u = \mathrm{diag}(\boldsymbol{\Theta}_u)$ and $\boldsymbol{R}_{j,i}^r \in \mathbb{C}^{N \times S}$ denotes the channel without phase-shift from node i to AP j. Then, the phase-shift optimization problem is rewritten as

$$\begin{aligned} \underset{\boldsymbol{\theta}_u}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{\substack{i \in \mathcal{K}_j \\ j \neq l}} \frac{\left\| \boldsymbol{v}_l^H \left(\boldsymbol{R}_{j,i}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{j,i}^l \right) \right\|^2}{\left\| \boldsymbol{v}_j^H \left(\boldsymbol{R}_{j,i}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{j,i}^j \right) \right\|^2}, \\ \text{subject to} \quad & \left\| \boldsymbol{v}_l^H \left(\boldsymbol{R}_{l,k}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{l,k}^l \right) \right\|^2 \geqslant 1, \forall k \in \mathcal{K}_l, \forall l, \\ & \left| \theta_s^{\mathrm{ul}} \right| = 1, \forall s = 1, \cdots, S. \end{aligned} \tag{30}$$

Unlike Problem (26), the optimization variable of the objective function in Problem (30) appears in both the numerator and denominator, which requires that we have to optimize the phase shift of all cells at the same time. For the equation constraints in Problem (30), we can reduce it to a convex constraint, i.e., $\left| \theta_s^{\mathrm{ul}} \right| \leqslant 1$. In addition, we let $\boldsymbol{x}_{lji} = \boldsymbol{v}_l^H (\boldsymbol{R}_{j,i}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{j,i}^l)$, $\boldsymbol{x}_{jji} = \boldsymbol{v}_j^H (\boldsymbol{R}_{j,i}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{j,i}^l)$, $\boldsymbol{x}_{llk} = \boldsymbol{v}_l^H (\boldsymbol{R}_{l,k}^r \boldsymbol{\theta}_u + \boldsymbol{h}_{l,k}^l)$ and $\left\| \boldsymbol{x}_{lji} \right\|^2 \Big/ \left\| \boldsymbol{x}_{jji} \right\|^2 \leqslant r_{j,i}$. After applying the SCA method, the corresponding phase-shift problem is expressed as:

$$\begin{aligned} \underset{\boldsymbol{\theta}_u, \boldsymbol{r}, \boldsymbol{y}}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{\substack{i \in \mathcal{K}_j \\ j \neq l}} r_{j,i}, \\ \text{subject to} \quad & \left\| \boldsymbol{y}_{llk}^{(t)} \right\|^2 + 2(\boldsymbol{y}_{llk}^{(t)})^T (\boldsymbol{y}_{llk} - \boldsymbol{y}_{llk}^{(t)}) \geqslant 1, \forall k, \forall l, \\ & \frac{\left\| \boldsymbol{y}_{lji}^{(t)} \right\|^2 + 2(\boldsymbol{y}_{lji}^{(t)})^T (\boldsymbol{y}_{lji} - \boldsymbol{y}_{lji}^{(t)})}{r_{j,i}} \leqslant \\ & \left\| \boldsymbol{y}_{jji}^{(t)} \right\|^2 + 2(\boldsymbol{y}_{jji}^{(t)})^T (\boldsymbol{y}_{jji} - \boldsymbol{y}_{jji}^{(t)}), \forall i, \forall j, \\ & \boldsymbol{y}_{lji} = \left[\mathrm{Re}\left(\boldsymbol{x}_{lji} \right), \mathrm{Im}\left(\boldsymbol{x}_{lji} \right) \right], \\ & \boldsymbol{y}_{jji} = \left[\mathrm{Re}\left(\boldsymbol{x}_{jji} \right), \mathrm{Im}\left(\boldsymbol{x}_{jji} \right) \right], \\ & \boldsymbol{y}_{llk} = \left[\mathrm{Re}\left(\boldsymbol{x}_{llk} \right), \mathrm{Im}\left(\boldsymbol{x}_{llk} \right) \right], \\ & \left| \theta_s^{\mathrm{ul}} \right| \leqslant 1, \forall s, \end{aligned} \tag{31}$$

where $\boldsymbol{y}_{llk}^{(t)}$, $\boldsymbol{y}_{lji}^{(t)}$ and $\boldsymbol{y}_{jji}^{(t)}$ are the t-th iteration solution. For Problem (31), the objective function and all constraints are convex, which indicates the optimal solution can be obtained from a convex program. Since we have scaled down the phase-shift equation constraints, when we get the optimal phase-shift solution from the convex program, we need to normalize it to satisfy the equation constraint.

The framework of optimization is summarized in Algorithm 2, where the process of solving Problems (29) and (31) is based on the SCA algorithm. For the equation constraint, we first relax it to obtain the optimal solution and then normalize the solution to satisfy the original condition.

Algorithm 2: Alternative beamforming and phase-shift algorithm

Input: The number of cells L, initial beamforming vector of each cell $\boldsymbol{v}_l$, $\in L$, initial random phase-shift matrix $\boldsymbol{\Theta}_u$, and

constant q.

Alternative beamforming optimization:

for the number of cell $l = 1, 2, \cdots, L$ **do**

Fixing $\boldsymbol{\Theta}_u$ and other cell $\boldsymbol{v}_j, j \neq l$, introducing the auxiliary variable $b_{j,i}$;

$\boldsymbol{v}_l \leftarrow$ solve Problem (29) by $(\boldsymbol{v}_l, \boldsymbol{v}_j, \boldsymbol{\Theta}_u, b_{j,i})$;

end

Phase-shift algorithm:

Fixing the beamforming vector $\boldsymbol{v}_l, \forall l$, introducing the auxiliary variable $z_{j,i}$;

Relaxing the equation constraint of Problem (30), i. e., $\left|\theta_s^{\mathrm{ul}}\right| = 1$.

$\boldsymbol{\Theta}_u \leftarrow$ solve Problem (31) by $(\boldsymbol{v}_l, \boldsymbol{v}_j, \boldsymbol{\Theta}_u, z_{j,i})$;

$\boldsymbol{\Theta}_u \leftarrow$ normalize $\boldsymbol{\Theta}_u$, i.e., $\left|\theta_s^{\mathrm{ul}}\right| = \left|\theta_s^{\mathrm{ul}}\right| / abs(\theta_s^{\mathrm{ul}})$.

Output: $\{\boldsymbol{v}_l, \forall l \in L, \boldsymbol{\Theta}_u\}$.

4.2 Downlink Optimization

The downlink optimization problem is

$$\begin{aligned} \underset{\boldsymbol{t}_l, \boldsymbol{\Theta}_d}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{k \in \mathcal{K}} \frac{\sum_{j \neq l} \left\| \left(\boldsymbol{\theta}_d^H \boldsymbol{T}_{j,k}^{r,H} + \boldsymbol{h}_{j,k}^{l,H} \right) \boldsymbol{t}_j \right\|^2 + \sigma_d^2}{\left\| \left(\boldsymbol{h}_{l,k}^{rH} \boldsymbol{\Theta}_d \boldsymbol{G}_l^H + \boldsymbol{h}_{l,k}^{l,H} \right) \boldsymbol{t}_l \right\|^2}, \\ \text{subject to} \quad & \left\| \boldsymbol{t}_l \right\|^2 \leqslant P_d, \forall l, \\ & \left| \theta_s^{\mathrm{dl}} \right| = 1, \forall s = 1, \cdots, S. \end{aligned} \tag{32}$$

We observe that Problem (32) is nonconvex for any optimization variable, and we cannot directly solve this optimization problem. For simplicity, we first let $\boldsymbol{h}_{j,k}^{l,\boldsymbol{\Theta}_d} = \boldsymbol{h}_{j,k}^{rH} \boldsymbol{\Theta}_d \boldsymbol{G}_l^H + \boldsymbol{h}_{j,k}^{l,H}$ and $\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} = \boldsymbol{h}_{l,k}^{rH} \boldsymbol{\Theta}_d \boldsymbol{G}_l^H + \boldsymbol{h}_{l,k}^{l,H}$. Then we divide this optimization problem into two parts (transmit beamforming and downlink phase-shift optimizations).

1) Transmit beamforming optimization: For a given diagonal phase-shift matrix $\boldsymbol{\Theta}_d$, we mainly focus on the downlink received beamforming optimization. Then, we introduce an auxiliary variable $\Delta_{l,k}$ which satisfies $\left(\sum_{j \neq l} |\boldsymbol{h}_{j,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_j|^2 + \sigma_d^2 \right) \Big/ |\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l|^2 \leqslant \Delta_{l,k}$. The optimization Problem (32) can now be converted to

$$\begin{aligned} \underset{\{\boldsymbol{t}, \Delta\}}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{k \in \mathcal{K}} \Delta_{l,k}, \\ \text{subject to} \quad & \left\| \boldsymbol{t}_l \right\|^2 \leqslant P_d, \forall l, \\ & \frac{\sum_{j \neq l} \cdot \left\| \boldsymbol{h}_{j,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_j \right\|^2 + \sigma_d^2}{\left\| \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l \right\|^2} \leqslant \Delta_{l,k}, \forall k \in \mathcal{K}_l, \forall l \end{aligned}. \tag{33}$$

For Constraint (33), we can adjust the inequality to $\left(d_{l,k} / \Delta_{l,k} \right) \leqslant \left\| \boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l \right\|^2$, where $d_{l,k} = \sum_{j \neq l} \cdot \left\| \boldsymbol{h}_{j,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_j \right\|^2 + \sigma_d^2$. In this case, for the nonconvex quadratic constraints concerning the variable $\boldsymbol{t}_l$, we can exploit the SCA algorithm to linearly approximate the constraint as

$$\left\| \boldsymbol{c}_{l,k}^{(t)} \right\|^2 + 2 (\boldsymbol{c}_{l,k}^{(t)})^T (\boldsymbol{c}_{l,k} - \boldsymbol{c}_{l,k}^{(t)}) \geqslant \frac{d_{l,k}}{\Delta_{l,k}}, \forall k, \forall l, \tag{34}$$

where $\boldsymbol{c}_{l,k} = [\mathrm{Re}(\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l), \mathrm{Im}(\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l)], \forall k, \forall l$, and $\boldsymbol{c}_{l,k}^{(t)}$ is the optimized solution after the t-th iterative optimization. Then, the optimization problem at the l-th iteration is

$$\begin{aligned} \underset{\boldsymbol{t}_l, \Delta}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{k \in \mathcal{K}} \Delta_{l,k}, \\ \text{subject to} \quad & \left\| \boldsymbol{t}_l \right\|^2 \leqslant P_d, \forall l, \\ & \left\| \boldsymbol{c}_{l,k}^{(t)} \right\|^2 + 2 (\boldsymbol{c}_{l,k}^{(t)})^T (\boldsymbol{c}_{l,k} - \boldsymbol{c}_{l,k}^{(t)}) \geqslant \frac{d_{l,k}}{\Delta_{l,k}}, \forall k, \forall l, \\ & \boldsymbol{c}_{l,k} = \left[\mathrm{Re}\left(\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l \right), \mathrm{Im}\left(\boldsymbol{h}_{l,k}^{l,\boldsymbol{\Theta}_d} \boldsymbol{t}_l \right) \right], \forall k, \forall l \end{aligned}. \tag{35}$$

Problem (35) is convex and we can easily solve it by utilizing convex optimization tools.

2) Downlink phase-shift optimization: We fix the transmit beamforming vector $\boldsymbol{t}$ and denote the channel as $\boldsymbol{G}_l \boldsymbol{\Theta}_d \boldsymbol{h}_{j,k}^r = \boldsymbol{T}_{j,k}^r \boldsymbol{\theta}_d$, $\boldsymbol{G}_l \boldsymbol{\Theta}_d \boldsymbol{h}_{l,k}^r = \boldsymbol{T}_{l,k}^r \boldsymbol{\theta}_d$, where $\boldsymbol{\theta}_d = \mathrm{diag}(\boldsymbol{\Theta}_d)$, $\boldsymbol{T}_{j,k}^r$ and $\boldsymbol{T}_{l,k}^r \in \mathbb{C}^{N \times S}$. The corresponding phase-shift optimization problem can be reformulated as

$$\begin{aligned} \underset{\boldsymbol{t}_l, \Delta}{\text{minimize}} \quad & \sum_{l=1}^{L} \sum_{k \in \mathcal{K}} \frac{\sum_{j \neq l} \left\| \left(\boldsymbol{\theta}_d^H \boldsymbol{T}_{j,k}^{r,H} + \boldsymbol{h}_{j,k}^{l,H} \right) \boldsymbol{t}_j \right\|^2 + \sigma_d^2}{\left\| \left(\boldsymbol{\theta}_d^H \boldsymbol{T}_{l,k}^{r,H} + \boldsymbol{h}_{l,k}^{l,H} \right) \boldsymbol{t}_l \right\|^2}, \\ \text{subject to} \quad & \left| \theta_s^{\mathrm{dl}} \right| = 1, \forall s = 1, \cdots, S. \end{aligned} \tag{36}$$

Problem (36) is in the same form as Problem (30), which means we can use the same strategy to solve the downlink phase-shift optimization.

5 Simulation Results

In this section, we provide some important simulation results to demonstrate the performance of the proposed RIS-assisted multi-cell FL network.

5.1 Experiment Setup

We consider a RIS-assisted two-cell wireless FL network in two-dimensional space where the coordinates of the APs are $(0, 0)$ and $(200, 0)$. The RIS is deployed at the edge of the two cells, i. e., $(100, 0)$. The edge devices of each cell are randomly scattered within a circle with a center of $(90, 0)$ or $(100, 0)$ and a radius of 10 m. We assume that the antennas of the APs and the reflecting elements of the RIS are both arranged in a uniform linear array. In the experiments, the path loss is modeled as $T(d/d_0)^{-\alpha}$ at a distance of $d_0 = 1$ m, where d denotes the link distance and α is the pass loss exponent.

We consider Rician fading for all channels and the channel coefficients are given as

$$\boldsymbol{h}_k = \sqrt{T(d_k/d_0)^{-\alpha}}\left(\sqrt{\frac{\beta}{1+\beta}}\boldsymbol{h}_k^{\mathrm{LoS}} + \sqrt{\frac{\beta}{1+\beta}}\boldsymbol{h}_k^{\mathrm{NLoS}}\right), \quad (37)$$

where $\boldsymbol{h}_k^{\mathrm{LoS}}$ and $\boldsymbol{h}_k^{\mathrm{NLoS}}$ represent the line-of-sight (LoS) and non-line-of-sight (NLoS) components. The Rician factor β is set to be 3. Particularly, we consider the same path loss exponent for all links, which is 2.2. Besides, we set P_d = 30 dBm, and σ_d^2 = σ^2 = −10 dBm, which means the constant q = 1.

In this paper, we adopt the sample-wise loss function and Modified National Institute of Standards and Technology (MNIST) datasets[36] in the process of learning. We assume that each cell performs a different learning task (0 − 4 in Cell 1 and 5 − 9 in Cell 2) and that the learning rate is 0.1. The mini-batch datasets at different cells are 12 and 16, respectively. Next, we make the following specific schemes to compare the performance:

1) Without RIS: This scheme does not consider the RIS, which indicates the channel only contains the direct link between the APs and devices, i.e., $\boldsymbol{\Theta}$ = 0 (for both downlink and uplink communications).

2) Random phase-shift: Under this scheme, the phase-shift matrix is randomly generated in a RIS-assisted system, that is, we only need to optimize the beamforming vectors.

3) Optimal phase-shift: Under such a scheme, we optimize both the beamforming vectors and the phase-shift matrix of the RIS (Algorithm 2).

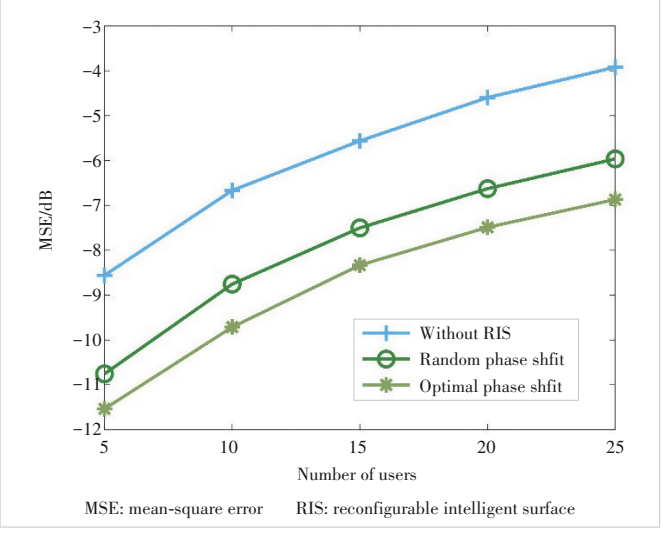
4) Error-free: The scheme is the benchmark of FL, which implies both the downlink model dissemination and uplink gradient aggregation are transmitted in an error-free manner.

5.2 Performance of RIS-Assisted FL Two-Cell System

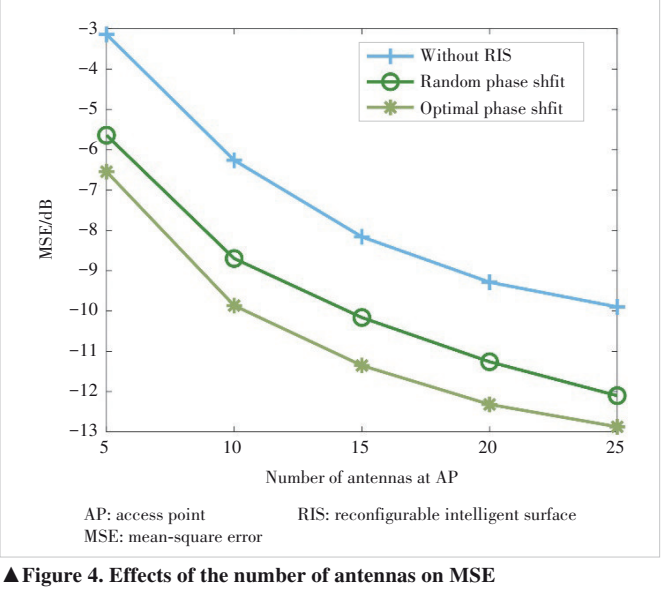
In this subsection, we first present the performance of the uplink aggregation based on AirComp and downlink dissemination error. Then we compare the performance of a two-cell FL system under different schemes.

For the uplink aggregation, the mean-square error (MSE) is a very common performance metric in AirComp[12, 14, 25, 34]. Therefore, we discuss the impact of the number of users, the number of antennas at each AP, and the number of reflecting elements at RIS on the average MSE across all cells. Fig. 3 displays the relationship between the MSE and the number of users, where the number of antennas at AP and the number of elements at RIS are set to be N_1 = N_2 = 10 and S = 30, respectively. It is obvious that the MSE increases with the number of users and deploying the RIS can significantly reduce the value of MSE compared to the absence of the RIS. This is because RIS can perform channel compensation for users at the edge of the corresponding cells with poor signals. On the one hand, with the increase of users, the inter-cell interference is more obvious, which also enlarges the MSE. On the other hand, when a RIS is deployed at the edge of two cells, it can mitigate inter-cell interference. Besides, the RIS with optimal phase-shift is better than that RIS with random phase-shift on MSE, which indicates that the RIS with optimal phase-shift significantly enhances the signal strengths received at the APs. Fig. 4 compares the effects of the different numbers of antennas at AP on MSE, where the number of users per cell is fixed to 10 and the number of elements at RIS is also 30. We observe that the MSE decreases with the number of antennas, due to the diversity gain of antennas. RIS can improve the total MSE performance of the two-cell system. Correspondingly, the RIS with optimal phase shift can also achieve better MSE performance than the other two baseline schemes.

To compare the effect of the number of RIS elements on



▲Figure 3. Relationship between MSE and the number of users



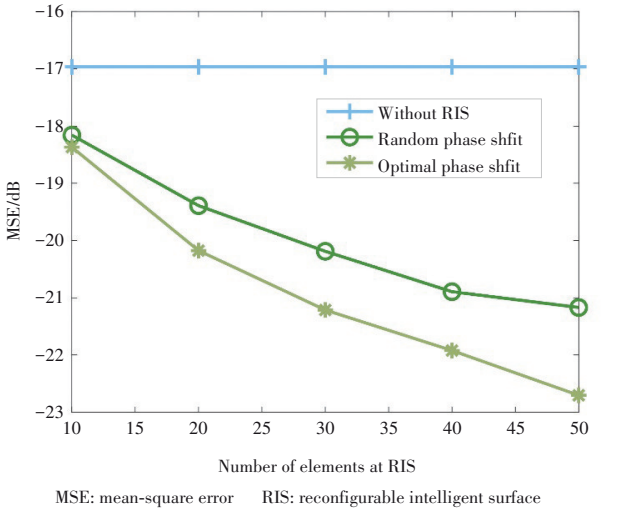
▲Figure 4. Effects of the number of antennas on MSE

MSE, we first set the number of users and antennas at AP to 10, i.e., $N_1 = N_2 = K_1 = K_2 = 10$, and then we fix the location of users in each cell to avoid the influence of channel randomness. Fig. 5 shows that the number of elements at a RIS has a positive tendency correlated with the MSE, and as a result, the performance gradually gets better as the number of elements increases. In addition, the gap between random phase-shift and optimal phase-shift becomes larger and larger as the number of elements increases, which demonstrates the benefits of the optimal phase-shift scheme.

Since the downlink optimization and the uplink optimization have similar forms and are solved by the same algorithm, the impacts of the number of users and antennas at AP and the elements at the RIS on the downlink MSE have the same performance trend as those on the uplink MSE. We further compare the downlink errors in the case that $K_1 = K_2 = 10$, $N_1 = N_2 = 10$, and $S = 30$, i.e., $\sum_{k \in \mathcal{K}} \mathbb{E}[\|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\|^2]$. The results are shown in Table 1.

According to the results, the RIS with optimal phase shift still achieves the best performance, despite the small gaps in these errors. Moreover, we observe that the downlink error is much smaller than the uplink MSE, which indicates the downlink error has little effect on the convergence result of the overall system when the number of users is relatively small and $M = 10$ (the learning rate is $\zeta = 0.1$).

Next, we compare the performance of these schemes in the proposed two-cell FL system, where the number of users and that of antennas at AP in each cell are 5, and the number of elements at RIS is set by 15. Each cell performs the same FL task with one local update in different mini-batch datasets. In order to compare the performance of the entire system, we average the train loss and test accuracy of the two cells and the results are shown in Fig. 6. Fig. 6 (a) shows, although the training loss of these schemes varies, all the schemes can achieve convergence and converge fast. Based on the proposed schemes, the RIS with optimal phase-shift scheme can demonstrate its advantages to enhance the performance of FL. From Fig. 6 (b), we notice that the RIS with optimal phase shift can achieve approximately 85% accuracy, the RIS with random phase-shift can get 83.5% accuracy, and the scheme without RIS only attains 82.7% accuracy, which proves that the RIS-assisted schemes can improve the performance of FL. To clearly show the effectiveness of our proposed system, we make additional time statistics for each scheme and each scheme runs for almost 800 s under $K = 5$, $M = 15$, $N = 5$, and $T = 300$, indicating that the proposed system can converge quickly. In



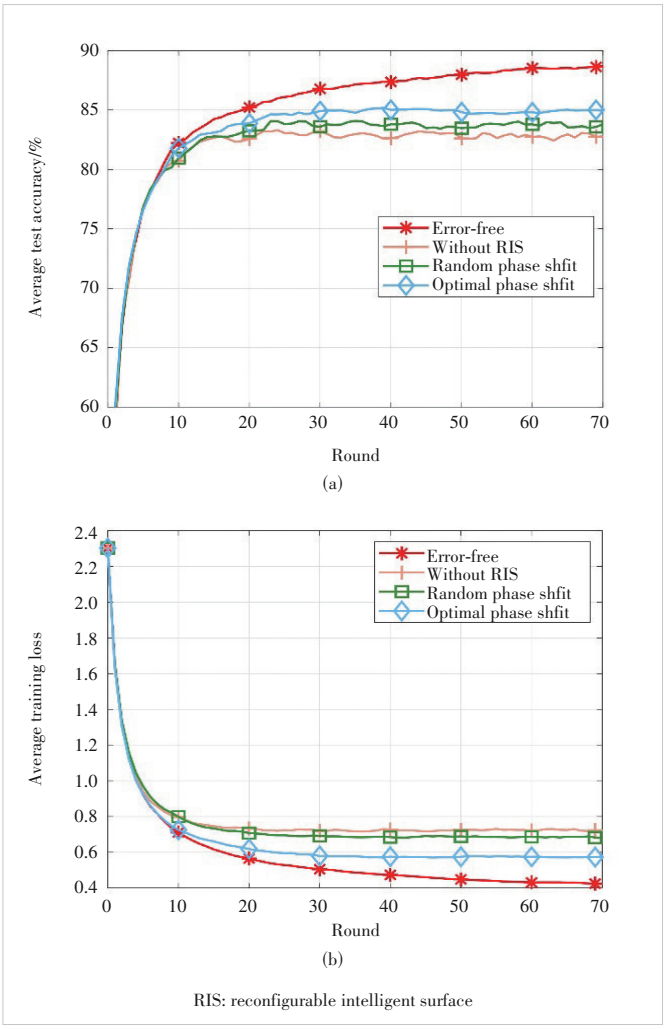
MSE: mean-square error RIS: reconfigurable intelligent surface

▲ **Figure 5. Relationship between MSE and the number of elements at RIS**

▼ **Table 1. Comparison of downlink errors**

Scheme	Error/dB
Without RIS	−52.77
Random PS	−53.16
Optimal PS	−53.91

PS: phase-shift RIS: reconfigurable intelligent surface



RIS: reconfigurable intelligent surface

▲ **Figure 6. Performance of different schemes in the proposed two-cell FL system: (a) training loss vs communication rounds; (b) test accuracy vs communication rounds**

summary, RIS can compensate for the signal degradation of edge users and thereby decrease the error of communication. Moreover, we can adjust the phase-shift matrix of RIS to mitigate the inter-cell interference.

6 Conclusions and Future Work

In this paper, we develop a RIS-assisted AirComp-based two-cell FL wireless network, where each cell learns a different FL task and both the effects of downlink and uplink communications are considered. We first analyze the convergence of FL in the proposed system and show that the convergence is mainly influenced by the error of downlink and uplink transmissions. To enhance the performance of FL, we formulate the joint uplink and downlink optimization problem to minimize the optimality gap. To solve the problem, we divide the optimization problem into two separate subproblems. The beamforming vector and phase-shift matrix in each subproblem are optimized by alternative optimization based on SCA. In the end, simulation results show the performance and advantage of our proposed system and optimization algorithm.

In this work, we mainly focus on a scenario where a RIS assists two cells. In our future work, we will consider the scenario of a multi-RIS-assisted multi-cell wireless network, which makes the system model more complex. Since the placement of multi-RIS has a great impact on multi-cell performance, it is necessary to improve the average learning performance of all cells, as well as to avoid the poor performance of one cell. Most existing RISes only support a reflection or transmission mode. A new simultaneous transmitting and reflecting reconfigurable intelligent surface (STAR-RIS) can achieve full spatial coverage and have the advantage of adjusting more degrees of freedom. Therefore, promoting the deployment of SART-RIS is conducive to the implementation of more application scenarios.

Appendix

Proof of Theorem 1

For presentation clarity, we omit the cell index in the following analysis. According to Eqs. (5), (7) and (14), we have

$$\boldsymbol{w}^{t+1}=\boldsymbol{w}^{t}-\frac{\zeta^{t}}{K}\left(\sum_{k\in\mathcal{K}_{i}}\nabla F_{k}\left(\boldsymbol{w}^{t}+\boldsymbol{e}_{k}^{\mathrm{dl}}\right)+\boldsymbol{e}_{t}^{\mathrm{ul}}\right). \tag{38}$$

Let $\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)=\frac{1}{K}\sum_{k\in\mathcal{K}_{i}}\nabla F_{k}\left(\boldsymbol{w}^{t}+\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right)$ and $\boldsymbol{e}_{t}^{\mathrm{up}}=\frac{1}{K}\boldsymbol{e}_{t}^{\mathrm{ul}}$, and then the global update can be rewritten by

$$\boldsymbol{w}^{t+1}=\boldsymbol{w}^{t}-\zeta_{t}\left(\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)+\boldsymbol{e}_{t}^{\mathrm{up}}\right). \tag{39}$$

According to Assumption 1, we obtain

$$F\left(\boldsymbol{w}^{t+1}\right)-F\left(\boldsymbol{w}^{t}\right)\leqslant\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\right\rangle+\frac{M}{2}\left\|\boldsymbol{w}^{t+1}-\boldsymbol{w}^{t}\right\|^{2}=\frac{M\zeta_{t}^{2}}{2}\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)+\boldsymbol{e}_{t}^{\mathrm{up}}\right\|^{2}-\zeta_{t}\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)+\boldsymbol{e}_{t}^{\mathrm{up}}\right\rangle. \tag{40}$$

By taking the expectation on both sides of Eq. (40) and utilizing $\mathbb{E}\left[\boldsymbol{e}_{t}^{\mathrm{up}}\right]=0$, we have

$$\begin{aligned}\mathbb{E}\left[F\left(\boldsymbol{w}^{t+1}\right)-F\left(\boldsymbol{w}^{t}\right)\right]&\leqslant-\zeta_{t}\mathbb{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)+\boldsymbol{e}_{t}^{\mathrm{up}}\right\rangle\right]+\frac{M\zeta_{t}^{2}}{2}\mathbb{E}\left[\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)+\boldsymbol{e}_{t}^{\mathrm{up}}\right\|^{2}\right]=\\&-\zeta_{t}\mathbb{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\rangle\right]+\frac{M\zeta_{t}^{2}}{2}\mathbb{E}\left[\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\|^{2}\right]+\frac{M\zeta_{t}^{2}}{2}\mathbb{E}\left[\left\|\boldsymbol{e}_{t}^{\mathrm{up}}\right\|^{2}\right].\end{aligned} \tag{41}$$

We let $T_{1}=\mathbb{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\rangle\right]$ and $T_{2}=\mathbb{E}\left[\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\|^{2}\right]$. First, we make a upper bound of T_{2}, and then we have

$$\begin{aligned}T_{2}&=\mathbb{E}\left[\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\pm\nabla F\left(\boldsymbol{w}^{t}\right)\right\|^{2}\right]=\\&\mathbb{E}\left[\left\|\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)-\nabla F\left(\boldsymbol{w}^{t}\right)\right\|^{2}\right]+\mathbb{E}\left[\left\|\nabla F\left(\boldsymbol{w}^{t}\right)\right\|^{2}\right]+\\&2\mathrm{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)-\nabla F\left(\boldsymbol{w}^{t}\right)\right\rangle\right]\leqslant\\&\frac{M^{2}}{K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left\|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right\|^{2}\right]-\mathbb{E}\left[\left\|\nabla F\left(\boldsymbol{w}^{t}\right)\right\|^{2}\right]+\\&2\mathbb{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\rangle\right],\end{aligned} \tag{42}$$

where $\boldsymbol{a}\pm\boldsymbol{b}=\boldsymbol{a}+\boldsymbol{b}-\boldsymbol{b}$ and the last inequality follows M-smoothness property $\|\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{y})\|\leqslant M\|\boldsymbol{x}-\boldsymbol{y}\|$. Therefore,

$$\mathbb{E}\left[F\left(\boldsymbol{w}^{t+1}\right)-F\left(\boldsymbol{w}^{t}\right)\right]\leqslant-\zeta_{t}(1-M\zeta_{t})\mathbb{E}\left[\left\langle\nabla F\left(\boldsymbol{w}^{t}\right),\nabla F\left(\hat{\boldsymbol{w}}^{t}\right)\right\rangle\right]+\frac{M^{3}\zeta_{t}^{2}}{2K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left\|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right\|^{2}\right]+\frac{M\zeta_{t}^{2}}{2}\mathbb{E}\left[\left\|\boldsymbol{e}_{t}^{\mathrm{up}}\right\|^{2}\right]-\frac{M\zeta_{t}^{2}}{2}\mathbb{E}\left[\left\|\nabla F\left(\boldsymbol{w}^{t}\right)\right\|^{2}\right]. \tag{43}$$

By setting $0\leqslant\zeta_{t}\equiv\zeta=\frac{1}{M}$, we obtain

$$\mathbb{E}\left[F\left(\boldsymbol{w}^{t+1}\right)-F\left(\boldsymbol{w}^{t}\right)\right]\leqslant-\frac{1}{2M}\mathbb{E}\left[\|\nabla F(\boldsymbol{w}^{t})\|^{2}\right]+\frac{M}{2K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\|^{2}\right]+\frac{1}{2M}\mathbb{E}\left[\|\boldsymbol{e}_{t}^{\mathrm{up}}\|^{2}\right]. \tag{44}$$

Based on Assumption 2, we have $\|\nabla F(\boldsymbol{w}^{t})\|^{2}\geqslant 2\mu\left(F\left(\boldsymbol{w}^{t}\right)-F\left(\boldsymbol{w}^{\star}\right)\right)$.

Thus, Eq. (44) can be represented as

$$\mathbb{E}\left[F\left(\boldsymbol{w}^{t+1}\right)-F\left(\boldsymbol{w}^{t}\right)\right]\leqslant-\frac{\mu}{M}\mathbb{E}\left[F(\boldsymbol{w}^{t})-F(\boldsymbol{w}^{\star})\right]+\frac{M}{2K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\|^{2}\right]+\frac{1}{2M}\mathbb{E}\left[\|\boldsymbol{e}_{t}^{\mathrm{up}}\|^{2}\right]. \tag{45}$$

Rearranging Eq. (45) and applying recursion, we have

$$\mathbb{E}[F(\boldsymbol{w}^T)-F(\boldsymbol{w}^\star)]\leqslant\rho^T\mathbb{E}[F(\boldsymbol{w}^0)-F(\boldsymbol{w}^\star)]+\sum_{t=0}^{T-1}\rho^{T-t-1}\left(\frac{M}{2K}\sum_{k\in\mathcal{K}}\mathbb{E}\left[\left|\left|\boldsymbol{e}_{k,t}^{\mathrm{dl}}\right|\right|^2\right]+\frac{1}{2MK^2}\mathbb{E}\left[\left|\left|\boldsymbol{e}_t^{\mathrm{ul}}\right|\right|^2\right]\right),\quad(46)$$

where $\rho=1-\mu/M$. Therefore we get Theorem 1.

References

[1] LETAIEF K B, SHI Y M, LU J M, et al. Edge artificial intelligence for 6G: vision, enabling technologies, and applications [J]. IEEE journal on selected areas in communications, 2021, 40(1): 5 – 36. DOI: 10.1109/JSAC.2021.3126076
[2] LETAIEF K B, CHEN W, SHI Y M, et al. The roadmap to 6G: AI empowered wireless networks [J]. IEEE communications magazine, 2019, 57(8): 84 – 90. DOI: 10.1109/mcom.2019.1900271
[3] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications [J]. ACM transactions on intelligent systems and technology, 2019, 10 (2): 1 – 19. DOI: 10.1145/3298981
[4] WEN D Z, JEON K J, HUANG K B. Federated dropout—a simple approach for enabling federated learning on resource constrained devices [J]. IEEE wireless communications letters, 2022, 11(5): 923 – 927. DOI: 10.1109/LWC.2022.3149783
[5] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]//20th International Conference on Artificial Intelligence and Statistics (AISTATS). JMLR, 2017: 1273 – 1282. DOI: 10.48550/arXiv.1602.05629
[6] XU W, YANG Z H, NG D W K, et al. Edge learning for B5G networks with distributed signal processing: semantic communication, edge computing, and wireless sensing [J]. IEEE journal of selected topics in signal processing, 2023: 1 – 31. DOI: 10.1109/jstsp.2023.3239189
[7] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks [J]. IEEE transactions on wireless communications, 2021, 20(1): 269 – 283. DOI: 10.1109/TWC.2020.3024629
[8] YANG H H, LIU Z Z, QUEK T Q S, et al. Scheduling policies for federated learning in wireless networks [J]. IEEE transactions on communications, 2020, 68(1): 317 – 333. DOI: 10.1109/tcomm.2019.2944169
[9] YANG P, JIANG Y N, WANG T, et al. Over-the-air federated learning via second-order optimization [J]. IEEE transactions on wireless communications, 2022, 21(12): 10560 – 10575. DOI: 10.1109/TWC.2022.3185156
[10] NAZER B, GASTPAR M. Computation over multiple-access channels [J]. IEEE transactions on information theory, 2007, 53(10): 3498 – 3516. DOI: 10.1109/tit.2007.904785
[11] CHEN L, QIN X W, WEI G. A uniform-forcing transceiver design for over-the-air function computation [J]. IEEE wireless communications letters, 2018, 7 (6): 942 – 945. DOI: 10.1109/LWC.2018.2840157
[12] YANG Y H, ZHOU Y, WU Y L, et al. Differentially private federated learning via reconfigurable intelligent surface [J]. IEEE Internet of Things journal, 2022, 9(20): 19728 – 19743. DOI: 10.1109/JIOT.2022.3168066
[13] WANG Z B, ZHAO Y P, ZHOU Y, et al. Over-the-air computation: foundations, technologies, and applications [EB/OL]. [2022-10-19]. https://arxiv.org/abs/2210.10524
[14] YANG K, JIANG T, SHI Y M, et al. Federated learning via over-the-air computation [J]. IEEE transactions on wireless communications, 2020, 19(3): 2022 – 2035. DOI: 10.1109/TWC.2019.2961673
[15] FANG W Z, YU Z Y, JIANG Y N, et al. Communication-efficient stochastic zeroth-order optimization for federated learning [J]. IEEE transactions on signal processing, 2022, 70: 5058 – 5073. DOI: 10.1109/TSP.2022.3214122
[16] FU M, SHI Y M, ZHOU Y. Federated learning via unmanned aerial vehicle [EB/OL]. [2022-10-20]. https: //arxiv.org/abs/2210.10970
[17] LIM W Y B, GARG S, XIONG Z H, et al. UAV-assisted communication efficient federated learning in the era of the artificial intelligence of things [J]. IEEE network, 2021, 35(5): 188 – 195. DOI: 10.1109/MNET.002.2000334
[18] NI W L, LIU Y W, YANG Z H, et al. Federated learning in multi-RIS-aided systems [J]. IEEE Internet of Things journal, 2022, 9(12): 9608 – 9624. DOI: 10.1109/JIOT.2021.3130444
[19] WANG Z B, QIU J H, ZHOU Y, et al. Federated learning via intelligent reflecting surface [J]. IEEE transactions on wireless communications, 2022, 21 (2): 808 – 822. DOI: 10.1109/twc.2021.3099505
[20] YANG K, SHI Y M, ZHOU Y, et al. Federated machine learning for intelligent IoT via reconfigurable intelligent surface [J]. IEEE network, 2020, 34(5): 16 – 22. DOI: 10.1109/MNET.011.2000045
[21] LIU H, YUAN X J, ZHANG Y J A. Joint communication-learning design for RIS-assisted federated learning [C]//IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2021: 1 – 6. DOI: 10.1109/ICCWorkshops50388.2021.9473672
[22] YIN T, LI L X, MA D H, et al. FLIGHT: federated learning with IRS for grouped heterogeneous training [J]. Journal of communications and information networks, 2022, 7(2): 135 – 144. DOI: 10.23919/jcin.2022.9815197
[23] HU L, WANG Z B, ZHU H B, et al. RIS-assisted over-the-air federated learning in millimeter wave MIMO networks [J]. Journal of communications and information networks, 2022, 7(2): 145 – 156. DOI: 10.23919/jcin.2022.9815198
[24] WU Q Q, ZHANG R. Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming [J]. IEEE transactions on wireless communications, 2019, 18(11): 5394 – 5409. DOI: 10.1109/TWC.2019.2936025
[25] FANG W Z, JIANG Y N, SHI Y M, et al. Over-the-air computation via reconfigurable intelligent surface [J]. IEEE transactions on communications, 2021, 69(12): 8612 – 8626. DOI: 10.1109/tcomm.2021.3114791
[26] HUANG C W, ZAPPONE A, ALEXANDROPOULOS G C, et al. Reconfigurable intelligent surfaces for energy efficiency in wireless communication [J]. IEEE transactions on wireless communications, 2019, 18(8): 4157 – 4170. DOI: 10.1109/TWC.2019.2922609
[27] WEINBERGER K, AHMAD A A, SEZGIN A, et al. Synergistic benefits in IRS- and RS-enabled C-RAN with energy-efficient clustering [J]. IEEE transactions on wireless communications, 2022, 21(10): 8459 – 8475. DOI: 10.1109/TWC.2022.3166393
[28] ZHAI X F, HAN G J, CAI Y L, et al. Joint beamforming aided over-the-air computation systems relying on both BS-side and user-side reconfigurable intelligent surfaces [J]. IEEE transactions on wireless communications, 2022, 21 (12): 10766 – 10779. DOI: 10.1109/TWC.2022.3187156
[29] WANG Z B, ZHOU Y, SHI Y M, et al. Interference management for over-the-air federated learning in multi-cell wireless networks [J]. IEEE journal on selected areas in communications, 2022, 40(8): 2361 – 2377. DOI: 10.1109/JSAC.2022.3180799
[30] PAN C H, REN H, WANG K Z, et al. Multicell MIMO communications relying on intelligent reflecting surfaces [J]. IEEE transactions on wireless communications, 2020, 19(8): 5218 – 5233. DOI: 10.1109/twc.2020.2990766
[31] LUO C H, LI X, JIN S, et al. Reconfigurable intelligent surface-assisted multi-cell MISO communication systems exploiting statistical CSI [J]. IEEE wireless communications letters, 2021, 10(10): 2313 – 2317. DOI: 10.1109/LWC.2021.3100427
[32] XIE H L, XU J, LIU Y F. Max-min fairness in IRS-aided multi-cell MISO systems via joint transmit and reflective beamforming [C]//IEEE International Conference on Communications (ICC). IEEE, 2020: 1 – 6. DOI: 10.1109/ICC40277.2020.9148858
[33] NI W L, LIU X, LIU Y W, et al. Resource allocation for multi-cell IRS-aided NOMA networks [J]. IEEE transactions on wireless communications, 2021, 20 (7): 4253 – 4268. DOI: 10.1109/TWC.2021.3057232
[34] LI J, FU M, ZHOU Y, et al. Double-RIS assisted over-the-air computation [C]// IEEE Globecom Workshops (GC Wkshps). IEEE, 2022: 1 – 6. DOI: 10.1109/GCWkshps52748.2021.9682077
[35] ABARI O, RAHUL H, KATABI D, et al. AirShare: distributed coherent transmission made seamless [C]//IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015: 1742 – 1750. DOI: 10.1109/INFOCOM.2015.7218555

[36] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278 – 2324. DOI: 10.1109/5.726791

Biographies

WANG Yiji received his BS degree from Zhejiang University City College, China in 2020. He is currently pursuing his master's degree with the School of Information Science and Technology, ShanghaiTech University, China. His research interests include federated learning and wireless communications.

WEN Dingzhu (wendzh@shanghaitech.edu.cn) received his bachelor's and master's degrees from Zhejiang University, China in 2014 and 2017, respectively, and PhD degree from The University of Hong Kong, China in 2021. Subsequently, he joined ShanghaiTech University, China. He is currently an assistant professor at the School of Information Science and Technology there. His research interests include edge intelligence, integrated sensing, computation and communication, over-the-air computation, in-band full-duplex communications, etc.

MAO Yijie is an assistant professor at the School of Information Science and Technology, ShanghaiTech University, China. She received her BE degree from Beijing University of Posts and Telecommunications, China and BE (Hons.) degree from the Queen Mary University of London in 2014. She received her PhD degree from the Electrical and Electronic Engineering (EEE) Department, The University of Hong Kong, China in 2018. She was a postdoctoral research fellow at The University of Hong Kong from 2018 to 2019 and a postdoctoral research associate with the Department of the EEE at the Imperial College London, UK from 2019 to 2021. She is a senior member of China Institute of Communications. She is currently serving as an editor of *IEEE Communications Letters* and a guest editor of two special issues of *IEEE Journal on Selected Areas in Communications* and *IEEE Open Journal of the Communications Society*.

SHI Yuanming received his BS degree in electronic engineering from Tsinghua University, China in 2011. He received his PhD degree in electronic and computer engineering from The Hong Kong University of Science and Technology (HKUST), China in 2015. Since September 2015, he has been with the School of Information Science and Technology, ShanghaiTech University, China, where he is currently a tenured associate professor. He visited University of California, Berkeley, USA from October 2016 to February 2017. His research areas include optimization, machine learning, wireless communications, and their applications to 6G, IoT and edge AI. He was a recipient of the 2016 IEEE Marconi Prize Paper Award in Wireless Communications, the 2016 Young Author Best Paper Award by the IEEE Signal Processing Society, and the 2021 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He is also an editor of *IEEE Transactions on Wireless Communications*, *IEEE Journal on Selected Areas in Communications*, and *Journal of Communications and Information Networks*.

Hierarchical Federated Learning: Architecture, Challenges, and Its Implementation in Vehicular Networks

YAN Jintao[1], CHEN Tan[1], XIE Bowen[1], SUN Yuxuan[2], ZHOU Sheng[1], NIU Zhisheng[1]
(1. Tsinghua University, Beijing 100084, China;
2. Beijing Jiaotong University, Beijing 100044, China)

DOI: 10.12142/ZTECOM.202301005
https://kns.cnki.net/kcms/detail/34.1294.TN.20230220.1541.002.html, published online February 21, 2023
Manuscript received: 2022-11-04

Abstract: Federated learning (FL) is a distributed machine learning (ML) framework where several clients cooperatively train an ML model by exchanging the model parameters without directly sharing their local data. In FL, the limited number of participants for model aggregation and communication latency are two major bottlenecks. Hierarchical federated learning (HFL), with a cloud-edge-client hierarchy, can leverage the large coverage of cloud servers and the low transmission latency of edge servers. There are growing research interests in implementing FL in vehicular networks due to the requirements of timely ML training for intelligent vehicles. However, the limited number of participants in vehicular networks and vehicle mobility degrade the performance of FL training. In this context, HFL, which stands out for lower latency, wider coverage and more participants, is promising in vehicular networks. In this paper, we begin with the background and motivation of HFL and the feasibility of implementing HFL in vehicular networks. Then, the architecture of HFL is illustrated. Next, we clarify new issues in HFL and review several existing solutions. Furthermore, we introduce some typical use cases in vehicular networks as well as our initial efforts on implementing HFL in vehicular networks. Finally, we conclude with future research directions.

Keywords: hierarchical federated learning; vehicular network; mobility; convergence analysis

Citation (IEEE Format): J. T. Yan, T. Chen, B. W. Xie, and et al., "Hierarchical federated learning: architecture, challenges, and its implementation in vehicular networks," *ZTE Communications*, vol. 21, no. 1, pp. 38 – 45, Mar. 2022. doi: 10.12142/ZTECOM.202301005.

1 Introduction

Recently, the evolution of intelligent technologies gives rise to a wide range of emerging applications including the Internet of Things (IoT), autonomous driving, and so on. While opening up new ways of life for users, these applications also produce numerous data scattered on mobile devices. Transmitting these data to a centralized server for traditional machine learning (ML) is no longer capable due to limited communication resources, tight latency requirements and stringent privacy concerns. As a result, federated learning (FL) is proposed as a distributed learning solution, where multiple mobile devices and a parameter server cooperatively train an ML model by only exchanging the model parameters without directly sharing their local data.

In recent years, many works have been done to deal with the different challenges of FL[1]. Among them, communication efficiency is one of the most important issues[2]. Many FL frameworks consider the cloud server as the parameter server, but the communication between clients and the cloud server is inefficient and unpredictable. Federated edge learning (FEEL)[3], where the clients share the ML model parameters with edge servers, has been proposed to reduce communication latency. However, the edge servers in FEEL have limited coverage and the number of clients for FL training cannot meet the requirements, resulting in the degradation of training performance. Therefore, it is necessary to characterize the tradeoff between communication latency and training performance.

To deal with this issue, the concept of hierarchical FL (HFL) has been proposed[4-5], which leverages the large coverage of the cloud server and the high communication efficiency of the edge server. This architecture consists of one cloud server, multiple edge servers, and a multitude of clients. In HFL, the clients update their local parameters and send them to the edge servers for edge aggregations as conventional FL does. The difference is that after several rounds of edge aggregations, multiple edge servers send their parameters to a cloud

The work of YAN Jintao, CHEN Tan, XIE Bowen, ZHOU Sheng and NIU Zhisheng are sponsored in part by the National Key R&D Program of China under Grant No. 2020YFB1806605, the National Natural Science Foundation of China under Grant Nos. 62022049, 62111530197, and 61871254, and OPPO. The work of SUN Yuxuan is supported by the Fundamental Research Funds for the Central Universities under Grant No. 2022JBXT001.

server for cloud aggregation, which allows more clients to be involved in the framework. Experimental results and theoretical analysis have shown that this client-edge-cloud FL architecture has higher convergence speed and less training time compared with the conventional framework[5].

During the last several years, FL has witnessed its potential in vehicular networks. The advantage of implementing FL in vehicular networks is twofold. First, FL can satisfy the latency and privacy requirements that the applications in vehicular networks, such as trajectory planning and traffic flow optimization, call for. Second, intelligent vehicles have computation and communication capabilities and can sample abundant data for training[6]. There have been many papers on the implementation of FL in vehicular networks. In Ref. [7], an FL-based approach is proposed to allocate the power and resource for ultra-reliable low-latency communications in vehicular networks. In Ref. [8], FL is used to update an edge caching scheme for vehicular networks, which considers the cached content and vehicular mobility. Considering the computation and communication resources and local dataset of vehicles, the authors of Ref. [9] propose a joint vehicle selection and resource allocation scheme for FL training. In Ref. [10], the vehicle speed and position are taken into consideration and an optimization problem is formulated for resource allocation for FL.

However, implementing FL in vehicular networks may be more challenging than that in conventional wireless networks[11]. First, the ML application in vehicular networks has more stringent requirements for latency. This is because vehicles may leave the coverage of the central server due to mobility before successfully uploading their updated local model to the server. Second, since the physical distances between vehicles are much larger than those of humans or mobile devices, the number of clients participating in model aggregation in vehicular networks is much lower than that in conventional wireless networks, which degrades the convergence performance of FL. In this context, HFL stands out for its properties of lower latency, wider coverage, and more participants, inspiring us to search for the possibility of implementing HFL in vehicular networks.

There are some existing surveys and tutorials on FL, as shown in Table 1. In Ref. [1], a comprehensive survey of FL in wireless networks is provided, and research directions including compression and sparsification, convergence analysis, wireless resource management, and FL training method design are presented. The authors of Ref. [2] focus on minimizing the communication and computation latency and introduce the concept of timely edge learning. The key challenges and solutions to the timely issues are discussed. In Ref. [11], the concept of FL is combined with mobile edge computing (MEC) and a comprehensive survey of FL and MEC is provided. In Ref. [12], the implementation of FL in vehicular networks is studied and the major challenges are analyzed from a learning and communication perspective. In this work, we provide a comprehensive review of HFL and explore the feasibility of implementing HFL in vehicular networks.

▼Table 1. Existing surveys on FL

Highlight	Reference
A comprehensive survey of FL in wireless networks	Ref. [1]
A tutorial on timely edge learning, aiming to minimize the communication and computation latency in FL training	Ref. [2]
A comprehensive survey of FL and MEC	Ref. [11]
A tutorial on the implementation of FL in vehicular networks and the major challenges of learning and communications	Ref. [12]

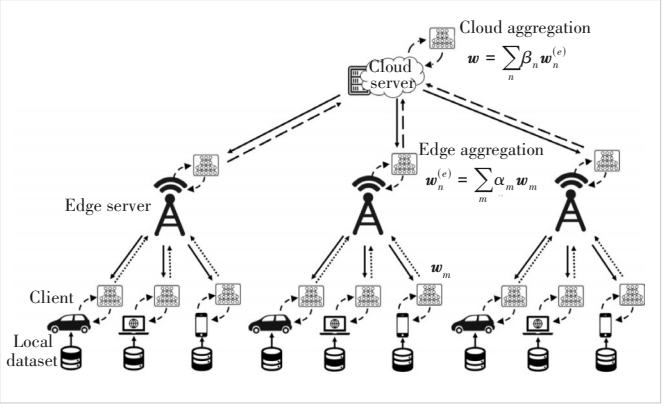
FL: federated learning MEC: mobile edge computing

The rest of this paper is organized as follows. The HFL architecture is introduced in Section 2. In Section 3, we clarify the new issues and challenges in HFL compared with FL and provide a review of existing works dealing with these issues. In Section 4, we introduce the typical use cases of HFL in vehicular networks. Section 4 concludes this paper and gives some future research directions in this field.

2 HFL Architecture

In a typical HFL system, a cloud, some edges and several clients collaboratively train an ML model. The cloud covers all the edges and each edge covers some of the clients. All of the participants initialize a model of the same parameters and perform cloud epochs. Each cloud epoch is composed of edge learning stages and a cloud aggregation stage. During the edge learning stage, each edge, together with clients under its coverage, trains the learning model in the way of FL for some iterations. During the cloud aggregation stage, edges transmit their model parameters or gradients to the cloud. The cloud aggregates the parameters or gradients to update the global model and broadcasts the global model to edges. The cloud epoch is repeated until the global model converges. The training procedure is illustrated in Fig. 1.

Different from clients in FL, who are always connected to the same parameter server, those clients in HFL can be associated with different edges during training. First, in cellular net-



▲Figure 1. Architecture of a hierarchical federated learning system

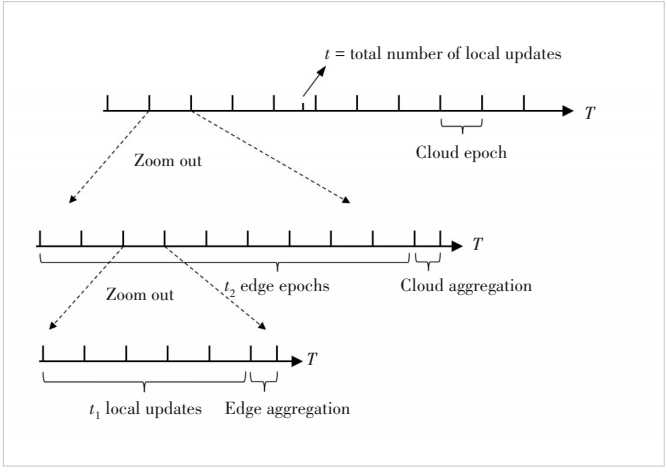
works, the coverage areas of cells are generally overlapping, so clients in the overlapped area of some edges can be associated with either of them. Second, in a scenario of wireless communications, especially in vehicular networks, clients may be moving, which means they can step from the coverage of one edge into that of another during training, while generally staying in the range of the cloud. Therefore, in HFL, edges may need to reconstruct connections with clients at the beginning of each iteration.

To formulate the training procedure, we assume there are M clients and N edges and denote $\boldsymbol{w}_m(t)$ as a client's m-th local model parameters at the t-th local update. Assume the clients perform local updates t_1 before edge aggregation and the edges perform FL iterations t_2 before cloud aggregation. For client m, given the loss function $F_m(\cdot)$, learning rate η and the set of clients that are associated with the same edge at the t-th local update $\mathcal{E}_m^{(t)}$, the local model evolves as follows:

$$\tilde{\boldsymbol{w}}_m(t) = \boldsymbol{w}_m(t-1) - \eta \nabla F_m(\boldsymbol{w}_m(t-1)),$$

$$\boldsymbol{w}_m(t) = \begin{cases} \tilde{\boldsymbol{w}}_m(t), t_1 \nmid t \\ \sum_{i \in \mathcal{E}_m^{(t)}} \alpha_i \tilde{\boldsymbol{w}}_i(t), t_1 \mid t, t_1 t_2 \nmid t \\ \sum_{j=1}^{N} \beta_j \sum_{i \in \mathcal{E}_m^{(t)}} \alpha_i \tilde{\boldsymbol{w}}_i(t), t_1 t_2 \mid t, \end{cases}$$

where α_i and β_j are edge and cloud aggregation weights separately, with $\sum_{i \in \mathcal{E}_m^{(t)}} \alpha_i = 1$ and $\sum_j \beta_j = 1$. Here $a \mid b$ means b is divisible by a. On the opposite, $a \nmid b$ means b is not divisible by a. The timescale of HFL training is further shown in Fig. 2, which demonstrates the relationships of t, t_1 and t_2 more clearly.



▲Figure 2. Timescale of hierarchical federated learning (HFL) training

3 Overview of New Research Issues in HFL

Compared with FL, HFL brings many new research issues, both theoretically and practically. From the theoretical perspective, the convergence analysis for HFL is more complex because of the multi-layer architecture. From the practical perspective, the resource management strategies for HFL should not only focus on allocating the wireless resources under one server, but also arrange resources among different edge servers. Also, the popularity of HFL gives rise to many new considerations, such as HFL with device-to-device (D2D) communications and mobility-aware HFL. We provide a survey on HFL based on these three categories: convergence analysis, resource management, and new considerations of HFL. Note that these three categories might overlap with each other. For instance, the convergence analysis results might be used to design the resource allocation strategy in some works.

3.1 Convergence Analysis

In FL, convergence analysis illustrates how different factors influence the FL training performance, and thus can be used as a guideline for FL system design. In HFL, the convergence analysis is more complex. For FL, the clients only perform local updates before global aggregation. However, edge aggregation is conducted before global aggregation, which results in a loose bound of convergence analysis. Many works on convergence analysis for HFL have been done. In Ref. [5], an HFL framework is proposed and the convergence analysis of this framework is provided. By investigating how the distributed weights deviate from the centralized sequence, the authors give an upper bound for the deviation. The results show how the edge and the cloud intervals influence the convergence performance for both convex and non-convex loss functions. Following this work, the authors of Ref. [13] provide a tighter convergence bound. In this work, model quantization is adopted to improve communication efficiency, and the edge and cloud aggregation intervals are optimized based on the theoretical results to improve the training performance. The authors of Ref. [14] assume a graph topology where each edge is considered as a node in the graph, and occasionally averages its model parameters with adjacent nodes in a decentralized manner. Furthermore, a probabilistic approach is adopted for analyzing local updates. Convergence analysis of this scenario is then provided, showing the influence of local iterations, edge epochs, cloud epochs, network topology and node heterogeneity on the convergence performance. Ref. [15] is the first work that takes both data heterogeneity and stochastic gradient descent into consideration for convergence analysis. By denoting client-edge and edge-cloud data divergence, data heterogeneity is connected to the convergence bound and a worst-case upper bound for convergence is provided. The convergence bound shows that local aggregates accelerate the convergence speed of the global model by a "sandwich" behavior. The results are also extended to the cases in which the group-

ing is random or there are more than three layers.

However, most of the above papers consider a static topology. In vehicular networks, the mobility of clients may degrade model convergence, which should be taken into consideration. The authors of Ref. [16] propose a mobility-aware HFL framework. First, the HFL framework with mobile clients is modeled by a Markov chain. Then, convergence analysis is provided, showing how user mobility influences training performance. Based on the theoretical analysis, the local update mode and access scheme are modified to reduce the impact of client mobility. Experimental results illustrate that the proposed scheme can outperform the baselines, especially when the data heterogeneity or user mobility is high or the number of users is small.

3.2 Resource Management

Resource management is an important issue in FL. It means how the communication bandwidth, power and computing resources are allocated to clients under the coverage of one server. In HFL, there is more than one edge server and new issues arise.

One new issue in HFL is edge association, which is defined to find which clients should be allocated to which edge server. In Ref. [17], a joint resource allocation and edge association problem is formulated under HFL. The authors first propose the architecture of HFL and an optimization problem that aims to minimize both latency and energy consumption. Then, this problem is decomposed into two subproblems: a resource allocation problem and an edge association one. The resource allocation problem is proved to be convex and the optimization value can be reached. The edge association problem is solved via an iterative global cost reduction adjustment method. Simulation results show that the proposed scheme can outperform the baselines in terms of FL training performance with low latency and energy consumption. The authors of Ref. [18] focus on the interactions and limited rationalities of the clients. A dynamic resource allocation and edge association problem is proposed based on the game theory in self-organizing HFL frameworks. The edge association problem is solved via a lower-level evolutionary game and the resource allocation problem is solved via an upper-level Stackelberg differential game. Experiments show that the proposed scheme can well suit the dynamics of the HFL system. In Ref. [19], the effect of data heterogeneity is taken into consideration. The model error and the latency for HFL are first analyzed, and the optimization problem of user association and resource allocation is then proposed under both independent identically distributed (i.i.d.) and non-i.i.d. settings. For the non-i.i.d. settings, the distance of data distribution is considered and a primal-dual algorithm is proposed to solve the problem. Simulation results show that under both i.i.d. and non-i.i.d. settings, the proposed scheme can outperform the baselines in terms of latency and testing accuracy.

Other issues in HFL include aggregation interval and incentive mechanism design. In Ref. [20], a joint resource allocation and aggregation interval control problem is proposed, aiming to minimize the training loss and the latency. Convergence analysis is provided to show the dependency of the convergence performance on the number of participants, the aggregation interval and training latency. Then, the original problem is decomposed into two subproblems. The resource allocation problem is proved to be convex and the optimal value can be reached. For the aggregation interval control problem, a rounding and relaxation approach is adopted. Experimental results show that the proposed scheme can reach lower latency and higher training performance compared with the baselines. In Ref. [21], a two-level joint incentive design and resource allocation problem is proposed. At the lower level, the cluster selection problem is formulated as an evolutionary game. At the upper level, the action of the cluster head is solved via a deep learning-based approach. Experiments show the robustness and uniqueness of the proposed scheme.

3.3 New Considerations of HFL

The popularity of HFL gives rise to many novel architectures, such as HFL with device-to-device (D2D) communications. In Ref. [22], a multi-layer hybrid FL framework is proposed. The authors first introduce the architecture of this new FL architecture, where there are more than three layers. In each layer, clients aggregate the model parameters via D2D

▼Table 2. Summary of recent papers on HFL

Category	Highlight	Reference
Convergence analysis	Effect of edge and cloud aggregation intervals and local update step size with both convex and non-convex loss functions	Ref. [5]
	Extending Ref. [5] into HFL with quantization and carrying out the convergence analysis	Ref. [13]
	Effect of local iterations, edge epochs, global epochs, network topology and node heterogeneity on the convergence performance for a graph-based edge topology	Ref. [14]
	Extending Ref. [14] into HFL with data heterogeneity, random grouping and multi-layer architecture	Ref. [15]
	Mobility-aware HFL	Ref. [16]
Resource management	Joint resource allocation and edge association	Refs. [17 – 19]
	Joint resource allocation and interval control	Ref. [20]
	Joint resource allocation and incentive mechanism design	Ref. [21]
Other practical considerations	Multi-stage HFL with device-to-device communications	Ref. [22]

HFL: hierarchical federated learning

communications and then transmit the parameters to the upper layers. Convergence analysis is provided to derive an upper bound of this framework and a distributed control algorithm is proposed to improve the convergence performance. Experiment results show that the proposed framework can utilize the network resources more efficiently without loss of convergence speed and testing accuracy.

4 HFL in Vehicular Networks

There are many application scenarios that can benefit from the deployment of HFL in vehicular networks, such as autonomous driving, intelligent transportation systems and smart wireless communications. Recent studies on these scenarios have adopted FL as the training framework of AI models to obtain advantages in higher convergence speed, lower energy consumption and better privacy protection[23]. However, research on applying HFL to vehicular networks is still in its infancy, leaving a large room for further study.

In this section, we first introduce several typical use cases of ML in vehicular networks, showing the great potential of HFL. Then we analyze the challenges and opportunities of HFL caused by mobility in vehicular networks. Finally, we show our own work on the implementation of HFL in vehicular networks, taking into account the mobility aspect.

4.1 Typical Use Cases

1) Autonomous driving: Autonomous driving is one of the key technologies in future vehicular networks. Trajectory prediction and path planning are two necessary capabilities of autonomous driving vehicles. To avoid collision with pedestrians, vehicles and other traffic agents, autonomous driving vehicles must reliably predict the future trajectories of surrounding agents and safely and efficiently plan their own future driving paths[24]. Their decisions are based on the sensing data from onboard cameras, Lidars, GPS, and map information. To meet the stringent latency and precision requirements, ML algorithms have been applied for these two tasks[25 - 26], which perform better than traditional approaches. However, the traffic environments of vehicular networks vary all the time as they keep driving, which requires vehicles to continually update their ML models with the latest data generated by sensors. HFL is more promising to provide well-trained and up-to-date ML models over centralized ML or conventional FL, since HFL can utilize much more training data generated from a large number of vehicles driving in various areas, which can improve the adaptability of ML models to dynamic environments.

2) Intelligent transportation systems (ITS): ITS are novel traffic systems that utilize advanced information technologies to reduce traffic congestion, accident rate, energy consumption and carbon emissions, and thus enhance efficiency, safety, reliability and eco-friendliness[27]. Many typical applications of ITS are critical to future vehicular networks, such as collaborative perception and vehicle platooning. Collaborative perception, where data from multiple traffic agents are collected and fused to conduct object detection, can achieve higher accuracy and precision than single-vehicle perception[28]. Vehicle platooning, where a coordinated group of autonomous vehicles travels collectively, can achieve faster and safer autonomous driving with shorter spacing than single-vehicle traveling[29]. Existing research[28, 30] on these use cases also considers applying machine learning methods to achieve better performance. Note that the ML models for ITS tasks usually require vehicles to share data, and the data, such as photographs and videos, can be private and sensitive. However, the centralized ML needs to collect the raw data from all vehicles to train an ML model, which leads to heavy communication burdens, as well as privacy problems. To reduce the unnecessary raw data transmission and the resulting privacy leakage, HFL is a promising paradigm of model training in ITS, since it only collects the lightweight gradient data, rather than the heavyweight and private raw data.

3) Smart wireless communications: In smart wireless communications, ML algorithms are utilized in many wireless communication tasks, such as multiple-input, multiple-output (MIMO) beam selection[31], channel modeling and estimation[32], and joint source-channel coding[33]. Compared to traditional wireless communications, ML algorithms designed and exploited for smart wireless communications can decrease communication overhead, improve the signal-to-noise ratio (SNR), and save transmission power, with much lower latency and fewer computing resources. Similar to the use cases of autonomous driving, it is a challenge for ML models to adapt to the dynamic characteristics of channel states in vehicular networks. Therefore, HFL is also a promising training approach for smart wireless communications.

Although the use cases aforementioned have taken ML into account, there are few papers applying HFL to train the ML models for these scenarios in vehicular networks. Actually, HFL can exploit the data and computing resources of more vehicles, and thus train ML models more efficiently than centralized ML. Compared to FL, vehicles from larger areas can bring richer data features to the training of HFL, which improves the robustness of ML models. Therefore, it is promising to further study the application of HFL in vehicular networks.

4.2 Challenges and Opportunities with Vehicle Mobility

Despite the promising potential of applying HFL to vehicular networks, some properties of vehicular networks may stand as great barriers, in particular the mobility. Unlike other FL scenarios where clients stay in the same place or move at a low speed, intelligent vehicles usually travel fast on road, especially when they drive on the highway. This brings more dynamics and uncertainties to the topology of vehicles, leading to a change of association between vehicles and edges. First, vehicles may leave the coverage of an edge when uploading its

model parameters while transmitting model parameters, or even before finishing one round of local updates, leading to a waste of communication and computation resources as well as leakage of training data. Second, the varying channel conditions of vehicular communication links and the Doppler effect caused by vehicle mobility may result in the failure of model transmission or transmission errors in the received parameters, which also influences the FL training performance.

However, there are also chances brought by mobility. On the one hand, the mobility of vehicles creates more opportunities to meet[6] other vehicles, inspiring the leverage of vehicle-to-vehicle (V2V) communications through side links to compensate for the loss of changing edge and also accelerate the speed of edge aggregation. On the other hand, since the hierarchical structure of HFL brings a wide coverage, even though vehicles step out of the coverage of an edge, there's a great chance that they still stay in the range of the cloud, so their data can still be used by training. What's more, due to the heterogeneity of clients and the dynamic nature of the road environment, data distribution generally varies from one edge to another. Mobility of vehicles promotes data fusion of edges and thus reduces data heterogeneity, which helps the global training model to converge faster. In the following section, we will give two case studies as examples of leveraging these opportunities.

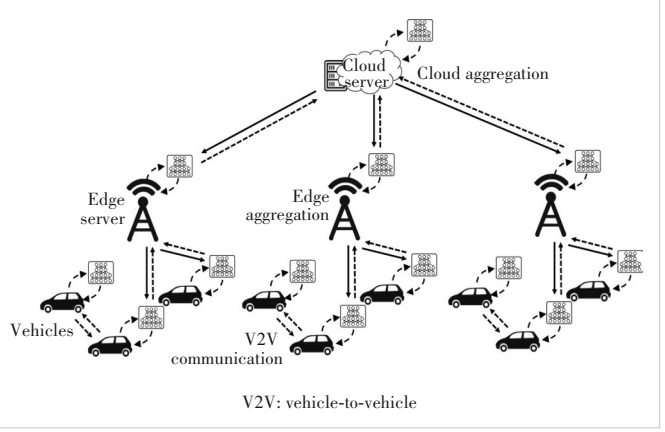
4.3 Case Study 1: V2V-Assisted Hierarchical Federated Learning

In this case study, we propose a V2V-assisted hierarchical federated learning (VAHFL) framework, where the V2V communication is utilized to speed up the aggregation process. In this framework (Fig. 3), the uploading of model parameters includes both vehicle-to-infrastructure (V2I) and V2V communication. Some vehicles act as relay nodes that help other vehicles with parameter transmission. Vehicles leaving the coverage of the central server can transmit their model parameters to the nearby relay nodes via the V2V link before it leaves, while vehicles near the server directly transmit its parameter to the server via the V2I link. We formulate a communication latency minimization problem by optimizing the uploading strategy, and a graph neuron network-reinforcement learning (GNN-RL) based algorithm is designed to solve this problem.

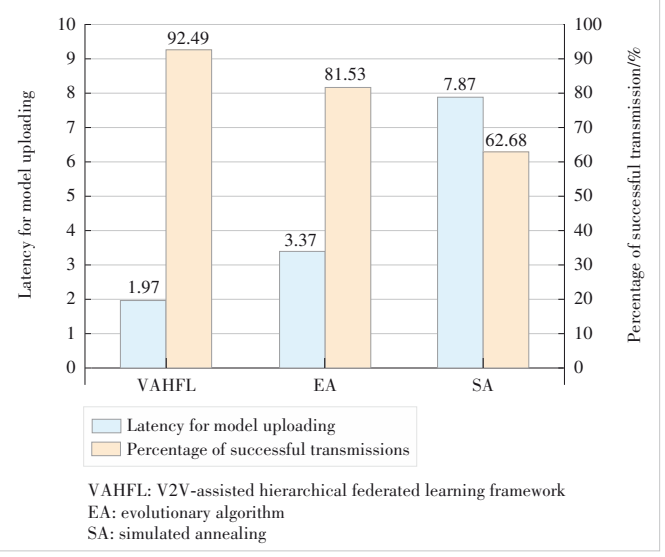
An experimental platform is built based on Simulation of Urban Mobility (SUMO) to evaluate the proposed framework, where there is one cloud server, four edge servers and 200 vehicles. The vehicles move over time according to the Manhattan mobility model. The vehicles cooperatively train a convolutional neural network (CNN) model for an image classification task using the CIFAR-10[34] dataset. The V2I bandwidth is set to 30 MHz, and the V2V bandwidth is set to 10 MHz. For the benchmark, we consider that the vehicles directly transmit their model parameters to the server. Fig. 4 illustrates that the proposed framework can reduce transmission latency by 41.54% and increase the percentage of successful transmissions by 10.97%.

4.4 Case Study 2: Edge-Heterogeneous Hierarchical Federated Learning

In this case study, we investigate the influence of mobility when training data of edge servers are heterogeneous. Before training, vehicles sample data to form local datasets. Data distribution is dependent on the location of vehicles, which means vehicles under the coverage of the same edge server sample from the same distribution, while vehicles from the coverage of different edge servers sample differently. Therefore, at the start of training, the data distribution of edges is heterogeneous. During training, vehicles constantly travel across edges, driving the data from different edge servers to mix up. We analyze the convergence speed of this edge-heterogeneous HFL system and prove that mobility accelerates convergence by promoting data fusion.



▲ **Figure 3. Schematic of V2V-assisted hierarchical federated learning framework**



▲ **Figure 4. Latency and the percentage of successful transmission of the proposed scheme and baseline**

Experiments are also conducted based on SUMO. We assume one cloud server, four edge servers and 32 vehicles cooperatively train a four-layer CNN on the CIFAR-10 dataset, and we only choose data of eight classes from 10 classes for training and inference. Initially, each edge has data of two classes, which is uniformly distributed in vehicles under the coverage of the edge. During training, vehicles travel by the Manhattan mobility model, with their local datasets unchanged, which leads to changes in edge data distribution. The network is trained on three settings of vehicle mobility: no mobility, low mobility and high mobility. As Fig. 5 shows, mobility increases the convergence speed and final test accuracy of HFL. What's more, when vehicles are moving, a higher vehicle speed results in a faster convergence speed. As is shown by the dashed line and stars in the figure, if we set the target test accuracy as 0.75, the low mobility and high mobility scenario reduces the training epochs by 40.6% and 51.9% separately.

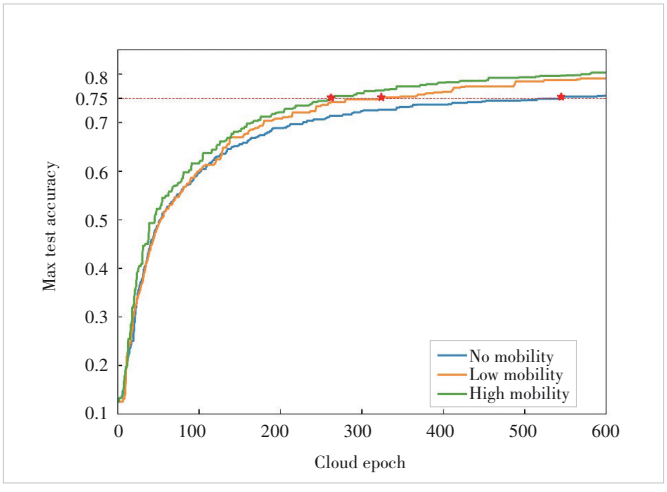
5 Conclusions

This paper presents an overview of HFL and its application in vehicular networks. First, we introduce the background and motivation of HFL and the possibility of implementing it in vehicular networks. Then, the architecture of HFL is presented. Afterward, we discuss new issues and challenges of HFL compared with FL and review existing solutions. Furthermore, some typical use cases in vehicular networks are introduced and our existing works of implementing HFL in vehicular networks are presented. Apart from the works mentioned above, there are still some challenges and research directions for HFL and its implementation in vehicular networks:

1) Heterogenous vehicular networks: For HFL in vehicular networks, the participants may be more than just vehicles. Mobile devices and other transportation infrastructures can also participate in model aggregation. In such a case, the network is heterogeneous, i.e., the computing capability, the communication capacity and the mobility patterns of clients in this network are quite different. This brings challenges to FL system design and resource management strategy.

2) Variation of channel conditions: Due to the high mobility of vehicles, the channel conditions of vehicular communication links may vary rapidly. This may result in the failure of model transmission or transmission errors in the received parameters. Therefore, the communication system should be carefully designed to prevent such cases.

3) Exploration of benefits of mobility: Usually, mobility is considered a bottleneck for FL implementation and training. However, mobility may also be explored to enhance FL training performance. In our initial efforts, the convergence speed of an edge-heterogeneous HFL is shown to be enhanced by the data fusion brought by vehicle mobility. Apart from that, other benefits of utilizing vehicle mobility are also worth being explored.



▲Figure 5. Maximum achievable test accuracy of cloud model with different mobility

References

[1] CHEN M Z, GÜNDÜZ D, HUANG K B, et al. Distributed learning in wireless networks: recent progress and future challenges [J]. IEEE journal on selected areas in communications, 2021, 39(12): 3579 – 3605. DOI: 10.1109/JSAC.2021.3118346
[2] SUN Y X, SHI W Q, HUANG X F, et al. Edge learning with timeliness constraints: challenges and solutions [J]. IEEE communications magazine, 2020, 58(12): 27 – 33. DOI: 10.1109/MCOM.001.2000382
[3] SHI Y M, YANG K, JIANG T, et al. Communication-efficient edge AI: algorithms and systems [J]. IEEE communications surveys & tutorials, 2020, 22(4): 2167 – 2191. DOI: 10.1109/comst.2020.3007787
[4] ABAD M S H, OZFATURA E, GUNDUZ D, et al. Hierarchical federated learning across heterogeneous cellular networks [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 8866 – 8870. DOI: 10.1109/ICASSP40776.2020.9054634
[5] LIU L M, ZHANG J, SONG S H, et al. Client-edge-cloud hierarchical federated learning [C]//IEEE International Conference on Communications (ICC). IEEE, 2020: 1 – 6. DOI: 10.1109/ICC40277.2020.9148862
[6] SUN Y X, XIE B W, ZHOU S, et al. MEET: mobility-enhanced edge intelligence for smart and green 6G networks [J]. IEEE communications magazine, 2023, 61(1): 64 – 70. DOI: 10.1109/MCOM.001.2200252
[7] SAMARAKOON S, BENNIS M, SAAD W, et al. Distributed federated learning for ultra-reliable low-latency vehicular communications [J]. IEEE transactions on communications, 2020, 68(2): 1146 – 1159. DOI: 10.1109/TCOMM.2019.2956472
[8] YU Z X, HU J, MIN G Y, et al. Mobility-aware proactive edge caching for connected vehicles using federated learning [J]. IEEE transactions on intelligent transportation systems, 2021, 22(8): 5341 – 5351. DOI: 10.1109/TITS.2020.3017474
[9] XIAO H Z, ZHAO J, PEI Q Q, et al. Vehicle selection and resource optimization for federated learning in vehicular edge computing [J]. IEEE transactions on intelligent transportation systems, 2022, 23(8): 11073 – 11087. DOI: 10.1109/TITS.2021.3099597
[10] WANG S Y, LIU F F, XIA H L. Content-based vehicle selection and resource allocation for federated learning in IoV [C]//IEEE Wireless Communications and Networking Conference Workshops (WCNCW). IEEE, 2021: 1 – 7. DOI: 10.1109/WCNCW49093.2021.9419986
[11] LIM W Y B, LUONG N C, HOANG D T, et al. Federated learning in mobile edge networks: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2020, 22(3): 2031 – 2063. DOI: 10.1109/COMST.2020.2986024
[12] ELBIR A M, SONER B, ÇÖLERI S, et al. Federated learning in vehicular networks [C]//IEEE International Mediterranean Conference on Communications

and Networking (MeditCom). IEEE, 2022: 72 – 77. DOI: 10.1109/MeditCom55741.2022.9928621
[13] LIU L M, ZHANG J, SONG S H, et al. Hierarchical federated learning with quantization: convergence analysis and system design [J]. IEEE transactions on wireless communications, 2023, 22(1): 2 – 18. DOI: 10.1109/TWC.2022.3190512
[14] CASTIGLIA T, DAS A, PATTERSON S. Multi-level local SGD: distributed SGD for heterogeneous hierarchical networks [C/OL]. International Conference on Learning Representations, 2021 [2021-05-03]. https://openreview.net/pdf?id=C70cp4Cn32
[15] WANG J Y, WANG S Q, CHEN R R, et al. Demystifying why local aggregation helps: convergence analysis of hierarchical SGD [J]. Proceedings of the AAAI conference on artificial intelligence, 2022, 36(8): 8548 – 8556. DOI: 10.1609/aaai.v36i8.20832
[16] FENG C, YANG H H, HU D, et al. Mobility-aware cluster federated learning in hierarchical wireless networks [J]. IEEE transactions on wireless communications, 2022, 21(10): 8441 – 8458. DOI: 10.1109/TWC.2022.3166386
[17] LUO S Q, CHEN X, WU Q, et al. HFEL: joint edge association and resource allocation for cost-efficient hierarchical federated edge learning [J]. IEEE transactions on wireless communications, 2020, 19(10): 6535 – 6548. DOI: 10.1109/TWC.2020.3003744
[18] LIM W Y B, NG J S, XIONG Z H, et al. Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks [J]. IEEE journal on selected areas in communications, 2021, 39(12): 3640 – 3653. DOI: 10.1109/JSAC.2021.3118401
[19] LIU S L, YU G D, CHEN X F, et al. Joint user association and resource allocation for wireless hierarchical federated learning with non-IID data [C]//IEEE International Conference on Communications. IEEE, 2022: 74 – 79. DOI: 10.1109/ICC45855.2022.9839164
[20] XU B, XIA W C, WEN W L, et al. Adaptive hierarchical federated learning over wireless networks [J]. IEEE transactions on vehicular technology, 2022, 71(2): 2070 – 2083. DOI: 10.1109/tvt.2021.3135541
[21] LIM W Y B, NG J S, XIONG Z H, et al. Decentralized edge intelligence: a dynamic resource allocation framework for hierarchical federated learning [J]. IEEE transactions on parallel and distributed systems, 2022, 33(3): 536 – 550. DOI: 10.1109/TPDS.2021.3096076
[22] HOSSEINALIPOUR S, AZAM S S, BRINTON C G, et al. Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks [J]. IEEE/ACM transactions on networking, 2022, 30(4): 1569 – 1584. DOI: 10.1109/TNET.2022.3143495
[23] DU Z Y, WU C, YOSHINAGA T, et al. Federated learning for vehicular Internet of Things: recent advances and open issues [J]. IEEE open journal of the computer society, 2020, 1: 45 – 61. DOI: 10.1109/OJCS.2020.2992630
[24] MA Y X, ZHU X G, ZHANG S B, et al. TrafficPredict: trajectory prediction for heterogeneous traffic-agents [J]. Proceedings of the AAAI conference on artificial intelligence. AAAI, 2019, 33(1): 6120 – 6127. DOI: 10.1609/aaai.v33i01.33016120
[25] ALTCHÉ F, DE LA FORTELLE A. An LSTM network for highway trajectory prediction [C]//IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018: 353 – 359. DOI: 10.1109/ITSC.2017.8317913
[26] SHALEV-SHWARTZ S, SHAMMAH S, SHASHUA A. Safe, multi-agent, reinforcement learning for autonomous driving [EB/OL]. (2016-10-11)[2022-05-01]. https://arxiv.org/abs/1610.03295
[27] QURESHI K N, ABDULLAH A H. A survey on intelligent transportation systems [J]. Middle-east journal of scientific research, 2013, 15(5): 629 – 642. DOI: 10.5829/idosi.mejsr.2013.15.5.11215
[28] MAO R, GUO J, JIA Y, et al. DOLPHINS: dataset for collaborative perception enabled harmonious and interconnected self-driving [EB/OL]. (2022-07-15)[2022-08-01]. https://arxiv.org/abs/2207.07609
[29] AXELSSON J. Safety in vehicle platooning: a systematic literature review [J]. IEEE transactions on intelligent transportation systems, 2017, 18(5): 1033 – 1045. DOI: 10.1109/TITS.2016.2598873
[30] PRATHIBA S B, RAJA G, DEV K, et al. A hybrid deep reinforcement learning for autonomous vehicles smart-platooning [J]. IEEE transactions on vehicular technology, 2021, 70(12): 13340 – 13350. DOI: 10.1109/TVT.2021.3122257
[31] KLAUTAU A, BATISTA P, GONZÁLEZ-PRELCIC N, et al. 5G MIMO data for machine learning: application to beam-selection using deep learning [C]// Information Theory and Applications Workshop (ITA). IEEE, 2018: 1 – 9. DOI: 10.1109/ITA.2018.8503086
[32] ALDOSSARI S M, CHEN K C. Machine learning for wireless communication channel modeling: an overview [J]. Wireless personal communications, 2019, 106(1): 41 – 70. DOI: 10.1007/s11277-019-06275-4
[33] KURKA D B, GÜNDÜZ D. Bandwidth-agile image transmission with deep joint source-channel coding [J]. IEEE transactions on wireless communications, 2021, 20(12): 8081 – 8095. DOI: 10.1109/TWC.2021.3090048
[34] KRIZHEVSKY A. Learning multiple layers of features from tiny images [D]. Toronto: University of Toronto, 2009

Biographies

YAN Jintao is a PhD student at Tsinghua University, China. His research interests include federated learning and vehicular edge computing and vehicular networks.

CHEN Tan is a PhD student at Tsinghua University, China. His research interests include federated learning and vehicular networks.

XIE Bowen is a PhD student at Tsinghua University, China. His research interests include federated learning and vehicular networks.

SUN Yuxuan is an associate professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, China and was previously a postdoctoral researcher with Tsinghua University, China. Her research interests include edge computing and edge learning.

ZHOU Sheng (sheng.zhou@tsinghua.edu.cn) is an associate professor with the Department of Electronic Engineering, Tsinghua University, China. His research interests include vehicular networks, mobile edge computing, and green wireless communications.

NIU Zhisheng is a professor with the Department of Electronic Engineering, Tsinghua University, China. His major research interests include queueing theory, traffic engineering, radio resource management of wireless networks, and green communication and networks.

Secure Federated Learning over Wireless Communication Networks with Model Compression

DING Yahao[1], Mohammad SHIKH-BAHAEI[1],
YANG Zhaohui[2], HUANG Chongwen[2], YUAN Weijie[3]

(1. King's College London, London WC2R 2LS, U.K.;
2. Zhejiang University, Hangzhou 310058, China;
3. Southern University of Science and Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202301006

https://kns.cnki.net/kcms/detail/34.1294.TN.20230314.1752.002.html, published online March 16, 2023

Manuscript received: 2023-02-11

Abstract: Although federated learning (FL) has become very popular recently, it is vulnerable to gradient leakage attacks. Recent studies have shown that attackers can reconstruct clients' private data from shared models or gradients. Many existing works focus on adding privacy protection mechanisms to prevent user privacy leakages, such as differential privacy (DP) and homomorphic encryption. These defenses may cause an increase in computation and communication costs or degrade the performance of FL. Besides, they do not consider the impact of wireless network resources on the FL training process. Herein, we propose weight compression, a defense method to prevent gradient leakage attacks for FL over wireless networks. The gradient compression matrix is determined by the user's location and channel conditions. We also add Gaussian noise to the compressed gradients to strengthen the defense. This joint learning of wireless resource allocation and weight compression matrix is formulated as an optimization problem with the objective of minimizing the FL loss function. To find the solution, we first analyze the convergence rate of FL and quantify the effect of the weight matrix on FL convergence. Then, we seek the optimal resource block (RB) allocation by exhaustive search or ant colony optimization (ACO) and then use the CVX toolbox to obtain the optimal weight matrix to minimize the optimization function. The simulation results show that the optimized RB can accelerate the convergence of FL.

Keywords: federated learning (FL); data leakage from gradient; resource block (RB) allocation

Citation (IEEE Format): Y. H. Ding, M. Shikh-Bahaei, Z. H. Yang, et al., "Secure federated learning over wireless communication networks with model compression," *ZTE Communications*, vol. 21, no. 1, pp. 46 – 54, Mar. 2022. doi: 10.12142/ZTECOM.202301006.

1 Introduction

Federated learning (FL) [1], an emerging distributed learning algorithm, has received much attention in recent years due to its data protection property[2]. This algorithm has been extensively employed in applications where preserving user privacy is of utmost importance, such as in the case of hospital data[3]. FL allows clients to utilize private sensitive data to collaboratively train a machine learning model locally without explicitly sharing individual sensitive data. In the context of wireless networks with limited bandwidth and latency requirements, the advantages of FL are even more pronounced, especially when there are a large number of users and data. This is because only models or gradients are transmitted, which not only enhances the privacy of the data but also significantly improves communication efficiency.

Although FL offers default data privacy by avoiding the exchange of raw data between participants and a server, recent studies have noted that FL faces various attacks such as membership inference attacks[4], generative adversarial network attacks[5 – 6], gradient leakage attacks[7 – 10], model inversion attacks[11], model poisoning, data poisoning and free-riding attack during the training process[12]. These attacks will expose users' private data, such as the location of confidential sites, and the condition of patients, or corrupt the global model and affect the performance of the model. One of the most advanced privacy leakage techniques is gradient leakage, where an honest-but-curious server could illegally reconstruct the user's privacy data by performing gradient leakage attacks on the client's uploaded model weights or gradients. Furthermore, even if the federated server is reliable, gradient leakage can occur by eavesdroppers near the clients or server in the wireless network. Therefore, tackling the gradient leakage issue is essential for promoting FL in practical applications, such as edge computing and UAV swarms.

The related work is as follows.

1) Gradient leakage attacks: Gradient leakage attacks are used to reconstruct training input data (e.g., images or text) and labels through shared gradients or weights. The work in Ref. [7] first discussed the recovery of image data from gradients in neural networks and demonstrated the feasibility of reconstructing data from a single neuron or linear layer net-

works. In Ref. [6], a single image was reconstructed from a 4-layer CNN comprising a significantly large fully-connected layer. ZHU et al. in Ref. [8] proposed the deep leakage from gradient (DLG) algorithm. In particular, it yields dummy gradients by randomly generating dummy data and dummy labels, then minimizing the difference between the dummy gradient and the original gradient, which in turn makes the dummy input close to the original input, and finally recovering the original data. They successfully reconstructed training data and ground-truth labels from a 4-layer CNN. Moreover, they demonstrated that it is indeed possible to recover multiple images from their averaged gradients (maximum batch size of 8). Following up Ref. [8], due to the difficulties of DLG in convergence performance and extracting ground truth labels consistently, the improved deep leakage from gradient (iDLG) algorithm was proposed in 2020[9] as a simple and effective method to recover the original data and discover ground truth labels. GEIPING et al.[13] studied the reconstruction of multiple images from their averaged gradients, where they used cosine similarity as a cost function and optimized the sign of the gradient. The simulations show that it only reconstructs single images from gradients. Furthermore, the work in Ref. [10] introduced a GradInversion method to recover training image batches by inverting averaged gradients.

2) Defense methods for privacy leakage: Recently, a number of studies have focused on defense strategies for privacy leakage in FL. These methods can be categorized into four types: homomorphic encryption[7, 14 – 15], multi-party computation[16 – 17], differential privacy (DP)[18 – 20], and gradient compression. Homorphic encryption and multi-party computation incur a significant extra computational cost, thus it is not suitable for wireless network scenarios with limited communication resources and delay requirements. For the DP method, it is to add Gaussian noise or Laplacian noise to the gradient before transmission, which can mitigate privacy leakage, but it also negatively affects the training process and model performance[21]. Gradient compression defends against data leakage by pruning gradients with small magnitudes to zero so that eavesdroppers cannot match the original gradients. The work in Ref. [8] demonstrated that it is not possible to prevent leakage when the sparsity is less than 10%, but when the compression rate is more than 20%, the recovered image is no longer recognizable, and the leakage is successfully prevented. However, excessive compression may affect the model's performance. Overall, these defense approaches achieve adequate defense either by incurring significant overhead or by compromising the accuracy of the model and they are not specifically designed to defend against data leakage on a gradient[22]. Unlike the general-purpose protection mentioned above, the studies in Refs. [22 – 24] focus on defending against gradient leakage attacks. SUN et al. in Ref. [22] observed that the class-wise data presentations of each client's data are embedded in shared local model updates, which is why privacy can be inferred from the gradient, and the proposed Soteria could effectively protect training data via perturbing data presentation in an FC layer. In PRECODE[23], variational modeling is used to disguise the original latent feature space susceptible to privacy leakage by DLG attacks. Moreover, WANG et al.[24] proposed a lightweight defense mechanism against data leakage from gradients. They used the sensitivity of gradient changes w.r.t. the input data to quantify the leakage risk and perturb gradients according to leakage risk. In addition, global correlations of gradients are applied to compensate for this perturbation. These three methods provide a significant defense against DLG attacks and have little effect on model performance. However, one essential part, wireless network resources (e.g., bandwidth and power), are not considered in these defense frameworks.

Although the aforementioned methods (Soteria, PRE CODE, and a lightweight defense mechanism) have been successful in defending against DLG attacks, all the proposed defense methods focus solely on the theoretical process of FL training and only the server or participants are considered malicious attackers. To the best of our knowledge, there is a lack of research on defending against DLG attacks for FL in wireless networks. The fact is that the convergence and performance of FL may be affected by bandwidth, noise, delay, power, etc. in dynamic wireless networks. Therefore, to fill in the blank, we propose a novel defensive mechanism, weight compression for gradients, to protect data privacy from DLG attacks in FL. Moreover, we consider external eavesdroppers, such as users around the clients or servers who are not involved in FL training. Key contributions of this work include:

• We propose a novel defensive framework, weight compression, for protecting the data privacy of FL over wireless networks by considering FL and wireless metrics and factors. This defense is implemented by compressing the local gradient by taking into account the user's location and channel quality. In addition, Gaussian artificial noise is added to the compressed gradients for further defense.

• We formulate this joint resource allocation and weight compression matrix for FL as an optimization problem with the goal of minimizing the training loss while satisfying the delay and leakage requirement. Thus, our defensive mechanism jointly considers learning and wireless network metrics.

The rest of this paper is organized as follows. The system model and problem formulation are analyzed in Section 2. The analysis of the FL convergence rate is presented in Section 3. In Section 4, the joint optimization problem is simplified and solved. Then, the simulation result and analysis are described in Section 5. Finally, conclusions are summarized in Section 6.

2 System Model and Problem Formulation

In this paper, we consider a small network consisting of one server and a set of N clients to jointly train an FL model for task inference in a wireless environment, which includes an

eavesdropper, as shown in Fig. 1.

2.1 Federated Learning Model

In the FL model, the training data as input of the FL algorithm collected by each client i is denoted as $\boldsymbol{X}_i = \left[\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{iK_i}\right]$, where K_i is the number of samples collected by client i and each element $\boldsymbol{x}_{ik}$ denotes the K-th sample of client i. The matrix $\boldsymbol{y}_i = \left[y_{i1}, \cdots, y_{iK_i}\right]$ is the corresponding labels of training data $\boldsymbol{X}_i$. After collecting data, each client i trains its local model using $\left(\boldsymbol{X}_i, \boldsymbol{y}_i\right)$ and the server aggregates received local models to update the global model for the next round of training. The main objective of the FL training process is to find optimal model parameters $\boldsymbol{w}^*$ that minimize the global loss function and the training process can be considered as solving an optimization problem, defined as:

$$\min_{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_N} \frac{1}{K} \sum_{i=1}^{N} \sum_{k=1}^{K_i} f\left(\boldsymbol{w}_i, \boldsymbol{x}_{ik}, y_{ik}\right), \tag{1}$$

where $K = \sum_{i=1}^{N} K_i$ is the total size of the training data of all clients; $\boldsymbol{w}_i$ is a vector that represents the local model of each client i; $f\left(\boldsymbol{w}_i, \boldsymbol{x}_{ik}, y_{ik}\right)$ is the loss function of the i-th client with one data sample. $F_i\left(\boldsymbol{w}_i, \boldsymbol{x}_{i1}, y_{i1}, \cdots, \boldsymbol{x}_{iK_i}, y_{iK_i}\right)$ is the total loss function of the i-th client with the whole data sample, which is abbreviated as $F_i\left(\boldsymbol{w}_i\right)$. Moreover, the expression of $f(\cdot)$ is application-specific.

In general, Eq. (1) could be solved by performing gradient descent in each client periodically. The detailed training process consists of the following three steps:

1) Training initialization: The server first initiates a global model $\boldsymbol{w}^0$ and sets up hyperparameters of training processes, e. g., the number of epochs and learning rate. The initialized global model $\boldsymbol{w}^0$ is broadcast to clients in the first round. The clients start local model training after receiving $\boldsymbol{w}^0$.

2) Local training and updating: At each step j, after receiving the global weight $\boldsymbol{w}^j$ from the server, each client i samples a batch from their own dataset to compute the updated local gradients $\boldsymbol{g}_i^j$.

$$\boldsymbol{g}_i^j = \frac{1}{B} \sum_{k \in K_i^{j+1}} \frac{\partial f\left(\boldsymbol{w}^j, \boldsymbol{x}_{ik}, y_{ik}\right)}{\partial \boldsymbol{w}^j}, \tag{2}$$

where K_i^j is a randomly selected subset of B training data samples from user i's training dataset K_i at the j-th training round.

3) Model aggregation and download: Once the server receives all local gradients from N clients, it combines them to update the global gradients $\boldsymbol{g}_g^j$. Then, the weights $\boldsymbol{w}^{j+1}$ are updated and sent back to the clients for the next training round. The update of the global gradient vector and weights is given by[25]:

$$\boldsymbol{g}_g^j = \frac{1}{K} \sum_{i=1}^{N} K_i \boldsymbol{g}_i^j, \tag{3}$$

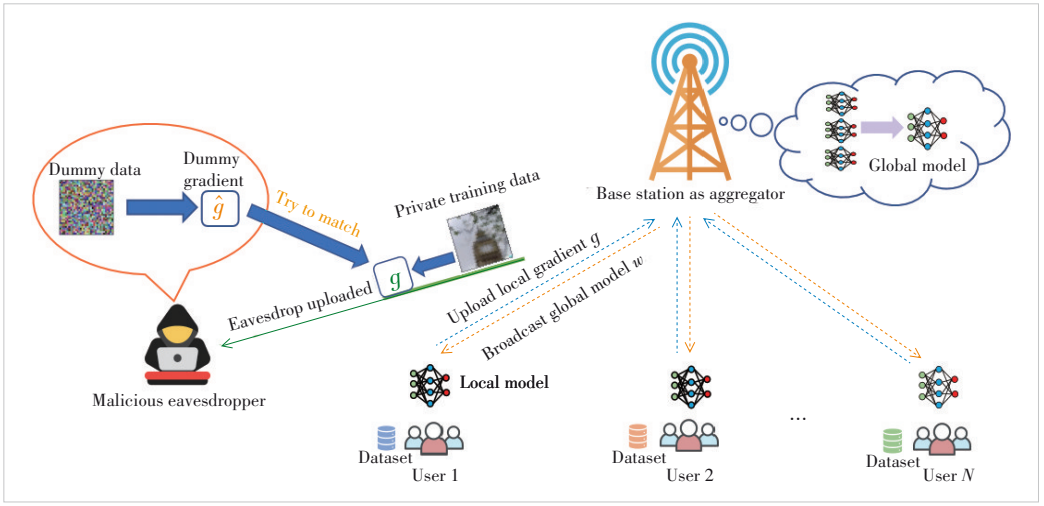
$$\boldsymbol{w}^{j+1} = \boldsymbol{w}^j - \eta \boldsymbol{g}_g^j, \tag{4}$$

where η is the learning rate. Finally, processes 2 and 3 are iterated until the global loss function converges or achieves the desired accuracy.

2.2 Threat Model

In this work, we consider the DLG attack[8] performed by the eavesdropper on the uplink and downlink to recover the original private data from the client. The DLG attack is conducted by making the gap between the generated dummy gradient and the eavesdropped local FL gradient smaller and smaller through multiple iterations, so that the corresponding dummy data become more and more similar to the original data.

We assume that the eavesdropper taps only one nearby client i at a time, eavesdropping on the last updated local gradient $(\boldsymbol{g}_i^J)$ of the uplink transmission and the weight $(\boldsymbol{w}^J)$ from the downlink, where J is the number of iterations for FL to reach convergence. After that, the eavesdropper randomly generates a set of dummy inputs $\hat{\boldsymbol{x}} = \left[\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_B\right]$ and $\hat{\boldsymbol{y}} = \left[\hat{y}_1, \cdots, \hat{x}_B\right]$, which are initialized as random noise and optimized toward the ground truth data $\boldsymbol{x}^*$. These dummy data and labels are updated by the difference between the dummy gradient and the original gradi-



▲Figure 1. Architecture of FL algorithm with one eavesdropper in wireless networks

ent in each loop. Finally, the privacy data are recovered by minimizing the following objective[10, 26].

$$\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{y}}^* = \arg\min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}} \left\| \hat{\boldsymbol{g}} - \boldsymbol{g}_i^J \right\|_2, \tag{5}$$

$$\hat{\boldsymbol{g}} = \frac{1}{B} \sum_{b=1}^{B} \frac{\partial f\left(\boldsymbol{w}^J, \hat{\boldsymbol{x}}_b, \hat{y}_b\right)}{\partial \boldsymbol{w}^J}, \tag{6}$$

where $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are the synthetic dummy data and labels, respectively; $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are the ground truth data and labels corresponding to the eavesdropped gradient $\boldsymbol{g}_i^J$; $\hat{\boldsymbol{x}}^*$ and $\hat{\boldsymbol{y}}^*$ are the recovered data and labels. If $B = 1$, Eq. (5) can be expressed as

$$\hat{\boldsymbol{x}}^*, \hat{y}^* = \arg\min_{\hat{\boldsymbol{x}}, \hat{y}} \left\| \frac{\partial f\left(\boldsymbol{w}^J, \hat{\boldsymbol{x}}, \hat{y}\right)}{\partial \boldsymbol{w}^J} - \boldsymbol{g}_i^J \right\|_2. \tag{7}$$

2.3 Defense Method

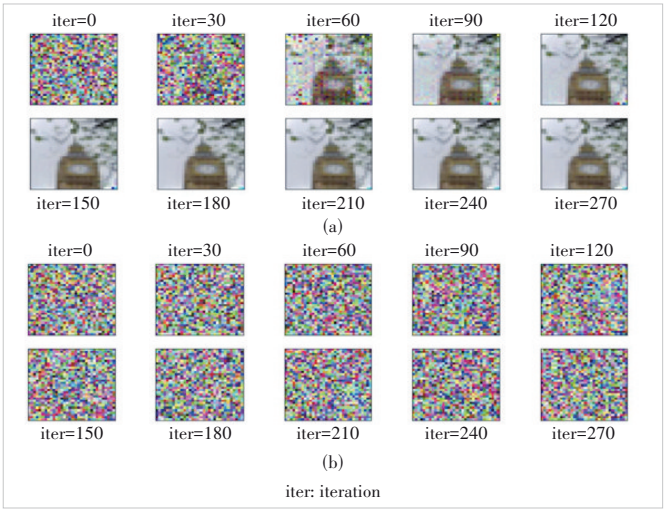
Data leakage is mainly caused by the leakage of the gradient transmitted in the wireless network. Therefore, it can be considered to compress or encrypt the gradient on the client side to make it difficult for eavesdroppers to recover private data. In this section, we propose a defense method against data leakage called weight compression. The weight compression scheme belongs to gradient compression, which is based on the user's location and channel quality to determine the compression matrix. Local gradients are divided into several parts by the compression matrix and only some of the gradients are sent to the server at a time for aggregation. Moreover, we add Gaussian noise to compressed gradients as the second defense strategy to strengthen the defense. Fig. 2 shows the result of applying DP to defend against DLG attacks. Fig. 2(a) illustrates that DLG can recover the original image easily without adding any defense methods and Fig. 2(b) demonstrates its effectiveness with the addition of the Gaussian noise defense approach.

We define u_i^j as the weight matrix of client i at the j-th iteration. To further prevent privacy data leakage, we add artificial Gaussian noise to the compressed gradient, and then the selected partial local gradient is given as:

$$\tilde{\boldsymbol{g}}_i^j = \boldsymbol{g}_i^j \odot \boldsymbol{u}_i^j + \boldsymbol{n}_i^j, \tag{8}$$

where $\boldsymbol{g}_i^j = \left[g_{i,1}^j, \cdots, g_{i,M}^j\right]$ and $\boldsymbol{u}_i^j = \left[u_{i,1}^j, \cdots, u_{i,M}^j\right]$, M refers to the number of gradients, and $\odot$ is the dot product. In Eq. (8), the first part $\boldsymbol{g}_i^j \odot \boldsymbol{u}_i^j$ represents the selected partial gradient, and the second part represents the addition of Gaussian noise, where $n \sim N\left(0, \sigma^2\right)$. An example is shown in Fig. 3. Moreover, the compression ratio is controlled by α, i. e., $\sum_{m=1}^{M} u_{i,m}^j \leqslant \alpha_i M, u_{i,m}^j \in \{0, 1\}$.

In this work, we define Eq. (9) to restrict the leakage of gradients[27].



▲Figure 2. Illustration of the differential privacy (DP) method to protect the privacy of federated learning (FL)

User 1		User 2		User 3	
Original gradients g_1	Transmitted gradients $\tilde{g}_1$	Original gradients g_2	Transmitted gradients $\tilde{g}_2$	Original gradients g_3	Transmitted gradients $\tilde{g}_3$
1	0	0.8	0.8	−2	0
1	1	0.3	0	1.5	0
−1	0	0.7	0.7	−1	0
1	0	2	0	3	3
−1	−1	−1	0	−0.4	0
−1	0	−0.5	0	2	2

Selected partial gradients

$g_i \odot u_i + n_i = \tilde{g}_i$

▲Figure 3. An example of proposed weight compression

$$\sum_{m=1}^{M} \rho_{i,m} u_{i,m} \leqslant DP_0, \tag{9}$$

where $\rho_{i,m} = 1/(K_i \sigma^2)$ stands for the data leakage level of each gradient and DP_0 denotes the maximum amount of gradient leakage.

2.4 Transmission Model

In the FL training process, all clients upload their local FL gradient to the BS via orthogonal frequency domain multiple access (OFDMA). For the uplink, the upper bound of the transmission rate of client i can be given by:

$$r_i^U = b_i B_0 \log_2\left(1 + \frac{P_i h_i}{N_0 B_0}\right), \tag{10}$$

where $b_i = \sum_{q=1}^{Q} b_{i,q}$ is the number of RBs allocated to client i. Note that we assume that all clients participate in the FL training, so $b_i \geqslant 1$. Q is the total number of RBs, B_0 is the bandwidth of each RB, and $\sum_{i=1}^{N} b_i B_0 \leqslant B$, where B is the total

bandwidth. P_i is the transmit power of client i, h_i is the channel gain between client i and the BS. N_0 is the Gaussian noise power spectral density.

According to the data rate of the uplink in Eq. (10), the transmission delay between client i and the BS on the uplink can be expressed by:

$$t_i^U = \frac{Z(\tilde{\boldsymbol{g}}_i)}{r_i^U}, \tag{11}$$

where the function $Z(\tilde{\boldsymbol{g}}_i)$ denotes the size of the data transmitted by each client i to the BS, i.e., the number of bits corresponding to the selected local gradients. We set $Z(\tilde{\boldsymbol{g}}_i) = C\sum_m^M u_{i,m} + 1\sum_m^M(1 - u_{i,m})$, where C denotes the number of bits per selected gradient.

2.5 Problem Formulation

In order to prevent eavesdroppers from recovering the private data of clients and to guarantee FL model convergence, we propose a defense method called weight compression to compress the transmission gradient and formulate an optimization problem to implement this joint-designed defense method and the FL algorithm. The objective is to minimize data leakage with limited iterations or delays by optimizing the portion selection of the local FL gradient for transmission. The optimization function is defined by

$$\min_{\boldsymbol{u},\boldsymbol{b}} \frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K_i} f(\boldsymbol{w}_i, \boldsymbol{x}_{ik}, y_{ik}), \tag{12}$$

$$\text{s.t. } b_i = \sum_{q=1}^{Q} b_{i,q} \geqslant 1, \forall i \in N, \tag{12a}$$

$$\sum_{i=1}^{N} b_i B_0 \leqslant B, \forall i \in N, \tag{12b}$$

$$u_{i,m} \in \{0,1\}, \forall i \in N, \tag{12c}$$

$$\sum_{m=1}^{M} u_{i,m} \leqslant \alpha_i M, \forall i \in N, \tag{12d}$$

$$\sum_{m=1}^{M} \rho_{i,m} u_{i,m} \leqslant DP_0, \forall i \in N, \tag{12e}$$

$$t_i^U(b_i, \boldsymbol{u}_i) \leqslant \tau, \forall i \in N, \tag{12f}$$

where B_0 is the bandwidth of each RB, B is the total uplink bandwidth, τ is the requirement for uplink transmission delay, and DP_0 is the constraint of gradient leakage. Eq. (12c) shows the sum of the bandwidth allocated to each user is less than or equal to the total bandwidth of the uplink. Eq. (12e) indicates the compression requirement for the number of valid gradients uploaded by each user.

3 Analysis of FL Convergence Rate

Since we add defense methods to the original FL algorithm, we need to investigate how transmitting compressed gradient affects the performance of FL to solve Eq. (12). Therefore, in this section, we derive the upper bound on the optimality gap of the defense-added FL algorithm.

We assume that $F(\boldsymbol{w}) = \frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K_i} f(\boldsymbol{w}^j, \boldsymbol{x}_{ik}, y_{ik})$ and $F_i(\boldsymbol{w}) = \sum_{k=1}^{K_i} f(\boldsymbol{w}^j, \boldsymbol{x}_{ik}, y_{ik})$. Based on Eq. (4), the updated global FL model $\boldsymbol{w}$ at step j will be

$$\boldsymbol{w}^{j+1} = \boldsymbol{w}^j - \eta\left(\nabla F(\boldsymbol{w}^j) - \boldsymbol{o}\right), \tag{13}$$

where $\boldsymbol{o} = \nabla F(\boldsymbol{w}^j) - \frac{\sum_{i=1}^{N}\sum_{k=1}^{K_i} \boldsymbol{u}_i \odot \nabla f(\boldsymbol{w}, \boldsymbol{x}_{ik}, y_{ik})}{\sum_{i=1}^{N} K_i}$.

Before deriving the convergence rate of FL, we first make the following assumptions, the same as Ref. [28].

• A1: We assume that the gradient $\nabla F(\boldsymbol{w})$ of $F(\boldsymbol{w})$ is uniformly Lispschitz continuous with respect to w, such that

$$\left\| \nabla F(\boldsymbol{w}^{j+1}) - \nabla F(\boldsymbol{w}^j) \right\| \leqslant L \left\| \boldsymbol{w}^{j+1} - \boldsymbol{w}^j \right\|, \tag{14}$$

where L is a positive constant which is determined by the loss function and $\|\cdot\|$ presents the two-norm.

• A2: We assume that $F(\boldsymbol{w})$ is the μ-strongly convex, such that

$$F(\boldsymbol{w}^{j+1}) \geqslant F(\boldsymbol{w}^j) + (\boldsymbol{w}^{j+1} - \boldsymbol{w}^j)^T \nabla F(\boldsymbol{w}^j) + \frac{\mu}{2}\left\| \boldsymbol{w}^{j+1} - \boldsymbol{w}^j \right\|^2. \tag{15}$$

• A3: We assume that $F(\boldsymbol{w})$ is twice continuously differentiable. Based on A1 and A2, we have

$$\mu I \leq \nabla^2 F(\boldsymbol{w}) \leq LI. \tag{16}$$

• A4: we assume that $\left\| \nabla f(\boldsymbol{w}^j, \boldsymbol{x}_{ik}, y_{ik}) \right\|^2 \leqslant \delta_1 + \delta_2 \left\| \nabla F(\boldsymbol{w}^j) \right\|^2$ with $\delta_1, \delta_2 \geqslant 0$.

Theorem 1: If we run the FL algorithm with the weight matrix $\boldsymbol{u}$, optimal global model $\boldsymbol{w}^*$ and learning rate $\eta = 1/L$, we have

$$F(\boldsymbol{w}^{j+1}) - F(\boldsymbol{w}^*) \leqslant A^t\left(F(\boldsymbol{w}^0) - F(\boldsymbol{w}^*)\right) + \frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N} K_i(1 - u_{i,m})\frac{A^t - 1}{A - 1}, \tag{17}$$

where $A = 1 - \frac{\mu}{L} + \frac{4\mu\delta_2}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N} K_i(1 - u_{i,m})$ and the

proof process of $F\left(\boldsymbol{w}^{j+1}\right)-F\left(\boldsymbol{w}^{*}\right)$ is shown below.

According to the second-order Taylor expansion, $F\left(\boldsymbol{w}^{j+1}\right)$ can be rewritten as

$$F\left(\boldsymbol{w}^{j+1}\right)=F\left(\boldsymbol{w}^{j}\right)+\left(\boldsymbol{w}^{j+1}-\boldsymbol{w}^{j}\right)^{T}\nabla F\left(\boldsymbol{w}^{j}\right)+\frac{1}{2}\left(\boldsymbol{w}^{j+1}-\boldsymbol{w}^{j}\right)^{T}\nabla^{2}F\left(\boldsymbol{w}\right)\left(\boldsymbol{w}^{j+1}-\boldsymbol{w}^{j}\right)\leqslant F\left(\boldsymbol{w}^{j}\right)+\left(\boldsymbol{w}^{j+1}-\boldsymbol{w}^{j}\right)^{T}\nabla F\left(\boldsymbol{w}^{j}\right)+\frac{L}{2}\left\|\boldsymbol{w}^{j+1}-\boldsymbol{w}^{j}\right\|^{2}. \tag{18}$$

Based on Eq. (13) and given the learning rate $\eta=1/L$, the $F\left(\boldsymbol{w}^{j+1}\right)$ can be expressed as

$$F\left(\boldsymbol{w}^{j+1}\right)\leqslant F\left(\boldsymbol{w}^{j}\right)-\eta\left(\nabla F\left(\boldsymbol{w}^{j}\right)-\boldsymbol{o}\right)^{T}\nabla F\left(\boldsymbol{w}^{j}\right)+\frac{L\eta^{2}}{2}\left\|\nabla F\left(\boldsymbol{w}^{j}\right)-\boldsymbol{o}\right\|^{2}=F\left(\boldsymbol{w}^{j}\right)-\frac{1}{2L}\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}+\frac{1}{2L}\left\|\boldsymbol{o}\right\|^{2}. \tag{19}$$

Next, we derive $\left\|\boldsymbol{o}\right\|^{2}$, and the derivation is given as follows:

$$\left\|\boldsymbol{o}\right\|^{2}=\sum_{m=1}^{M}\left\|\boldsymbol{o}_{m}\right\|^{2}=\left\|\nabla F\left(\boldsymbol{w}^{j}\right)-\frac{\sum_{i=1}^{N}\sum_{k=1}^{K_i}\boldsymbol{u}_{i}\odot\nabla f\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)}{\sum_{i=1}^{N}K_i}\right\|^{2}=\sum_{m=1}^{M}\left\|\nabla F\left(\boldsymbol{w}^{j}\right)-\frac{\sum_{i=1}^{N}\sum_{k=1}^{K_i}u_{i,m}\nabla f_{m}\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)}{\sum_{i=1}^{N}K_iu_{i,m}}\right\|^{2}=\sum_{m=1}^{M}\left\|-\frac{\left(K-\sum_{i=1}^{N}K_iu_{i,m}\right)\sum_{i\in\mathcal{D}_{1,m}}\sum_{k=1}^{K_i}\nabla f_{m}\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)}{K\sum_{i=1}^{N}K_iu_{i,m}}+\frac{\sum_{i\in\mathcal{D}_{0,m}}\sum_{k=1}^{K_i}\nabla f_{m}\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)}{K}\right\|^{2}\leqslant\sum_{m=1}^{M}\left(\frac{\left(K-\sum_{i=1}^{N}K_iu_{i,m}\right)\sum_{i\in\mathcal{D}_{1,m}}\sum_{k=1}^{K_i}\left\|\nabla f_{m}\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)\right\|}{K\sum_{i=1}^{N}K_iu_{i,m}}+\frac{\sum_{i\in\mathcal{D}_{0,m}}\sum_{k=1}^{K_i}\left\|\nabla f_{m}\left(\boldsymbol{w},\boldsymbol{x}_{ik},y_{ik}\right)\right\|}{K}\right)^{2}, \tag{20}$$

where $\mathcal{D}_{1,m}$ is the set of users with $u_{i,m}=1$ and $\mathcal{D}_{0,m}$ is the set of users with $u_{i,m}=0$; the inequality equation is realized based on the triangle inequality. According to A4, $\left\|\boldsymbol{o}\right\|^{2}$ can be expressed by

$$\left\|\boldsymbol{o}\right\|^{2}\leqslant\sum_{m=1}^{M}\left(\frac{4}{K^{2}}\left(K-\sum_{i=1}^{N}K_iu_{i,m}\right)^{2}\left(\delta_1+\delta_2\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}\right)\right). \tag{21}$$

Since $0\leqslant K-\sum_{i=1}^{N}K_iu_{i,m}\leqslant K$, we have

$$\left\|\boldsymbol{o}\right\|^{2}\leqslant\sum_{m=1}^{M}\left(\frac{4}{K}\left(K-\sum_{i=1}^{N}K_iu_{i,m}\right)\left(\delta_1+\delta_2\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}\right)\right)\leqslant\frac{4}{K}\sum_{m=1}^{M}\left(\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)\left(\delta_1+\delta_2\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}\right)\right). \tag{22}$$

Substituting Eq. (22) into Eq. (19), we have

$$F\left(\boldsymbol{w}^{j+1}\right)\leqslant F\left(\boldsymbol{w}^{j}\right)+\frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)-\frac{1}{2L}\left(1-\frac{4\delta_2}{K}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)\right)\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}, \tag{23}$$

$$F\left(\boldsymbol{w}^{j+1}\right)-F\left(\boldsymbol{w}^{*}\right)\leqslant\left(F\left(\boldsymbol{w}^{j}\right)-F\left(\boldsymbol{w}^{*}\right)\right)+\frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)-\frac{1}{2L}\left(1-\frac{4\delta_2}{K}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)\right)\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}. \tag{24}$$

Based on Eq.(15) and Eq.(16), we get

$$\left\|\nabla F\left(\boldsymbol{w}^{j}\right)\right\|^{2}\geqslant 2\mu\left(F\left(\boldsymbol{w}^{j}\right)-F\left(\boldsymbol{w}^{*}\right)\right), \tag{25}$$

$$F\left(\boldsymbol{w}^{j+1}\right)-F\left(\boldsymbol{w}^{*}\right)\leqslant\frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)+A\left(F\left(\boldsymbol{w}^{j}\right)-F\left(\boldsymbol{w}^{*}\right)\right), \tag{26}$$

where $A=1-\frac{\mu}{L}+\frac{4\mu\delta_2}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)$. Applying Eq. (26) recursively, we have

$$F\left(\boldsymbol{w}^{j+1}\right)-F\left(\boldsymbol{w}^{*}\right)\leqslant A^{t}\left(F\left(\boldsymbol{w}^{0}\right)-F\left(\boldsymbol{w}^{*}\right)\right)+\frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)\frac{A^{t}-1}{A-1}. \tag{27}$$

This completes the proof.

According to Theorem 1, we obtain the gap between $F\left(\boldsymbol{w}^{j+1}\right)$ and $F\left(\boldsymbol{w}^{*}\right)$. Next, we derive the conditions for δ_2 that guarantees the convergence of FL and simplify the optimization problem in Eq. (12). In Theorem 1, if we set $A<1$ and $A^{t}=0$, we can get $F\left(\boldsymbol{w}^{j+1}\right)-F\left(\boldsymbol{w}^{*}\right)=\sum_{m=1}^{M}\sum_{i=1}^{N}K_i$ (1 –

$u_{i,m}\right)\frac{A^t-1}{A-1}$ and FL converges. Therefore, we only need to make $A=1-\frac{\mu}{L}+\frac{4\mu\delta_2}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)<1$ to ensure FL convergence. Moreover, we can get the relationship between μ and L, $\mu<L$, from Eq. (16). Hence, we get $\delta_2<K\Big/4\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)$. In addition, since δ_2 satisfies the assumption A4, we have

$$0<\delta_2<\frac{K}{\max\limits_{\boldsymbol{u},\boldsymbol{b}}\ 4\sum\limits_{m=1}^{M}\sum\limits_{i=1}^{N}K_i\left(1-u_{i,m}\right)}. \tag{28}$$

4 Optimization of Training Loss

In this section, we aim to minimize the training loss of FL by optimizing the weight compression matrix and RB allocation and considering the constraints under the wireless network. We first simplify the objective function in Eq. (12). From Theorem 1 and the analysis of FL convergence conditions in Section 3, we see that if we want to minimize the training loss of FL, we only need to minimize the gap between $F\left(\boldsymbol{w}^{j+1}\right)$ and $F\left(\boldsymbol{w}^{*}\right)$, under the condition that $A<1$. Then we get

$$\frac{2\delta_1}{LK}\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)\frac{A^t-1}{A-1}=\frac{\frac{2\delta_1}{LK}\sum\limits_{m=1}^{M}\sum\limits_{i=1}^{N}K_i\left(1-u_{i,m}\right)}{\frac{\mu}{L}-\frac{4\mu\delta_2}{LK}\sum\limits_{m=1}^{M}\sum\limits_{i=1}^{N}K_i\left(1-u_{i,m}\right)}. \tag{29}$$

It is obvious to find that to minimize Eq. (29), only $\sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right)$ needs to be minimized, so the optimization problem can be simplified as

$$\min_{\boldsymbol{u},\boldsymbol{b}}\ \sum_{m=1}^{M}\sum_{i=1}^{N}K_i\left(1-u_{i,m}\right), \tag{30}$$

$$\text{s.t. } b_i=\sum_{q=1}^{Q}b_{i,q}\geqslant 1, \tag{30a}$$

$$\sum_{i=1}^{N}b_iB_0\leqslant B,\forall i\in N, \tag{30b}$$

$$u_{i,m}\in\{0,1\},\forall i\in N, \tag{30c}$$

$$\sum_{m=1}^{M}u_{i,m}\leqslant\alpha_iM,\forall i\in N, \tag{30d}$$

$$\sum_{m=1}^{M}\rho_{i,m}u_{i,m}\leqslant DP_0,\forall i\in N, \tag{30e}$$

$$t_i^U\left(b_i,\boldsymbol{u}_i\right)\leqslant\tau,\forall i\in N. \tag{30f}$$

Next, we aim to find the optimal RB allocation and weight compression matrix for each user. To accomplish this, we utilize ant colony optimization (ACO) for a large number of RBs and exhaustive search for a small number of RBs.

5 Simulation Results and Analysis

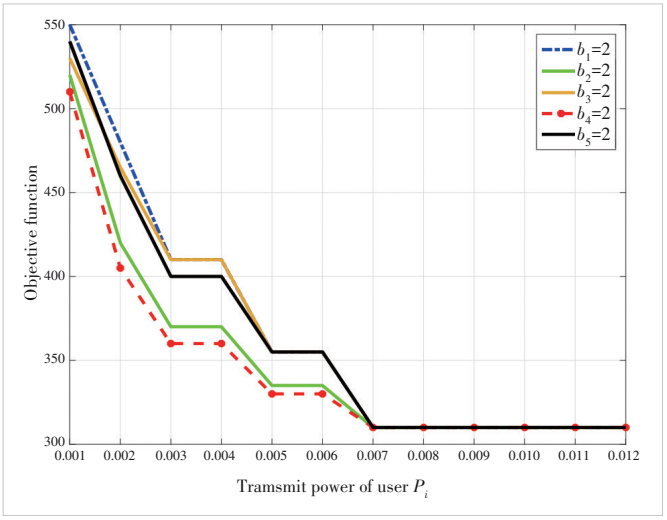
For our simulations, we investigate how the wireless network parameters $\left(P_i,\boldsymbol{b}\right)$, user sample size K_i and gradient compression restrictions α_i affect the convergence rate under the premise that FL can converge. This simulation topology is a circular wireless network area with a central base station serving $N=5$ uniformly distributed users with $d=30$ m. Specifically, we consider only six RBs and five users, first finding all solutions for $\boldsymbol{b}$ by exhaustive search (at most one user is assigned two RBs), and then we solve the optimization problem by using a CVX (a Matlab-based modeling system for convex optimization) toolbox and MOSEK solver in MATLAB. Other key parameters used in this simulation are listed in Table 1.

Fig. 4 shows how the change of P_i and the allocation of RB

▼Table 1. Simulation Parameters

Description	Parameter	Value
Total bandwidth of uplink	B	20 MHz
Bandwidth of each RB	B_0	3.33 MHz
Noise power spectral density	N_0	−174 dBm/MHz
Total number of training samples for user	K_i	[10, 20, 15, 25, 10]
Gradient compression ratio of user	α_i	$\left[\frac{3}{9},\frac{6}{9},\frac{4}{9},\frac{6}{9},\frac{5}{9}\right]$
Number of gradients for each user	M	9
Delay requirement of uplink	τ	2 s
Distance between user and BS	d	30 m
Number of RBs	Q	6
Transmit power of user	P_i	0.001 – 0.012 W

BS: base station RB: resource block



▲Figure 4. Objective function as user power and resource block (RB) allocation varies

change the objective function value, i.e., the convergence rate of the FL algorithm. As can be seen from Fig. 4, with the increase of P_i, the objective function first decreases and then tends to remain unchanged. This is because when the user power increases, the uplink transmission rate of the user becomes larger, allowing the user to upload more gradients, thus accelerating the convergence speed and optimizing the objective function. However, when P_i is very large, the optimal number of gradients that users can upload is already saturated due to DP_0 constraints, so the objective function cannot continue to decline.

Different RB allocations also affect the convergence speed of FL at the same P_i, and here we analyze three cases. The objective function value of the red line in Fig. 4 is the smallest, which is because the number of samples K_4 and the compression ratio α_4 of user 4 are the largest. Therefore, assigning more RBs to the user with more samples and larger α_i can increase the transmission rate of that user and reduce the total delay of uplink transmission, thereby accelerating the convergence speed. When K_i is the same but α_i is different, that is, the blue line and the black line, the larger α_i is, the smaller the value of the objective function is. The reason is that if α_i is large, more gradients can be transmitted, so assigning more RBs to it will result in faster convergence. When α_i is the same and K_i is different, i.e., green and red lines, the larger K_i is, the smaller the value of the objective function is. This is because the larger K_i is, the smaller DP_0 is and the smaller $\rho_{i,m}$ is. According to Constraint (30e), more $u_{i,m}$ can be taken as 1, resulting in a smaller objective function and better performance. Overall, optimizing $\boldsymbol{b}$ can make the convergence faster given a fixed P_i.

6 Conclusions

In this work, we propose a novel defensive framework to protect data privacy from DLG attacks in wireless networks. We jointly optimize RBs allocations and weight compression matrix to minimize FL training loss. We first formulate this optimization problem and simplify it by finding the relationship between the weight matrix and FL convergence rate. Optimal RB allocation is solved by ACO for a large number of RBs and exhaustive search for a small number of RBs. The optimal weight matrix is solved by the CVX toolbox. The simulation results illustrate that optimizing RBs can effectively improve the convergence speed given fixed user power.

References

[1] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: concept and applications [J]. ACM transactions on intelligent systems and technology, 2019, 10 (2): 1 – 19. DOI: 10.1145/3298981
[2] JOCHEMS A, DEIST T M, VAN SOEST J, et al. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital—a real life proof of concept [J]. Radiotherapy and oncology, 2016, 121(3): 459 – 467. DOI: 10.1016/j.radonc.2016.10.002
[3] BONAWITZ K, EICHNER H, GRIESKAMP W, et al. Towards federated learning at scale: system design [C]//Conference on machine learning and systems. MLSys, 2019: 374 – 388
[4] SHOKRI R, STRONATI M, SONG C Z, et al. Membership inference attacks against machine learning models [C]//IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 3 – 18. DOI: 10.1109/SP.2017.41
[5] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep models under the GAN: information leakage from collaborative deep learning [C]//ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 603 – 618. DOI: 10.1145/3133956.3134012
[6] WANG Z B, SONG M K, ZHANG Z F, et al. Beyond inferring class representatives: user-level privacy leakage from federated learning [C]//IEEE Conference on Computer Communications. IEEE, 2019: 2512 – 2520. DOI: 10.1109/INFOCOM.2019.8737416
[7] PHONG L T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning via additively homomorphic encryption [J]. IEEE transactions on information forensics and security, 2017, 13(5): 1333 – 1345. DOI: 10.1109/TIFS.2017.2787987
[8] ZHU L G, LIU Z J, HAN S. Deep leakage from gradients [C]//33rd Conference on Neural Information Processing Systems. NeurIPS, 2019: 8389
[9] ZHAO B, MOPURI K R, BILEN H. iDLG: improved deep leakage from gradients [EB/OL]. [2020-01-08]. https://arxiv.org/abs/2001.02610
[10] YIN H X, MALLYA A, VAHDAT A, et al. See through gradients: image batch recovery via GradInversion [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 16332 – 16341. DOI: 10.1109/CVPR46437.2021.01607
[11] CHEN S, JIA R X, QI G J. Improved techniques for model inversion attack [EB/OL]. [2021-08-19]. https://arxiv.org/abs/2010.04092v1
[12] JERE M S, FARNAN T, KOUSHANFAR F. A taxonomy of attacks on federated learning [J]. IEEE security privacy, 2021, 19(2): 20 – 28. DOI: 10.1109/MSEC.2020.3039941
[13] GEIPING J, BAUERMEISTER H, DRÖGE H, et al. Inverting gradients: how easy is it to break privacy in federated learning? [C]//34th International Conference on Neural Information Processing Systems. NeurIPS, 2020: 16937 – 16947
[14] ZHANG C L, LI S Y, XIA J Z, et al. BatchCrypt: efficient homomorphic encryption for cross-silo federated learning [C]//USENIX Conference on Usenix Annual Technical Conference. ACM, 2020: 493 – 506. DOI: 10.5555/3489146.3489179
[15] CHENG K W, FAN T, JIN Y L, et al. SecureBoost: a lossless federated learning framework [J]. IEEE intelligent systems, 2021, 36(6): 87 – 98. DOI: 10.1109/MIS.2021.3082561
[16] MOHASSEL P, ZHANG Y P. Secureml: a system for scalable privacy-preserving machine learning [C]//IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 19 – 38. DOI: 10.1109/SP.2017.12
[17] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning [C]//ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 1175 – 1191. DOI: 10.1145/3133956.3133982
[18] GEYER R C, KLEIN T, NABI M. Differentially private federated learning: a client level perspective [EB/OL]. [2022-03-01]. https://arxiv. org/abs/1712.07557
[19] MCMAHAN H B, RAMAGE D, TALWAR K, et al. Learning differentially private recurrent language models [EB/OL]. [2022-02-24]. https://arxiv. org/abs/1710.06963
[20] WEI K, LI J, DING M, et al. Federated learning with differential privacy: algorithms and performance analysis [J]. IEEE transactions on information forensics and security, 2020, 15: 3454 – 3469. DOI: 10.1109/TIFS.2020.2988575
[21] WEI W Q, LIU L, LOPER M, et al. A framework for evaluating gradient leakage attacks in federated learning [EB/OL]. [2021-04-23]. https://arxiv.org/abs/2004.10397
[22] SUN J W, LI A, WANG B H, et al. Soteria: provable defense against privacy leakage in federated learning from representation perspective [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 9307 – 9315. DOI: 10.1109/CVPR46437.2021.00919

[23] SCHELIGA D, MÄDER P, SEELAND M. Precode—a generic model extension to prevent deep gradient leakage [C]//IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2022: 3605 – 3614. DOI: 10.1109/WACV51458.2022.00366
[24] WANG J X, GUO S, XIE X, et al. Protect privacy from gradient leakage attack in federated learning [C]//IEEE Conference on Computer Communications. IEEE, 2022: 580 – 589. DOI: 10.1109/INFOCOM48880.2022.9796841
[25] KONEČNÝ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: distributed machine learning for on-device intelligence [EB/OL]. [2021-10-08]. https://arxiv.org/abs/1610.02527
[26] HATAMIZADEH A, YIN H X, MOLCHANOV P, et al. Do gradient inversion attacks make federated learning unsafe? [EB/OL]. [2022-02-14]. https://arxiv.org/abs/2202.06924
[27] TAVANGARAN N, CHEN M Z, YANG Z H, et al. On differential privacy for federated learning in wireless systems with multiple base stations [EB/OL]. [2022-08-25]. https://arxiv.org/abs/2208.11848
[28] CHEN M Z, YANG Z H, SAAD W, et al. A joint learning and communications framework for federated learning over wireless networks [J]. IEEE transactions on wireless communications, 2021, 20(1): 269 – 283. DOI: 10.1109/TWC.2020.3024629

Biographies

DING Yahao received her master's degree in communications and signal processing from Imperial College London, U.K. in 2020. She is currently pursuing her PhD degree in information and communication engineering with King's College London, U.K. Her current research interests include federated learning, security, and UAV swarms.

Mohammad SHIKH-BAHAEI received his BSc degree from the University of Tehran, Iran in 1992, MSc degree from the Sharif University of Technology, Iran in 1994, and PhD degree from King's College London, U.K. in 2000. He has worked for two start-up companies and for National Semiconductor Corporation, USA (now part of Texas Instruments Inc.). In 2002, he joined King's College London, where he is currently a full professor. Since then, he has authored numerous journals and conference papers and worked as an expert consultant to a number of international high-tech companies and legal firms. His research interests are secure communications and connected intelligence, full-duplex and cognitive dense networks, visual data communications over the IoT, applications of wireless communications in healthcare, and communication protocols for autonomous vehicle/drone networks. He has been the founder and the chair of the Wireless Advanced (formerly SPWC) Annual International Conference from 2003 to 2018.

YANG Zhaohui received his PhD degree from Southeast University, China in 2018. From 2018 to 2020, he was a postdoctoral research associate with the Center for Telecommunications Research, Department of Informatics, King's College London, U.K. From 2020 to 2022, he was a research fellow with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a young professor with the College of Information Science and Electronic Engineering, Zhejiang Key Laboratory of Information Processing Communication and Networking, Zhejiang University, China, and also a research scientist with Zhejiang Laboratory. His research interests include joint communication, sensing and computation, federated learning, and semantic communications. He is an associate editor for *IEEE Communications Letters*, *IET Communications*, and *EURASIP Journal on Wireless Communications and Networking*. He was the guest editor of several journals, including *JSAC*, *WCM* and *CM*. He was the co-chair for international workshops with more than ten times, including ICC, GLOBECOM, WCNC, PIMRC and INFOCOM.

HUANG Chongwen (chongwenhuang@zju. edu. cn) received his BSc degree from the Binhai College, Nankai University, China in 2010, and MSc degree from the University of Electronic Science and Technology of China (UESTC), China in 2013. He has been joining the Institute of Electronics, Chinese Academy of Sciences (IECAS) as a research engineer, since July 2013. Since September 2015, he has been starting his PhD journey with the Singapore University of Technology and Design (SUTD), Singapore and CentraleSupélec University, Paris, France under the supervision of Prof. Chau YUEN and Prof. Mérouane DEBBAH. From October 2019 to September 2020, he was a post-doctoral researcher at SUTD. Since September 2020, he has been joining Zhejiang University as a Tenure-Track Young Professor. His main research interests include holographic MIMO surface/reconfigurable intelligent surface, B5G/6G wireless communications, mmWave/THz communications, and deep learning technologies for wireless communications. He was a recipient of the IEEE Marconi Prize Paper Award in Wireless Communications in 2021. He was also a recipient of the Singapore Government PhD Scholarship and received PHC Merlion PhD Grant (2016 – 2019) for studying in CentraleSupélec, France. He has been serving as an editor of *IEEE Communications Letter*, *Signal Processing* (Elsevier), *EURASIP Journal on Wireless Communications and Networking*, and *Physical Communication* since 2021. In addition, he has served as the chair of several wireless communications flagship conferences, including the session chair of 2021 IEEE WCNC, 2021 IEEE VTC-Fall, and the symposium chair of IEEE WCSP 2021.

YUAN Weijie received his BE degree from the Beijing Institute of Technology, China in 2013, and PhD degree from the University of Technology Sydney, Australia in 2019. In 2016, he was a visiting PhD student with the Institute of Telecommunications, Vienna University of Technology, Austria. He was a research assistant with the University of Sydney, Australia, a visiting associate fellow with the University of Wollongong, Australia and a visiting fellow with the University of Southampton, U.K. from 2017 to 2019. From 2019 to 2021, he was a research associate with the University of New South Wales, Australia. He is currently an assistant professor with the Department of Electrical and Electronic Engineering, Southern University of Science and Technology, China. He was a recipient of the Best PhD Thesis Award from the Chinese Institute of Electronics and an Exemplary Reviewer from IEEE TCOM/WCL. He currently serves as an associate editor of *IEEE Communications Letters*, an associate editor and an award committee member of *EURASIP Journal on Advances in Signal Processing*. He has led the guest editorial teams for three special issues in *IEEE Communications Magazine*, *IEEE Transactions on Green Communications and Networking*, and *China Communications*. He was an organizer/the chair of several workshops and special sessions on orthogonal time frequency space and integrated sensing and communication in flagship IEEE and ACM conferences, including IEEE ICC, IEEE/CIC ICCC, IEEE SPAWC, IEEE VTC, IEEE WCNC, IEEE ICASSP, and ACM MobiCom. He is the founding chair of the IEEE ComSoc Special Interest Group on Orthogonal Time Frequency Space.

Efficient Bandwidth Allocation and Computation Configuration in Industrial IoT

HUANG Rui, LI Huilin, ZHANG Yongmin
(Central South University, Changsha 410012, China)

DOI: 10.12142/ZTECOM.202301007
https://kns.cnki.net/kcms/detail/34.1294.TN.20230213.1639.003.html, published online February 14, 2023
Manuscript received: 2022-12-01

Abstract: With the advancement of the Industrial Internet of Things (IoT), the rapidly growing demand for data collection and processing poses a huge challenge to the design of data transmission and computation resources in the industrial scenario. Taking advantage of improved model accuracy by machine learning algorithms, we investigate the inner relationship of system performance and data transmission and computation resources, and then analyze the impacts of bandwidth allocation and computation resources on the accuracy of the system model in this paper. A joint bandwidth allocation and computation resource configuration scheme is proposed and the Karush-Kuhn-Tucker (KKT) conditions are used to get an optimal bandwidth allocation and computation configuration decision, which can minimize the total computation resource requirement and ensure the system accuracy meets the industrial requirements. Simulation results show that the proposed bandwidth allocation and computation resource configuration scheme can reduce the computing resource usage by 10% when compared to the average allocation strategy.

Keywords: bandwidth allocation; computation resource management; industrial IoT; system accuracy

Citation (IEEE Format): R. Huang, H. L. Li, and Y. M. Zhang, "Efficient bandwidth allocation and computation configuration in industrial IoT," *ZTE Communications*, vol. 21, no. 1, pp. 55–63, Mar. 20223. doi: 10.12142/ZTECOM.202301007.

1 Introduction

In recent years, the advances in computation, communication and application design and the rapid development of the Internet of Things (IoT) have been driving the realization of intelligence and automation in the industry[1-2]. Through various IoT devices, a large number of data can be collected, such as images, sounds and temperatures, to judge the operating status of equipment and efficient follow-up maintenance/management strategies are then made. For the traditional Industrial IoT with the cloud, the system performance may be affected by the network performance and computing capability of the cloud since data should be transmitted to a remote cloud via the Internet for processing. With the development of industry, more and more data needs to be collected and processed in real time, leading to explosive growth in communication overhead and computation requirements, which brings significant challenges in the design of Industrial IoT, especially with high-reliability requirements.

To solve data transmission issues in the Industrial IoT, the communication framework has been updated to improve the speed and reliability of data transmission[3-4] and a new wireless transmission system framework has also been proposed to help design an operable and effective end-to-end wireless solution[5]. Moreover, many works have focused on improving data transmission technologies, such as time slot frequency hopping technologies[6] and clustering of data transmission[7]. Besides optimizing the communication framework of the Industrial IoT, some researchers have considered and studied the energy consumption, delay, cost and other parameters associated with the data transmission issues; for example, Ref. [8] proposed a bandwidth allocation strategy based on deep reinforcement learning algorithm and Ref. [9] enables the control of transmission energy consumption for the dynamic change of bandwidth. To deal with the large and unstable communication latency, an edge computing system with computing resources deployed at the network edge has been introduced into the Industrial IoT and become a potential mainstream solution[10-12]. These works alleviate the problem of insufficient wireless resources.

To solve the computation resource issues, there exists a lot of work focusing on the optimization of task offloading performance for cloud computing/edge computing or collaborative edge-cloud computing, such as computation delay, energy consumption, resource efficiency and data quality, to guarantee the quality of computation service[13-16]. With the gradual deepening of machine learning research, it has been discov-

This work has been supported in part by the National Natural Science Foundation of China under Grant No. 62172445 and in part by the Young Talents Plan of Hunan Province, China.
Corresponding author: ZHANG Yongmin

ered that the number of training epochs directly affects the accuracy of the system model after training[17 - 18]. Considering that allocated computation resources can determine the training epochs in a given time scale, some researchers have investigated the relationships among accuracy of the system model, the number of processed data, the number of computation resources, the training speed/delay and the energy consumption, and made use of machine learning based algorithms to further improve the performance of the computation system[19 - 22]. In such a way, the performance of the computation system can be further improved.

However, most of the current works do not consider the inner relationship between the bandwidth allocation and the computation resource management incurred by the data transmission and just assume that the IoT devices transmit all of collected data to the edge server via access points (AP). The AP needs to try its best to forward the data to the edge server and the edge server processes all the received data making use of its available computation resources. Unfortunately, with the explosive increase of IoT devices, it is difficult for the existing Industrial IoT system to carry on such a heavy workload, which may lead to network congestion, even network crash when wireless communication resources are exhausted. Therefore, to solve the data transmission problem, it is worthwhile to consider the inner relationship among the computation resources, the accuracy of the system model and the data transmission. Making up for the shortage of wireless resources by increasing the computing resources can guarantee system accuracy.

In this paper, we aim at the scenario of resource management in the Industrial IoT, which can allocate the wireless communication resources by the AP and train a high-accuracy model by computation resources at the edge server. First, we model the available channel bandwidth for each IoT device based on the allocated bandwidth and the distance between the IoT device and the AP. Second, we formulate the bandwidth allocation and computation configuration as a resource requirement minimization problem. Then, we analyze the relationship among the transmitted data, the computation resources and the system accuracy, and design a heuristic algorithm to obtain the optimal computation resources allocation and communication resources management to each IoT device. The contributions of this paper can be summarized as follows:

• The bandwidth allocation and computation resource management problem for Industrial IoT is formulated as a cost minimization problem with the given accuracy requirement.

• The relationship among the accuracy of the system model, the transmitted data and the computation resources is investigated and an efficient bandwidth allocation and resource management scheme is designed to satisfy the system requirement with a minimal resource requirement.

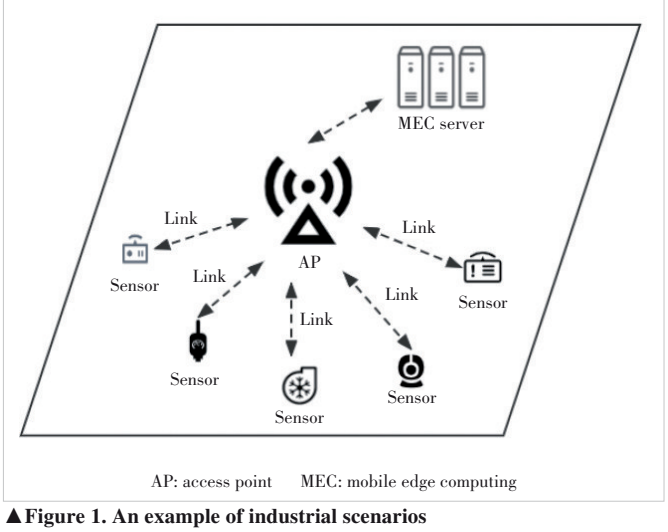
• Simulation results show the proposed algorithm can minimize the resource requirement with a performance guarantee.

The rest of the paper is organized as follows. Section 2 presents the system model and problem formulation. An algorithm that can minimize the resource requirement with performance guarantee is proposed in Section 3. An operational performance analysis is demonstrated based on simulation results in Section 4. Finally, Section 5 concludes our work.

2 System Model and Problem Formulation

Fig. 1 shows an industrial scenario of the Industrial IoT system. In this system, there are N IoT devices and the set of IoT devices is denoted as $\boldsymbol{N} = \{1, 2, \cdots, N\}$, and several APs (small cell base stations or Wi-Fi APs) with edge servers. Generally, an IoT device collects monitoring data and transmits the data to the edge server via AP using the wireless communication technology, the AP allocates the available wireless bandwidth to each IoT device and forwards the monitoring data to the edge server for processing, and the edge server processes the monitoring data using machine learning models to satisfy the requirement of system performance. Here, we assume that each IoT device can adjust its monitoring data according to the available wireless bandwidth, and the connections of the IoT devices and APs are given due to specified monitoring objects. Thus, for ease of description, we only focus on optimizing communication bandwidth allocation and computation resource management strategies in a single AP with an edge server with multiple IoT device scenarios in this paper, but the results can be extended to multiple APs with multiple edge servers based on the deployment of the inference model. Note that we mainly focus on the allocation of communication resources and the configuration of computation resources at the edge server.

Considering the time-varying feature of industrial scenarios, one optimization period can be divided into T time segments $\boldsymbol{T} = \{1, 2, \cdots, T\}$, and t represents the t-th time segment. The



▲Figure 1. An example of industrial scenarios

accuracy requirement of a system model for one IoT device during one time segment is a constant and can be changed at different time segments.

2.1 Communication Model

Generally, all IoT devices need to send their real-time monitoring data to the edge server for processing via the AP. It means that several IoT devices will transmit their data to the AP simultaneously. To mitigate interference among IoT devices, some effective interference cancellation techniques, such as orthogonal frequency-division multiple access (OFDMA) and time division multiple access (TDMA), can be used by the AP. In this paper, we assume that OFDMA is used for wireless communications. Besides the interference, signal pass loss is another important factor that affects data transmission. According to Refs. [23] and [24], the pass loss can be formulated as a function of transmission distance with a path loss exponent $2 \leqslant \alpha \leqslant 4$. Let $h_{n,t}$ denote the small-scale channel gain from the n-th mobile device to the AP during time segment t. The achievable data transmission rate for IoT device n during time segment t, denoted by $R_{n,t}$, can be given by

$$R_{n,t} = w_{n,t} \log_2\left(1 + \frac{P_{n,t}\left|h_{n,t}\right|^2}{\left(d_{n,t}\right)^{\alpha}\sigma^2}\right), n = 1, 2, \cdots, N, \tag{1}$$

where $w_{n,t}$ denotes the allocated bandwidth for IoT device n during time segment t, $P_{n,t}$ denotes the transmission power of IoT device n during time segment t, $d_{n,t}$ denotes the distance between IoT device n and AP, and σ^2 denotes the background noise power. Generally, $w_{n,t}$ is determined by AP, $d_{n,t}$ and σ^2 are constants, and the value of $P_{n,t}$ can be calculated by the power control algorithms[24-25]. Due to the limitation of AP's wireless communication resources, the total bandwidths that can be allocated to IoT devices have an upper bound, denoted by $\bar{W}$. Thus, we have

$$\sum_n w_{n,t} \leqslant \bar{W}. \tag{2}$$

It is obvious that the bandwidth allocation strategy of the AP should consider the distance $d_{n,t}$ for IoT device n and the requirements of all the IoT devices. The data set of IoT device n that have been transmitted to the edge server during time segment t, denoted by $D_{n,t}$, is

$$D_{n,t} = R_{n,t} * t, \tag{3}$$

where t is the total number of time units in one time segment.

2.2 Computation Model

The edge server can make use of the received data and the machine learning based algorithm to train a high-accuracy system model for each IoT device. We define accuracy as the opposite of the loss function in a training model based on federated learning. In general, the performance of the system model achieved by the machine learning algorithm can be affected by multiple factors, including feature selection, user-defined parameters, data sets and computation resources for training. In this paper, we mainly consider the impact of the data set and the computation resources on the training results and intend to find an appropriate data set and computing resources to satisfy the requirements of system accuracy.

According to Refs. [17] and [19], the accuracy of the system model through training of traditional machine learning algorithms generally tends to increase with the increase of the data set. At the same time, due to the noise $\delta_{n,t}$ in the data set and the limitation of the model capacity, the accuracy growth rate of the system model will gradually slow down until it becomes stable[26]. Besides, the precision of a machine learning algorithm, such as a neural network, has a logarithmic relationship with the number of training epochs[17]. Thus, with the increase of computation resources, the number of training epochs within the limited time scale can be increased, which can improve the precision of the system model in a logarithmic form. Thus, in this paper, the accuracy of the system model for IoT device n during time segment t, denoted by $\xi_{n,t}$, can be modeled as

$$\xi_{n,t} = a \log_{10}\left(\frac{C_{n,t}}{C_{\text{epoch}}} * \frac{D_{n,t}}{D_{\text{unit}}} + b\right) + \delta_n, \tag{4}$$

where a and b are accuracy parameters based on the machine learning algorithm and $0 \leqslant a,b \leqslant 1$, C_{epoch} denotes the computation resources required for training the data set for one epoch, D_{unit} denotes a reference unit of the data set for training, and δ_n is the influence factor of the noise in the data set on the accuracy. Generally, $\delta_{n,t}$ $(-1 \leqslant \delta_n < 0)$ is a constant affected by the noise during time segment t[27].

It can be found that both the data set and computation resources can affect the accuracy of the system model. Because of this, it provides the industrial IoT with an opportunity to solve the wireless resource issues by managing the computation resources.

2.3 Problem Formulation

In this paper, we intend to design an efficient bandwidth allocation and computation resource management scheme for the Industrial IoT to satisfy the accuracy requirement of each IoT devices. Let $\underline{\xi_{n,t}}$ denote the accuracy requirement of IoT device n during time segment t. Thus, we have

$$\xi_{n,t} \geqslant \underline{\xi_{n,t}}, \ \forall n, t. \tag{5}$$

To achieve the accuracy requirements of each IoT device, the AP allocates its available communication resources to IoT

devices for data transmission while the edge server manages the computation resources for data processing. In other words, when wireless communication resources are scarce/costly, more computation resources can be used to improve the accuracy of the system model. Otherwise, more computation resources can be saved to keep the accuracy of the system model at a given level.

Considering that the wireless communication resources for each AP are limited, our objective function is to minimize the total computation resources requirement, which can both minimize the operating cost and identify the bottleneck of the system performance. The bandwidth allocation and the computation resource management problem can be formulated as following

$$\mathbf{P1}: \min_{\vec{w},\vec{C}} \sum_{n} C_{n,t}, \tag{6}$$

$$\text{s.t.} \sum_{n} w_{n,t} \leqslant \bar{W}, \quad \forall t, \tag{7}$$

$$\xi_{n,t} \geqslant \underline{\xi}_{n,t}, \quad \forall n, t, \tag{8}$$

where $\vec{\boldsymbol{w}} = \left\{w_{n,t}, \forall n,t\right\}$ is the set of the bandwidth allocation of the AP and $\vec{\boldsymbol{C}} = \left\{C_{n,t}, \forall n,t\right\}$ is the set of the computation resources for data processing. The objective of Problem **P1** is to obtain the optimal bandwidth allocation, which can minimize the total number of computation resources. The first constraint ensures the sum of the bandwidth resources that are allocated to the IoT devices does not exceed the total numer of the available bandwidth resources of the AP. The second constraint guarantees that the accuracy of the system model for each IoT device can meet the industrial requirements.

3 Optimal Bandwidth Allocation and Computation Configuration Scheme

To solve this problem, from the perspective of the edge server, we study the relationship among accuracy $\xi_{n,t}$, computation resources $C_{n,t}$, and data set $D_{n,t}$ of a specific IoT device n. To satisfy the accuracy requirement of each IoT device, we can analyze the influence of data set $D_{n,t}$ on the computation resource requirements for each IoT device. Then, through the communication model, the relationship between the data set and the allocated bandwidth resources can be obtained. Thus, we can derive the impact of bandwidth allocation decisions on the computation resource requirements for each IoT device.

By analyzing the relationship among $\xi_{n,t}$, $C_{n,t}$ and $D_{n,t}$, we have the following results.

Lemma 1: The accuracy $\xi_{n,t}$ obtained by the edge server is an increasing and concave function with respect to the computation resources $C_{n,t}$ when the data set $D_{n,t}$ is given.

Proof: According to Eq. (4), we can derive the first and second derivatives of $\xi_{n,t}$ with respect to $C_{n,t}$ as follows:

$$\frac{\partial \xi_{n,t}}{\partial C_{n,t}} = \frac{1}{\ln 10} \frac{a}{\frac{C_{n,t}}{C_{\text{epoch}}} * \frac{D_{n,t}}{D_{\text{unit}}} + b} \frac{D_{n,t}}{C_{\text{epoch}} * D_{\text{unit}}}, \tag{9}$$

$$\frac{\partial^2 \xi_{n,t}}{\partial {C_{n,t}}^2} = \frac{1}{\ln 10} \left(\frac{D_{n,t}}{C_{\text{epoch}} * D_{\text{unit}}}\right)^2 \frac{-a}{\left(\frac{C_{n,t}}{C_{\text{epoch}}} * \frac{D_{n,t}}{D_{\text{unit}}} + b\right)^2}. \tag{10}$$

Since each item of Eq. (9) is positive, $(\partial \xi_{n,t}) / (\partial C_{n,t}) > 0$ holds. Since only $-a$ in Eq. (10) is negative, $(\partial^2 \xi_{n,t}) / (\partial {C_{n,t}}^2) < 0$ holds. Thus $\xi_{n,t}$ is an increasing and concave function of $C_{n,t}$.

Lemma 2: The accuracy $\xi_{n,t}$ obtained by the edge server is an increasing and concave function with respect to the data set $D_{n,t}$ when the computation resources $C_{n,t}$ is given.

The proof of Lemma 2 is similar to that of Lemma 1, so we omit it. We can also derive that $(\partial \xi_{n,t}) / (\partial D_{n,t}) > 0$ and $(\partial^2 \xi_{n,t}) / (\partial {D_{n,t}}^2) < 0$. Thus $\xi_{n,t}$ is an increasing and concave function of $D_{n,t}$.

Theorem 1: Accuracy $\xi_{n,t}$ is an increasing and concave function with respect to both the computation resources $C_{n,t}$ and the data set $D_{n,t}$.

Proof: According to Lemma 1 and Lemma 2, $\xi_{n,t}$ is an increasing and concave function with respect to $C_{n,t}$ or $D_{n,t}$ when the other variable is given. Furthermore, since $C_{n,t}$ and $D_{n,t}$ are independent, according to Ref. [28], it can be proved that $\xi_{n,t}$ is an increasing and concave function with respect to $C_{n,t}$ and $D_{n,t}$.

Based on Theorem 1, we have the following theorem for the optimal solution to **P1**.

Theorem 2: The optimal solution to **P1** should satisfy $\{\xi_{n,t} = \underline{\xi}_{n,t}, \forall n\}$ and $\sum_n w_{n,t} = \bar{W}$.

Proof: According to Theorem 1, for a specific IoT device n, $\xi_{n,t}$ is an increasing function of $C_{n,t}$ and $D_{n,t}$. First, we can prove that $\sum_n w_{n,t} = \bar{W}$ is a necessary condition for the optimal solution by contradiction as follows.

Assuming that there exists an optimal solution, denoted by $\{w'_{n,t}, \forall n\}$, satisfying $\sum_n w'_{n,t} < \bar{W}$ and $\xi_{n,t} = \underline{\xi}_{n,t}$, we can increase any $w'_{n,t}$ by δ_n, $0 < \delta_n \leqslant \bar{W} - \sum_n w'_{n,t}$, and find a smaller $C'_{n,t}$ satisfying $C'_{n,t} < C_{n,t}$ to make $\xi_{n,t} = \underline{\xi}_{n,t}$. This contradicts the objective function. Thus, $\sum_n w_{n,t} = \bar{W}$ always holds for the optimal solution to **P1**.

Then, we can prove that $\{\xi_{n,t} = \underline{\xi}_{n,t}, \forall n\}$ is a necessary condition for the optimal solution by contradiction. If there exists an optimal solution, denoted by $\xi'_{n,t}$, satisfying $\xi'_{n,t} > \underline{\xi}_{n,t}$ and $\sum_n w'_{n,t} < \bar{W}$. According to Theorem 1, we can decrease $C_{n,t}$ to

make $\xi_{n,t}=\underline{\xi_{n,t}}$ and keep $\sum_n w'_{n,t}=\bar{W}$. This contradicts the objective function. Thus, $\{\xi_{n,t}=\underline{\xi_{n,t}}, \forall n\}$ is another necessary condition for the optimal solution to P1.

According to Theorem 2, we have the relationship among $\xi_{n,t}$, $C_{n,t}$ and $D_{n,t}$ as follows:

$$\xi_{n,t}=\underline{\xi_{n,t}}=a\log_{10}\left(\frac{C_{n,t}}{C_{\text{epoch}}}*\frac{D_{n,t}}{D_{\text{unit}}}+b\right)+\delta_n. \tag{11}$$

Therefore, we can obtain the expression of $C_{n,t}$ about $D_{n,t}$ as follows:

$$C_{n,t}=C_{\text{epoch}}D_{\text{unit}}\left(10^{\frac{\underline{\xi_{n,t}}-\delta_n}{a}}-b\right)\frac{1}{D_{n,t}}. \tag{12}$$

Based on Eq. (12), we have the following property:

Lemma 3: The optimal computation resource $C_{n,t}$ is a decreasing and convex function of the data set $D_{n,t}$.

Proof: Based on Eq. (11), we can calculate the derivative of $C_{n,t}$ with respect to $D_{n,t}$ as follows:

$$\frac{\partial C_{n,t}}{\partial D_{n,t}}=C_{\text{epoch}}D_{\text{unit}}\left(10^{\frac{\underline{\xi_{n,t}}-\delta_n}{a}}-b\right)\frac{-1}{{D_{n,t}}^2}, \tag{13}$$

$$\frac{\partial^2 C_{n,t}}{\partial {D_{n,t}}^2}=C_{\text{epoch}}D_{\text{unit}}\left(10^{\frac{\underline{\xi_{n,t}}-\delta_n}{a}}-b\right)\frac{2}{{D_{n,t}}^3}. \tag{14}$$

It can be found that $(\partial\xi_{n,t})/(\partial C_{n,t})>0$ and $(\partial^2\xi_{n,t})/(\partial {C_{n,t}}^2)<0$, which means that $C_{n,t}$ is a decreasing and convex function of $D_{n,t}$.

According to the definition of data transmission rate $R_{n,t}$ in Eq. (1) and the data set $D_{n,t}$ in Eq. (3), it can be found that $D_{n,t}$ is a linear function of the bandwidth allocation $w_{n,t}$. Thus, we have the following lemma:

Lemma 4: The optimal computation resource $C_{n,t}$ is a decreasing and convex function of the bandwidth allocation $w_{n,t}$.

Proof: According to Lemma 3, $C_{n,t}$ is a decreasing and convex function of $D_{n,t}$. Thus, we have $(\partial\xi_{n,t})/(\partial C_{n,t})>0$ and $(\partial^2\xi_{n,t})/(\partial {C_{n,t}}^2)<0$. Since $D_{n,t}$ is a linear function of $w_{n,t}$, according to the chain rule of derivation, $(\partial\xi_{n,t})/(\partial C_{n,t})>0$ and $(\partial^2\xi_{n,t})/(\partial {C_{n,t}}^2)<0$ hold. Thus, $C_{n,t}$ is a decreasing and convex function of $w_{n,t}$.

Theorem 3: There exists a unique optimal solution $\{w_{n,t}, \forall n,t\}$ for P1.

Proof: According to Lemma 4, the objective function of P1 is a decrease and convex function of the bandwidth allocation $w_{n,t}$. It can be found that the first and second constraints are linear constraints of $w_{n,t}$. Hence, P1 is a convex optimization problem with respect to $w_{n,t}$. According to the properties of the convex optimization problem in Ref. [28], there exists a unique optimal bandwidth allocation $\{w_{n,t}, \forall n,t\}$ for P1.

Since P1 is a convex optimization problem, based on its KKT conditions, the optimal solution can be achieved by the following theorem.

Theorem 4: The optimal solution to P1 is

$$w^*_{n,t}=\frac{\bar{W}\sqrt{\frac{\beta^1_{n,t}}{\beta^2_{n,t}}}}{\sum_{n'}\sqrt{\frac{\beta_{1,n'}}{\beta_{2,n'}}}}. \tag{15}$$

Proof: Generally, since P1 for one time segment is independent with the other time segments, we can solve P1 for each time segment t.

Let v_t be the Lagrange multiplier associated with the constraint $\sum_n w_{n,t}\leqslant\bar{W}$. The Lagrangian of P1 is

$$L\left(w_{n,t},v_t\right)=\sum_n C_{n,t}+v_t\left(\sum_n w_{n,t}-\bar{W}\right)=$$
$$-\bar{W}*v_t+\sum_n\left(C_{n,t}+v_t*w_{n,t}\right). \tag{16}$$

It can be found that the above equation is separable. Thus, the dual function is

$$g\left(v_t\right)=-\bar{W}*v_t+\min_{\vec{w}}\sum_n\left(C_{n,t}+v_t*w_{n,t}\right). \tag{17}$$

According to Lemma 4, $C_{n,t}$ is a decreasing and convex function of $w_{n,t}$. Thus, we can get the minimal value of $\sum_n(C_{n,t}+v_t*w_{n,t})$ when $[\partial C_{n,t}+v_t*w_{n,t}]/\left(\partial w_{n,t}\right)=0$, which means

$$w^*_{n,t}=\sqrt{\frac{\beta^1_{n,t}}{\beta^2_{n,t}}}*\frac{1}{\sqrt{v_t}}, \tag{18}$$

where $\beta^1_{n,t}=C_{\text{epoch}}D_{\text{unit}}(10^{\frac{\underline{\xi_{n,t}}-\delta_n}{a}}-b)$ and $\beta^2_{n,t}=\log_2\left(1+\frac{P_{n,t}\left|h_{n,t}\right|^2}{(d_{n,t})^{\alpha}\sigma^2}\right)$. Thus we can rewrite Eq. (17) as

$$g\left(v_t\right)=-\bar{W}*v_t+2\sqrt{v_t}\sum_n\sqrt{\frac{\beta^1_{n,t}}{\beta^2_{n,t}}}, \tag{19}$$

and the dual problem is

$$\min_{v_t}\quad g\left(v_t\right),$$
$$\text{s.t.}\quad v_t\geqslant 0. \tag{20}$$

Since $[\partial^2 g(v_t)]/(\partial v_t^2) = -\frac{1}{2} v_t^{-\frac{3}{2}} \sum_n \sqrt{\beta_{n,t}^1 / \beta_{n,t}^2} < 0$, the dual function $g(v_t)$ is a concave function. According to the convex optimization theorem, the optimal v_t^* should satisfy $[\partial g(v_t)]/(\partial v_t) = 0$. Thus, the optimal v_t^* can be calculated by

$$v_t^* = \left(\frac{1}{\bar{W}} \sum_n \sqrt{\frac{\beta_{n,t}^1}{\beta_{n,t}^2}} \right)^2, \tag{21}$$

and substituting v_t^* into Eq. (18), we can obtain Eq. (15).

Therefore, according to the location information and accuracy requirements of all IoT devices, we design an efficient bandwidth allocation and computation configuration algorithm, named EBACC, which can solve P1 and get the optimal decision of bandwidth allocation and computation configuration.

Algorithm 1. Efficient Bandwidth Allocation and Computation Configuration Algorithm (EBACC)

1: **for** each time segment t, $t \in [1,T]$, **do**
2: **Input:** $\{d_{n,t}, P_{n,t}, \underline{\xi}_{n,t}, \forall n\}$.
3: According to Eq. (1) in the communication model, calculate $\beta_{n,t}^1$ of each IoT device.
4: According to Eq. (13) in the computation model, calculate $\beta_{n,t}^2$ of each IoT device.
5: According to Eq. (15), calculate the decision of bandwidth allocation $\vec{\boldsymbol{w}}_t$ by using $\beta_{n,t}^1$ and $\beta_{n,t}^2$.
6: According to Eqs. (1) and (4), calculate the decision of computation configuration $\vec{\boldsymbol{C}}_t$ based on $\vec{\boldsymbol{w}}_t$.
7: **Output:** optimal $\vec{\boldsymbol{w}}_t$ and $\vec{\boldsymbol{C}}_t$.
8: **end for**

4 Simulation

In this section, numerical experiments have been conducted to verify the correctness of the lemmas and performance of the proposed algorithm EBACC. We first consider a scenario where the AP has a coverage range of 200 m and there are $N = 60$ randomly scattered IoT devices within the coverage region. We randomly generate the distance $d_{n,t}$ between each IoT device and AP within $[10\text{ m}, 200\text{ m}]$. In the communication model, we assume that the upper bound of total bandwidth resources of the AP is $\bar{W} = 200$ MHz. And the reference signal-to-noise ratio (SNR) at the transmission distance $d_0 = 10$ m is set to $\gamma_0 = \left[P_{n,t} \left| h_{n,t} \right|^2 \right] / (d_{n,t})^{\alpha} \sigma^2 = 80$ dB. The propagation distance can be converted to $d'_{n,t} = d_{n,t}/d_0$, which is within $[1, 20]$, and the path loss exponent is set to $\alpha = 3$. Meanwhile, we randomly generate the accuracy requirement of each device within $[0.8, 0.95]$.

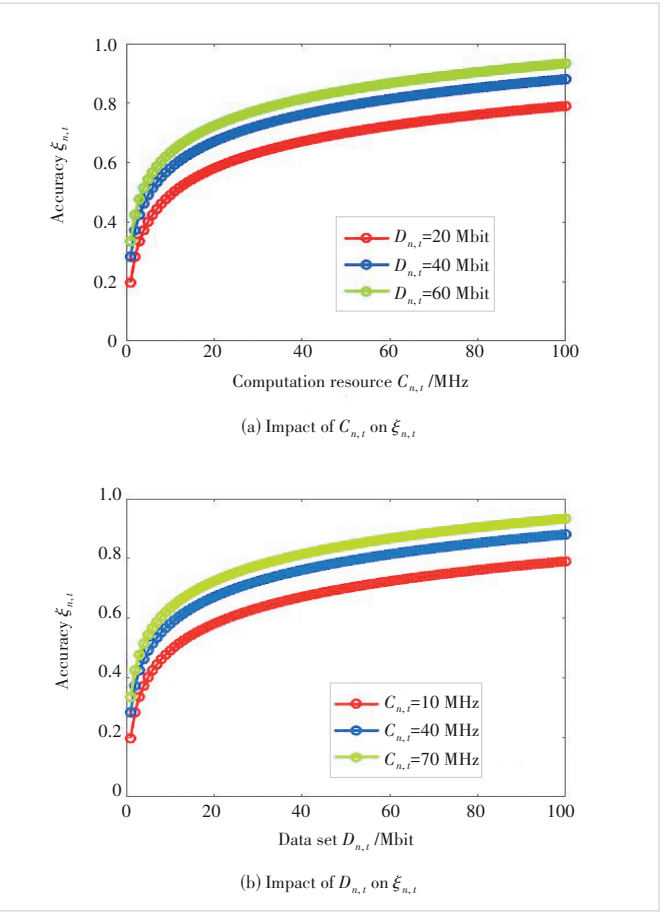
In the following subsection, we firstly explore the relationship between variables in the computation and communication models. Then we verify the correctness of the lemmas in Section 3. Last, we evaluate the performance of the proposed algorithm EBACC, which can get the optimal bandwidth resource allocation to minimize the total computation resources while satisfying the accuracy requirements of IoT devices.

4.1 Impact of Computation Resources and Data Set on Accuracy of Training Results

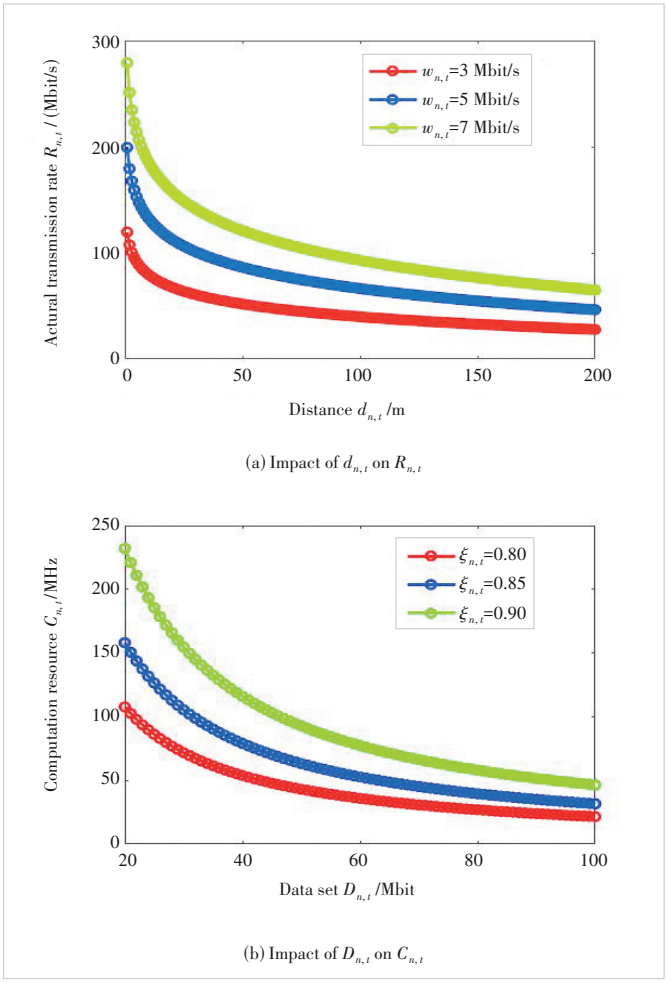
As shown in Figs. 2(a) and 2(b), the accuracy of training results shows a growing trend with the increase of data set size or computing resources, and the growth rate will be gradually slowed down, which verifies the conclusion of Lemma 1 that the accuracy $\xi_{n,t}$ is an increasing concave function with respect to $C_{n,t}$ and $D_{n,t}$. Thus, we can configure more computing resources or upload more data to improve the accuracy of training results.

4.2 Impact of Distance from IoT devices to AP on Actual Transmission Rate

As shown in Fig. 3(a), it is obvious that the actual transmission rate $R_{n,t}$ is a decreasing and convex function of distance



▲ **Figure 2. Impact of computation resources and data set on the accuracy of training results**



(a) Impact of $d_{n,t}$ on $R_{n,t}$

(b) Impact of $D_{n,t}$ on $C_{n,t}$

▲ **Figure 3. Impact of distance from IoT devices to AP on the actual transmission rate and that of data set on computation resources**

$d_{n,t}$ from IoT devices to the AP. The closer the IoT device is to the AP, the higher the actual transmission rate will be. Thus, we can allocate more bandwidth resources to the farther IoT devices, which can reduce the impact of distance to get smaller computing resource requirements.

4.3 Impact of Data Set on Computation Resources Requirement

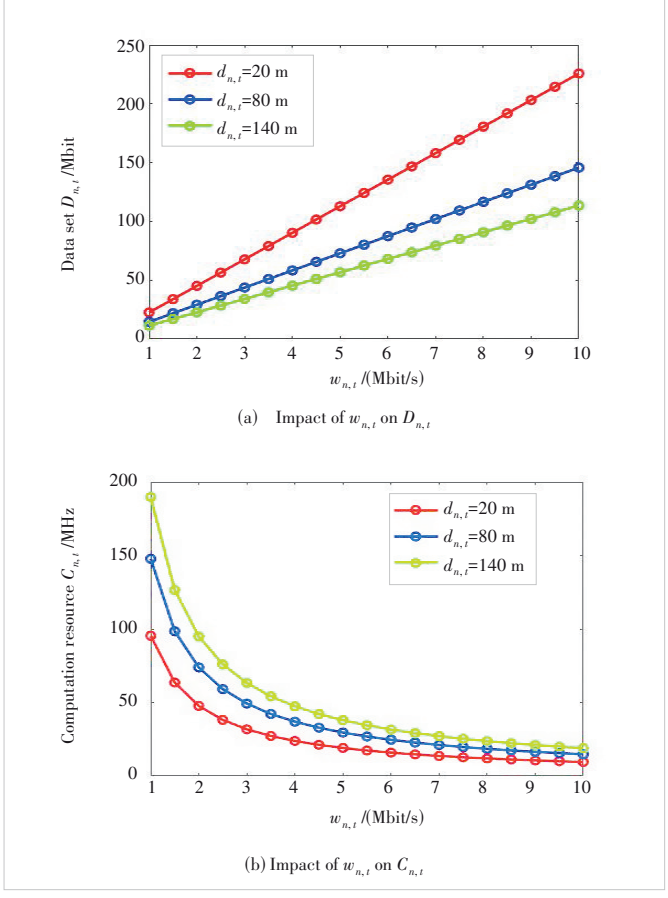
As shown in Fig. 3(b), when the accuracy $\xi_{n,t}$ is given, the computation resource requirement $C_{n,t}$ by the IoT device is a decreasing and convex function of the data set $D_{n,t}$, which has been proved by Lemma 3. It means when the uploaded data set is larger, the computing resources required by the model will be reduced. In addition, it can be found that, with the improvement of the accuracy $\xi_{n,t}$ of model requirements, the computation resources $C_{n,t}$ will become larger. Therefore, when the accuracy of model requirement is given, we can make a trade-off between the number of uploaded data and computing resources.

4.4 Impact of Bandwidth Resources Allocation on Data Set and Computation Resources Requirement

As shown in Fig. 4(a), during one time segment, data set $D_{n,t}$ that the IoT device can upload to the edge server is a linear and increasing function of the allocated bandwidth resources $w_{n,t}$. It can be found that the IoT device closer to the AP has a higher positive slope. Meanwhile, as shown in Fig. 4(b), for an IoT device, the computation resources requirement $C_{n,t}$ is a decreasing and convex function of the bandwidth resources allocated to it, which has proved the correctness of Lemma 4. We also find that the IoT device, which is farther away from the AP, will need more computation resources to satisfy the accuracy requirements when the bandwidth resource is given. Thus, if we want to minimize the total computation resources, we need to allocate bandwidth resources reasonably. In this way, the IoT device farther away from the AP should be allocated with more bandwidth resources.

4.5 Optimal Bandwidth Resources Allocation

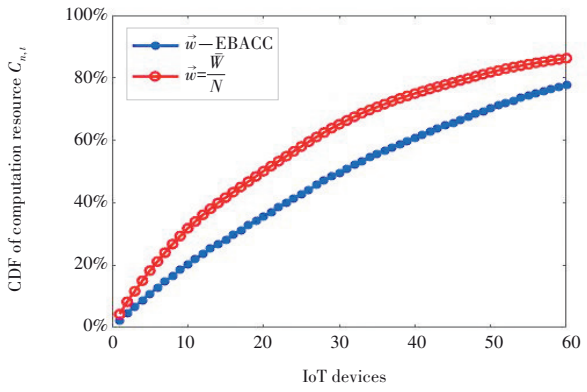
We compare two strategies of bandwidth allocation: 1) the optimal bandwidth resources allocation decided by EBACC; 2) allocating bandwidth resources equally to all IoT devices. As shown in Fig. 5(a), when the total computation resource



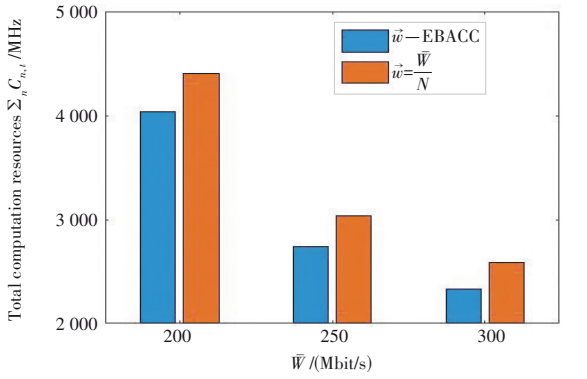
(a) Impact of $w_{n,t}$ on $D_{n,t}$

(b) Impact of $w_{n,t}$ on $C_{n,t}$

▲ **Figure 4. Impact of bandwidth resources allocated to IoT device on data set and computation resources**



(a) CDF of $C_{n,t}$



(b) Total computation resources requirement

CDF: Cumulative Distribution Function
EBACC: Efficient Bandwidth Allocation and Computation Configuration Algorithm
IoT: Internet of Things

▲Figure 5. CDF of computation resource requirement of each IoT device and total computation resources requirement under two situations: 1) optimal bandwidth resources allocation decided by EBACC; 2) allocating bandwidth resources equally to all IoT devices.

available is 3 000 MHz, the first strategy can use 77.85% of the total computation resource to satisfy the accuracy requirement of all IoT devices, but the second strategy needs 86.28%. It means that the proposed algorithm can significantly improve the efficiency of computing and bandwidth resources. Meanwhile, as shown in Fig. 5(b), the optimal bandwidth resource allocation can significantly reduce the demand of total computation resources of all IoT devices. Specifically, when the total bandwidth resource is $\bar{W}$ = 300 Mbit/s, the optimal bandwidth resource allocation can reduce the total computation resource requirement from 2 588.1 MHz to 2 335.4 MHz.

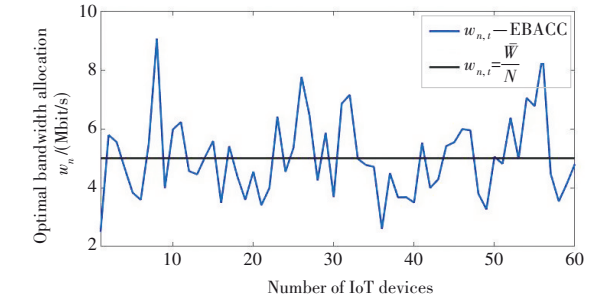
4.6 Relationship Between Optimal Bandwidth Allocation and Distance of IoT Devices

As shown in Figs. 6(a) and 6(b), we explore the relationship between the optimal bandwidth allocation decision $\boldsymbol{w}_t = \{\hat{w}_{1,t}, \hat{w}_{2,t}, \cdots, \hat{w}_{N,t}\}$ and the distances of all IoT devices $\boldsymbol{d}_t = \{d_{1,t}, d_{2,t}, \cdots, d_{N,t}\}$. Compared with the average allocation strategy, the optimal bandwidth allocation decision will be obviously affected by the accuracy requirement of IoT devices and the distance between the device and the AP. And it can be found that more bandwidth resources will be allocated to the IoT device farther away from the AP or with higher accuracy requirements.

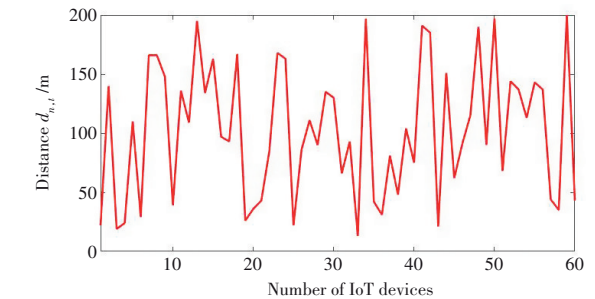
5 Conclusions

In this paper, we focus on the bandwidth allocation of AP and the computation resource management of the edge server to ensure the system accuracy can meet the industrial requirement. We formulate the bandwidth allocation and computation resource management problem for the industrial IoT as a cost minimization problem with a given accuracy requirement. Then, we analyze the relationship among the transmitted data, computation resources and system accuracy and then design an efficient algorithm to obtain the optimal computation resource allocation and communication resource management. Numerical experiment results demonstrate that the proposed algorithm EBACC can significantly reduce the number of total computation resources while satisfying the accuracy requirements of the industrial IoT.

For future work, we are going to consider the more general cases where IoT devices can choose different APs and edge servers to process their data and obtain a high-accuracy sys-



(a) Optimal bandwidth resource allocation decision $\vec{w}_t$



(b) Distances of IoT devices

EBACC: Efficient Bandwidth Allocation and Computation Configuration Algorithm
IoT: Internet of Things

▲Figure 6. Relationship between the optimal bandwidth allocation decision and distances of IoT devices

tem model. We will focus on the bandwidth allocation between multiple APs and multiple IoT devices, which would be more technically challenging.

References

[1] ADI E, ANWAR A, BAIG Z, et al. Machine learning and data analytics for the IoT [J]. Neural computing and applications, 2020, 32(20): 16205 – 16233. DOI: 10.1007/s00521-020-04874-y
[2] LEE J, STANLEY M, SPANIAS A, et al. Integrating machine learning in embedded sensor systems for Internet-of-Things applications [C]//IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, 2016: 290 – 294. DOI: 10.1109/ISSPIT.2016.7886051
[3] LIU Y K, CANDELL R, KASHEF M, et al. Dimensioning wireless use cases in Industrial Internet of Things [C]//14th IEEE International Workshop on Factory Communication Systems (WFCS). IEEE, 2018: 1 – 4
[4] LUO Y, DUAN Y, LI W F, et al. A novel mobile and hierarchical data transmission architecture for smart factories [J]. IEEE transactions on industrial informatics, 2018, 14(8): 3534 – 3546. DOI: 10.1109/TII.2018.2824324
[5] LIU Y K, KASHEF M, LEE K B, et al. Wireless network design for emerging IIoT applications: reference framework and use cases [J]. Proceedings of the IEEE, 2019, 107(6): 1166 – 1192. DOI: 10.1109/JPROC.2019.2905423
[6] SAVAZZI S, KIANOUSH S, RAMPA V, et al. A joint decentralized federated learning and communications framework for industrial networks [C]//IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). IEEE, 2020: 1 – 7. DOI: 10.1109/CAMAD50429.2020.9209305
[7] LONG N B, TRAN-DANG H, KIM D S. Energy-aware real-time routing for large-scale industrial Internet of Things [J]. IEEE Internet of Things journal, 2018, 5(3): 2190 – 2199. DOI: 10.1109/JIOT.2018.2827050
[8] JAGANNATH J, POLOSKY N, JAGANNATH A, et al. Machine learning for wireless communications in the Internet of Things: a comprehensive survey [J]. Ad hoc networks, 2019, 93: 101913. DOI: 10.1016/j.adhoc.2019.101913
[9] DING Z M, SHEN L F, CHEN H Y, et al. Energy-efficient relay-selection-based dynamic routing algorithm for IoT-oriented software-defined WSNs [J]. IEEE Internet of Things journal, 2020, 7(9): 9050 – 9065. DOI: 10.1109/JIOT.2020.3002233
[10] ZHAO R, WANG X J, XIA J J, et al. Deep reinforcement learning based mobile edge computing for intelligent Internet of Things [J]. Physical communication, 2020, 43: 101184. DOI: 10.1016/j.phycom.2020.101184
[11] KAUR K, GARG S, AUJLA G S, et al. Edge computing in the industrial Internet of Things environment: software-defined-networks-based edge-cloud interplay [J]. IEEE communications magazine, 2018, 56(2): 44 – 51. DOI: 10.1109/MCOM.2018.1700622
[12] ZHANG K, MAO Y M, LENG S P, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks [J]. IEEE access, 2016, 4: 5896 – 5907. DOI: 10.1109/access.2016.2597169
[13] HONG Z C, CHEN W H, HUANG H W, et al. Multi-hop cooperative computation offloading for industrial IoT-edge-cloud computing environments [J]. IEEE transactions on parallel and distributed systems, 2019, 30(12): 2759 – 2774. DOI: 10.1109/TPDS.2019.2926979
[14] GAO G J, XIAO M J, WU J, et al. Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud [J]. IEEE transactions on services computing, 2021, 14(6): 1702 – 1716. DOI: 10.1109/TSC.2019.2902549
[15] MA X, WANG S G, ZHANG S, et al. Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing [J]. IEEE transactions on cloud computing, 2021, 9(3): 968 – 980. DOI: 10.1109/TCC.2019.2903240
[16] YANG B, CAO X L, LI X F, et al. Mobile-edge-computing-based hierarchical machine learning tasks distribution for IIoT [J]. IEEE Internet of Things journal, 2020, 7(3): 2169 – 2180. DOI: 10.1109/JIOT.2019.2959035
[17] SUN C, SHRIVASTAVA A, SINGH S, et al. Revisiting unreasonable effectiveness of data in deep learning era [C]//IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 843 – 852. DOI: 10.1109/ICCV.2017.97
[18] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 770 – 778. DOI: 10.1109/CVPR.2016.90
[19] HUANG J, RATHOD V, SUN C, et al. Speed/accuracy trade-offs for modern convolutional object detectors [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 3296 – 3297. DOI: 10.1109/CVPR.2017.351
[20] STRUBELL E, GANESH A, MCCALLUM A. Energy and policy considerations for deep learning in NLP [C]//57th Annual Meeting of the Association for Computational Linguistics. ACL, 2019: 3645 – 3650
[21] QU Y B, LIU J J. Computation offloading for mobile edge computing with accuracy guarantee [C]//ACM Turing Celebration Conference. ACM, 2019: 1 – 5. DOI: 10.1145/3321408.3321582
[22] LIN J, CHEN W M, LIN Y J, et al. MCUNet: tiny deep learning on IoT devices [C]//Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems. NIPSF, 2020: 11711 – 11722
[23] CHEN X, JIAO L, LI W Z, et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. IEEE/ACM transactions on networking, 2016, 24(5): 2795 – 2808. DOI: 10.1109/TNET.2015.2487344
[24] CHIANG M, HANDE P, LAN T, et al. Power control in wireless cellular networks [J]. Foundations and trends in networking, 2008, 2(4): 381 – 533. DOI: 10.1561/1300000009
[25] XIAO M B, SHROFF N B, CHONG E K P. A utility-based power-control scheme in wireless cellular systems [J]. IEEE/ACM transactions on networking, 2003, 11(2): 210 – 221. DOI: 10.1109/TNET.2003.810314
[26] MIECH A, ZHUKOV D, ALAYRAC J B, et al. HowTo100M: learning a text-video embedding by watching hundred million narrated video clips [C]//IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019: 2630 – 2640. DOI: 10.1109/ICCV.2019.00272
[27] HUANG J W, BERRY R A, HONIG M L. Distributed interference compensation for wireless networks [J]. IEEE journal on selected areas in communications, 2006, 24(5): 1074 – 1084. DOI: 10.1109/JSAC.2006.872889
[28] BOYD S, VANDENBERGHE L. Convex Optimization [M]. Cambridge: UK: Cambridge University Press, 2004. DOI: 10.1017/cbo9780511804441

Biographies

HUANG Rui received his BS degree in computer science from Wuhan University of Technology, China. He is currently pursuing his master's degree with the School of Computer Science and Engineering, Central South University, China. His research interests include mobile edge computing and network optimization.

LI Huilin received his BS degree in mechanical design manufacture and automation from Shandong University, China. He is currently pursuing his master's degree with the School of Computer Science and Engineering, Central South University, China. His research interests include mobile edge computing and federal learning.

ZHANG Yongmin (zhangyongmin@csu.edu.cn) received his PhD degree in control science and engineering from Zhejiang University, China in 2015. From 2015 to 2019, he was a post-doctoral research fellow at the Department of Electrical and Computer Engineering, University of Victoria, Canada. He is currently a professor with the School of Computer Science and Engineering, Central South University, China. His research interests include resource management and optimization in wireless networks, smart grid, and mobile computing. He won the Best Paper Award of the IEEE PIMRC'12 and the IEEE Asia-Pacific Outstanding Paper Award 2018.

Ultra-Lightweight Face Animation Method for Ultra-Low Bitrate Video Conferencing

LU Jianguo[1,2], ZHENG Qingfang[1,2]
(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202301008
https://kns.cnki.net/kcms/detail/34.1294.TN.20230224.1425.002.html, published online February 24, 2023
Manuscript received: 2022-08-25

Abstract: Video conferencing systems face the dilemma between smooth streaming and decent visual quality because traditional video compression algorithms fail to produce bitstreams low enough for bandwidth-constrained networks. An ultra-lightweight face-animation-based method that enables better video conferencing experience is proposed in this paper. The proposed method compresses high-quality upper-body videos with ultra-low bitrates and runs efficiently on mobile devices without high-end graphics processing units (GPU). Moreover, a visual quality evaluation algorithm is used to avoid image degradation caused by extreme face poses and/or expressions, and a full resolution image composition algorithm to reduce unnaturalness, which guarantees the user experience. Experiments show that the proposed method is efficient and can generate high-quality videos at ultra-low bitrates.

Keywords: talking heads; face animation; video conferencing; generative adversarial network

Citation (IEEE Format): J. G. Lu and Q. F. Zheng, "Ultra-lightweight face animation method for ultra-low bitrate video conferencing," *ZTE Communications*, vol. 21, no. 1, pp. 64 - 71, Mar. 2022. doi: 10.12142/ZTECOM.202301008.

1 Introduction

During the COVID-19 Pandemic, video conferencing systems have become indispensable tools for individuals to keep in touch with friends and for enterprises and organizations to connect with customers. Inside these systems, video compression technologies play critical roles in the efficient representation and transportation of video data. Great progress has been achieved in past years in representing high-fidelity videos with low bitrates; e.g., the high-efficiency video coding (HEVC)[1] was designed with the goal of allowing video content to have a data compression ratio up to 1 000:1. However, video conferencing systems still face the dilemma between smooth streaming and decent visual quality because current video compression technologies fail to produce bitstreams low enough for bandwidth-constrained networks due to a large number of concurrent users.

Recently, some novel talking-head video compression methods[2-5] based on face animation have been proposed, which can significantly cut down the bandwidth usage of video conferences. These face animation methods usually consist of two parts: encoder and decoder. The encoder is a motion extractor to derive a compact motion feature representation from the driving video frame, and the decoder is an image generator to synthesize photorealistic images according to the motion feature. Due to its extreme compactness, the extracted face feature can be used to reduce the bandwidth of video conferences and hence improve user experience in bandwidth-constrained networks. However, most of the talking-head video compression methods are too complicated to run in real time without the support of high-end graphics processing units (GPUs), let alone on mobile devices. For example, the model size of the First Order Motion Model (FOMM)[6] is 355 MB and the computation complexity is 121 G multiply-accumulate operations (MACs). Aiming at practical applications, we propose an ultra-lightweight motion extractor to obtain effective motion representations from the driving video and an animation generator to synthesize high-quality face videos accordingly.

We find out that the face animation method may sometimes fail, which is usually caused by extreme head poses and/or facial expressions. To tackle the problem, we propose an efficient visual quality evaluation method to reject the synthesized images that are visually unacceptable. We also notice that only displaying face without context regions looks unnatural and weird to users. To cope with it, we composite full-resolution images by stitching face regions with other body parts and backgrounds. These two mechanisms effectively prevent user experience degradation during a conference.

Our main contributions are as follows:

• An ultra-lightweight motion extraction algorithm is proposed to derive effective facial motion features from driving videos, which is efficient enough to run on mobile devices without high-end GPUs.

• An efficient visual quality evaluation algorithm is proposed to select visually acceptable generated images and an image composition algorithm to generate full-resolution videos, which ensures consistent and natural user experience during conferences.

• A practical video conferencing system is built to integrate the best parts of face-animation-based methods and traditional video-compression-based methods, which significantly reduces uplink bandwidth usage and ensures decent user experience even when the network bandwidth is constrained.

2 Related Work

Due to the space limitation, we only review previous works about face animation and deep video compression that are most related to ours.

2.1 Face Animation

Face animation is an image-to-image translation task, which transfers the talking-head motion of a person in an image to persons in other images. The former image is called the driving image, while the latter image is called the source image. Face animation has become a popular topic since the generative adversarial network (GAN) [7] was proposed by GOODFELLOW et al. Most recently published face animation methods can synthesize photo-realistic images with the help of GANs.

Some works[8 - 12] were proposed to solve the face animation task with the prior knowledge of the 3D Morphable Model (3DMM) [13]. However, the traditional 3D-based works[8 - 10] failed to render details of talking heads, such as hair, teeth and accessories. Ref. [11] allowed fine-scale manipulation of any facial input image into a new expression while preserving its identity with the help of a conditional GAN. To improve the realism of the rendering, Ref. [12] designed a novel space-time GAN to predict photorealistic video frames from the modified 3DMM directly.

Contrary to 3D-based models, 2D-based models synthesize talking heads directly without any prior knowledge of 3DMM. They can be classified into warping-based models and warping-free models.

Warping-free models[14 - 19] directly synthesize images without any warping. Few-shot vid2vid[16] learned to transform landmark positions into realistically looking personalized photographs with the help of meta-learning. Ref. [19] decomposed a person's appearance into a pose-dependent coarse image and a pose-independent texture image. LI-Net[20] decoupled the face landmark image into pose and expression features and reenacted those attributes separately to generate identity-preserving faces with accurate expressions and poses.

Warping-based methods[21 - 25] predicted dense motion fields to warp the feature maps extracted from the source images and inpaint the warped feature maps to generate photorealistic images. X2Face[22] used an encoder-decoder architecture to learn the latent embedding to encode pose and expression and recover the dense motion fields from it. Many works attempted to predict the dense motion field from sparse object keypoints. The key to those methods is how to represent motions with sparse object keypoints. Monkey-Net[23] was proposed to learn pure keypoints to describe motions in an unsupervised manner. Although it cannot describe subtle motions, Monkey-Net provided a strong baseline for further improvements. FOMM[6] represented sparse motion with some keypoints along with local affine transformations. Motion representations for articulated animation (MRAA)[24] defined the motion with regions using the motion estimation based on principal component analysis (PCA), rather than keypoints, to describe locations, shapes and pose. The thin-plate spline (TPS) motion model[25] estimated thin-plate spline motion to produce a more flexible optical flow. Ref. [5] extended the baseline to 3D optical flows to produce 3D deformations. The above mentioned methods extracted compact motion representations, which showed great potential in lowering the bitrate of video conferencing.

2.2 Deep Learning-Based Video Compression

For decades, researchers have made great efforts to transmit higher quality videos with lower bitrates. Recently several approaches based on deep learning were explored.

For general-purpose video compression, some works[26 - 27] attempted to reduce the bandwidth by making a balance between the cost of transferring the region of interest (ROI) and background. Compared to traditional codecs, such methods can achieve better visual quality with the same bitrate. Other works[28 - 29] focused on enhancing the visual quality of low bitrate videos by image super-resolution and image enhancement.

For the compression of talking-head videos, great progress has been achieved. In Ref. [30], the encoder detected and transmitted keypoints representing the body pose and the face mesh information, and the receiver displayed the motion in the form of puppets. However, this method failed to produce photorealistic images. Inspired by the promising results achieved by face animation models, many works demonstrated the effectiveness of video compression based on face animation. VSBNet[3] reconstructed original frames from face landmarks with a low bitrate of around 1 kB/s. Ref. [5] proposed a neural talking-head video synthesis model and set up a video conferencing system that achieves the same visual quality as the commercial H. 264 standard with only one-tenth of the bandwidth. Ref. [2] introduced an adaptive intra-refresh scheme to address the problem of reconstruction quality that might rapidly degrade due to the loss of temporal correlation as frames get farther away from the initial one. Ref. [4] evaluated the advantages and disadvantages of several deep generative adversarial approaches and designed a mobile-compatible architecture that can run at 19 f/s on iPhone 8. However, those methods can hardly run in real time without the support of high-end GPUs. What's more, they could only generate near-

frontal faces, looking unnatural and weird when faces were not near-frontal. In this paper, we specifically focus on improving the efficiency and visual quality of video compression based on face animation.

3 Proposed Ultra-Lightweight Face Animation Method

3.1 Overview

The overall pipeline of our video conference system is shown in Fig. 1. Each user provides an avatar image to the system and uses its animation during a conference for ensuring privacy and elegant presence. When the system starts running, videos of users are captured and the face region in each video frame is cropped out by the face detection algorithm. Face images are then encoded by the keypoint detector and represented as the keypoints described in Section 3.2. Before the encoded data are sent out, the visual quality of the face image that will be reconstructed by a decoder according to these keypoints is evaluated to prevent unnatural results. It is highlighted here that the visual quality evaluation method in Section 3.3 requires no actual reconstruction of the face image but executes on encoded data, for the sake of efficiency.

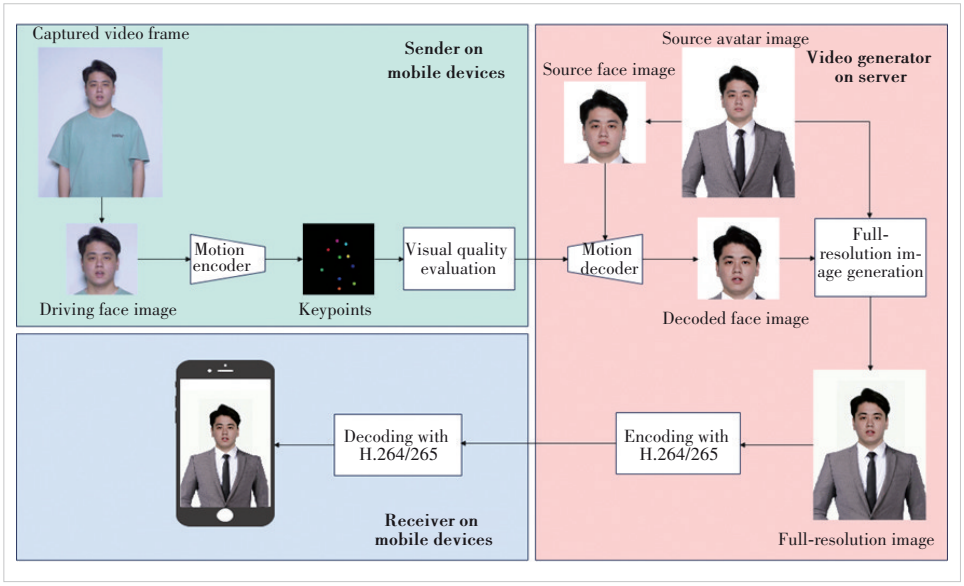
Upon receiving the encoded keypoint data from the sender, the conference server calls the image generator to synthesize the face image animated from the keypoints, as described in Section 3.2. The decoded face image replaces the face region in the avatar image by our method in Section 3.4 to create a full-resolution video frame, which is then encoded by H.264 or HEVC and sent to the receiver. The receiver simply decodes the video stream and displays it on the screen, which can usually take advantage of the hardware accelerator in the device's chip.

With the prevalence of mobile phones, the demand for running video conferencing on mobile devices is growing. In most commercial video conference systems, mobile devices account for a significant portion of all terminals. For better compatibility with existing commercial video conference systems, our system and algorithms here are intentionally designed to make the sender/receiver module deployable on mobile devices and to keep their computational burdens to a minimum, thus reducing power consumption and extending the working time of mobile devices.

3.2 Model Distillation

Giving a source image $\boldsymbol{S}$ of the target person, a driving video can be denoted as $\{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \cdots, \boldsymbol{D}_N\}$, where $\boldsymbol{D}_i$ is the i-th frame in the sequence and N is the total number of frames in the video. The output images can be denoted as $\{\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \cdots, \boldsymbol{O}_N\}$, where $\boldsymbol{O}_i$ is the i-th frame of the output sequence. The output $\boldsymbol{O}_i$ shares the same identity with $\boldsymbol{S}$ and the same face motions with $\boldsymbol{D}_i$. We adopt the face animation model similar to FOMM, which consists of a keypoint detector $\boldsymbol{K}$ (encoder) and a generator $\boldsymbol{G}$ (decoder). First, face landmarks are estimated from $\boldsymbol{S}$ and $\boldsymbol{D}_i$ separately by $\boldsymbol{K}$, whose locations serve as the sparse motion information. Second, dense motion fields and occlusion maps are predicted by $\boldsymbol{G}$. Finally, $\boldsymbol{G}$ warps the feature map extracted from $\boldsymbol{S}$ with the dense motion fields and the warped feature map is masked by the occlusion maps to generate the output image $\boldsymbol{O}_i$. Following the idea of FOMM, we extract 10 keypoints and their corresponding Jacobian matrices from the face image.

We design our model to be lightweight and can generate an image with excellent visual quality. For the decoder, we adopt the same architecture as the generator model in FOMM but cut down the channels of the model by half. We denote the simplified generator as $\boldsymbol{G}_{\text{sim}}$. For the encoder, we replace the hourglass network in FOMM, which brings about high computational cost, with a greatly simplified version of MobileNetV2[31]. However, it is very difficult to train the proposed model from scratch since the training process often fails to converge. We come up with a training strategy described as fol-



▲ **Figure 1. Proposed video conference system consists of three parts: the sender on mobile devices, video generator on servers, and receiver on mobile devices. In the encoder part, the motion encoder extracts keypoints from the driving images. The feature-based image quality evaluation filters out unnatural images. The decoder synthesizes images from the keypoints and reconstructs full-resolution images, which are encoded by H.264 or H.265 and sent to the receiver. The receiver decodes the video stream and shows it on the phone screen**

lows to solve the problem.

1) Step 1: model distillation. We use the original encoder K_{fomm} in FOMM as the teacher model and our proposed encoder K_{pro} as the student model. The loss function consists of distillation loss L_{dis} and equivariance loss L_{eq}, which can be written as Eq. (1).

$$L_1 = L_{\text{dis}} + L_{\text{eq}} = \left| K_{\text{pro}}(I) - K_{\text{fomm}}(I) \right| + \left| K_{\text{pro}}(T(I)) - T(K_{\text{pro}}(I)) \right|, \quad (1)$$

where I is the training sample and T is a thin plane spline deformation. The distillation loss ensures that the student encoder extracts the same motion representation as the teacher encoder. And the equivariance loss ensures the consistency of the motion representation when random geometric transformations are applied to the images.

2) Step 2: iterative model pruning and distillation. Since the encoder has to extract motion representation from every video frame, it should be as lightweight as possible to reduce computational costs. In our attempt to further simplify the encoder, we find out most of the complexity comes from the last several convolutional layers. Therefore, we drop the last convolutional layer in the encoder model and retrain it following Step 1. This step can be repeated several times until we obtain K_{best} that strikes a balance between the model complexity and accuracy.

3) Step 3: generator fine-tuning. Due to the simplification made to the generator, we train the simplified generator G_{sim} along with the keypoint detector K_{fomm} of the original FOMM to make a good initialization of G_{sim}.

4) Step 4: overall fine-tuning. Once the encoder models K_{best} and G_{sim} are determined, we fine-tune K_{best} and G_{sim} accordingly in an unsupervised manner. Finally, K_{best} and G_{sim} act as the encoder and the decoder in our system respectively.

3.3 Visual Quality Evaluation

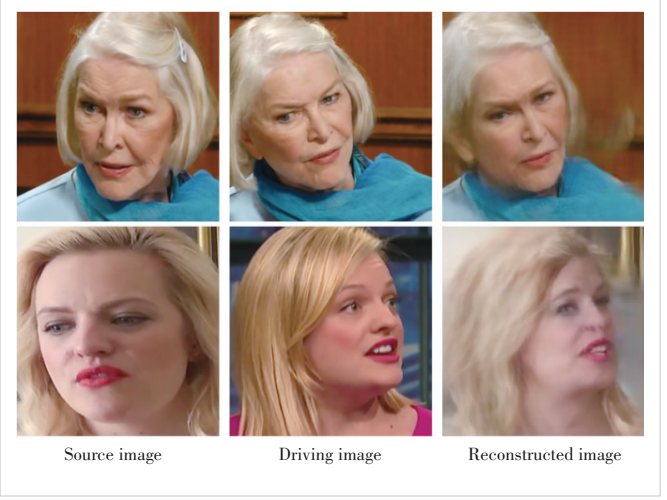
Although video conferences based on face animation can result in a very high video compression rate, the visual quality of a reconstructed image may sometimes degrade in the following two cases (Fig. 2). First, due to current algorithmic limitations, most of the face animation models may generate inaccurate expressions and visual artifacts on faces with large poses and/or extreme expressions. Second, with the increase of the frame distance, the temporal correlation weakens, and hence the quality of generated video deteriorates. This phenomenon becomes particularly obvious when faces are occluded. The degraded image brings inconsistent experience to users. In order to alleviate the problem, Ref. [2] introduced an adaptive intra-refresh scheme using multiple source frames. Before sending the features to the decoder, the sender reconstructs the image first and evaluates the generated image to avoid degraded images. However, this scheme not only incurs large computational costs which makes it impossible to run it on mobile devices, but also leads to significant time delay at the receiving end. What's more, frequent scene switching also requires the system's frequent sending of source frames, making the system lose its advantage of reducing video bandwidth.

We propose here an adaptive degraded frame filter method by an efficient image quality evaluation algorithm directly based on the extracted features. We find out that when a large head pose and/or extreme facial expression happens, most of the regions in the generated image are inpainted by the generator, which degrades the image quality. The difference between the driving image and the source image can be measured by analyzing the dense motion field, which is predicted from the sparse motion field in our setting. Therefore, instead of using the decoder to synthesize the generated image, we decide to evaluate image quality based on the relative motion. The loss L_2 in the algorithm can be formulated as follows.

$$L_2 = \alpha \sum_{i=0}^{10} \left\| v_{1i} - v_{2i} \right\| + \beta \sum_{i=0}^{10} \left\| J_{1i} J_{2i}^{-1} \right\|, \quad (2)$$

where v_{1i} is the value of the i-th keypoint in the first frame, v_{2i} is the value of the i-th keypoint in the second frame, J_{1i} is the Jacobian of the i-th keypoint in the first frame, J_{2i} is the Jacobian of the i-th keypoint in the second frame, and hyperparameters α and β control the weight of each part. In our experiments, we set the hyperparameters to 2 and 1 respectively.

In the proposed scheme, the balance between image quality and robustness is controlled by a threshold τ. Although the identity of the people in the driving images and the source image are the same, the two images may look different. For better visual quality, we adopt a relative motion transfer method, as described in Ref. [6]. We first find a driving image that has a



▲ **Figure 2. Examples of face animation failure. The first row shows a result caused by large-pose; the face area becomes blurred and there are some artifacts on the hair of the woman. The second row shows a degraded image caused by weak temporal correlation and the reconstructed image looks terrible and weird**

similar pose to the source image, which is called the initial image D_I. Then, we extract keypoints from the source image S and the initial image D_I, which can be denoted as Ks and K_I. The source keypoints are sent to the receiver. For every frame D_t, we estimate keypoints K_t from the frame, and compare the relative motion between K_t and K_s and that between K_I and K_s. If the former is smaller, we set this driving keypoint as an initial image. Finally, we compare the relative motion between K_t and K_I with the threshold τ. If the former is smaller, it means the relative motion is suitable for robust image generation. The relative motion is sent to the server. If the latter is smaller, the default motion is sent to avoid freezing in video streams. The default keypoints can be motions of some natural expressions, such as blinking and smiling. In this way, the degraded frames are replaced by frames of natural expressions. Compared to the method proposed in Ref. [2], our method can greatly reduce the computation cost at the sender and the delay at the receiver.

3.4 Full-Resolution Image Composition

The face animation described above cannot be directly used in video conferences due to two facts. Face animation cannot synthesize face images with a size up to video resolution (at least 1 280×720) because computational complexity grows exponentially with the image size. Also, only displaying the facial region on the screen without other body parts such as the neck and shoulder looks unnatural and weird. In order to make our face animation method applicable, instead of generating full-resolution images, we propose to generate a facial region with a size of no more than 384×384 and stitch it with other body parts and background regions in the source frame to form a full-resolution image. The problem is that there will be a sharp blocky artifact between the head region and body region because the head region moves while the body region may remain stationary. We find that the keypoints spread over the talking-head area and each keypoint is responsible for the local transformation of its neighborhood. To reduce the artifact, we fix the keypoints related to the shoulder part. As a result, the dense motion field predicted by the generator will stay stationary near the shoulder region and have a smooth transition from the head region to the shoulder region, which makes the composite image look more natural. We show the example images in Fig. 3 for comparison.

4 Experiments

4.1 Implementation Details

1) Datasets. We train and evaluate our face animation model on the VoxCeleb dataset and an in-house dataset. VoxCeleb[32] is a dataset of interview videos of different celebrities. We crop the videos and resize them to 256×256 for a fair comparison with the original FOMM and 384×384 for the generation of high-resolution images according to the bounding boxes of faces. The in-house dataset consists of 4 124 Chinese people videos collected from the Internet and is used to reduce bias towards Western people. We fine-tune our model on the in-house dataset to make better adaptations to Chinese.

2) Evaluation metrics. We evaluate the models using the L1 error, average keypoint distance (AKD) and average Euclidean distance (AED). The L1 error is the mean absolute difference between pixel values in the reconstructed images and the ground-truth images, which measures the reconstruction accuracy. AKD and AED stand for semantic consistency. AKD is the average distance between the face landmarks extracted from the ground-truth images and the reconstructed images respectively by the face landmark detector[33], which measures the pose difference between the two images. AED measures identity preservation, which is the L2 distance of the corresponding features extracted by a pre-trained re-identification network[34].

3) Hardware. In our video conference system, we implement a conferencing APP on a ZTE A30 Ultra mobile phone with Snapdragon 888 System on a Chip (SoC) and conferencing server software on a computer with Nvidia Tesla V100 GPU.

▲**Figure 3. Qualitative comparisons with state-of-the-art methods. The first three rows are images from the VoxCeleb dataset and the following four rows are images from our in-house dataset. Our method produces competitive results**

4.2 Comparisons with FOMM

1) Efficiency of the proposed face animation algorithm

First, we compare our encoder, i.e., the face motion extractor, with that of the original FOMM. We convert the encoder to the mobile neural network (MNN)[35] model and calculate the model size. As listed in Table 1, our encoder model is only 600 kB in size with theoretical computation complexity of 14.62 M MAC, both of which are about 1% of FOMM. Our encoder processes every frame in 3.5 ms on Snapdragon 888, which is 16.3 times faster than FOMM.

Second, we compare our decoder, i.e., the generator to synthesize a 384×384-resolution face image, also with FOMM. For the generator, we convert the model to TensorRT[36] model and calculate the model size. As listed in Table 1, our decoder model is 81.77 MB in size with theoretical computation complexity of 31.42 G MAC, and these two values are 26.0% and 27.3% of FOMM respectively. Our encoder runs in 5 ms on Tesla V100, which is 4 times faster than FOMM.

2) Effectiveness of the proposed face animation algorithm

We compare the visual quality of face images generated by our method with other face animation methods. For quantitative comparison, we evaluate our model with existing studies on the VoxCeleb dataset for an image generation task. For a fair comparison, we generate images with the resolution of 256×256. The first frame of each test video is set as the source image, while the subsequent frames are set as the driving images. Evaluation metrics are computed for every frame and our result is the mean value of all frames. The results are summarized in Table 2, which clearly shows the proposed method outperforms X2Face and Monkey-Net. Compared to FOMM, our method can generate competitive results, even though our model is much lighter than FOMM. For a qualitative comparison, we list some example images in Fig. 3 for visual comparisons.

▼ Table 1. Efficiency comparison between our face animation method and FOMM

Model		MAC	Parameters/M	Model size/MB	Inference time/ms
Encoder	FOMM	1 280 M	14.21	55.54	57
	Ours	14.62 M	0.16	0.60	3.5
Decoder	FOMM	120.70 G	45.56	299.10	20
	Ours	31.42 G	16.16	81.77	5

FOMM: First Order Motion Model MAC: multiply-accumulate operation

▼ Table 2. Visual quality comparison among different face animation methods on VoxCeleb dataset

	L1	AKD	AED
X2Face[22]	0.078	7.69	0.405
Monkey-Net[23]	0.049	1.89	0.199
FOMM[6]	0.041	1.27	0.134
Ours	0.043	1.37	0.147

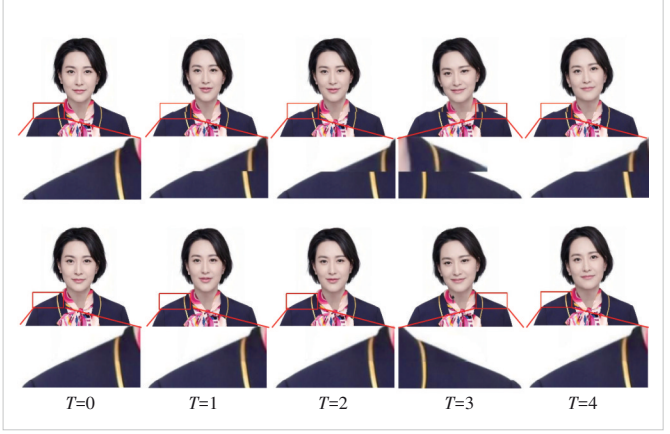
AED: average Euclidean distance FOMM: First Order Motion Model
AKD: average keypoint distance

4.3 Results of Full-Resolution Image Generation

The avatar images provided by a user are usually not face-only, but with other upper body parts. When head regions in the avatar images are cropped and animated by our method, they should be stitched back into original images to form new images with predefined resolutions, e.g., 1 280×720. Special treatment should be given to the point where the head region and body region connect because these regions move non-rigidly and disproportionately. As shown in the top two rows in Fig. 4, simply replacing the head region in an avatar image with a new animated head region will result in visual discontinuities. As comparisons, the bottom two rows show results of the proposed method described in Section 3.4. Our method successfully eliminates discontinuities and makes whole images visually natural.

4.4 Ultra-Low Bitrate Video Conference

As described in Section 3.1, our video conference system is comprised of server software running on the cloud server and application software, with the sender module and receiver module, running on the mobile phone. The most important difference between our sender module and those inside other video conference systems is we encode captured videos into compact keypoint motion information, rather than traditional H.264 or HEVC streams, which greatly cuts down the uplink bandwidth usage. For example, when encoded in H.264, 720 p conference videos are typical of bitrates between 1 Mbit/s and 2 Mbit/s. By comparison, each video frame is encoded by our sender module as 10 keypoint information, each of which includes a position (2 floating points) and a Jacobian matrix (4 floating points). We empirically determine the half precision floating point format (FP16) is enough for data representation and thus reaches the bitrate of 6×16×10×30=28.8 kbit/s, which is only less than 3% of H.264 encoding. We note the



▲ Figure 4. Results of full-resolution image generation. The first row shows images generated by simply replacing the head region in the source image with the new animated head region. The third row shows image results by our method in Section 3.4. In the second and fourth rows, connections between head regions and body regions are zoomed in for clearer comparison

keypoint information can be compressed by the entropy encoder for further bandwidth usage saving.

In our real-world user studies, reducing the uplink bitrate can greatly improve the conference user experience. For one thing, since wireless bandwidth is not evenly allocated for uplink and downlink data transportation, a smaller uplink bitrate can result in less congestion and faster upward transmission. For another thing, more aggressive schemes can be applied when Forward error correction (FEC) is used to tackle data loss in transmission, leading to less data retransmission, which brings about lower remote interaction latency and more real-time engagement.

The server software in our system runs on a cloud server with Nvidia GPUs because the image generator in face animation is much more computationally expensive than the keypoint extractor, as demonstrated in Section 4.1. Although our simplified image generator can be deployed on some flagship mobile phones with powerful GPUs, we choose server-side deployment to make our application software lightweight enough to run on most mobile phones and consume less power to extend working time, which is also critical to user experience.

5 Conclusions

In this paper, we propose a face-animation-based method to greatly reduce bandwidth usage in video conferences, compressing face video frames by using only 60 FP16 data to represent the face motion. We design an ultra-lightweight face motion extraction algorithm that runs on mobile devices, as well as an efficient visual quality evaluation algorithm and a full-resolution image composition algorithm to ensure consistent and natural user experience. We also build a practical system to enable user communication using animated avatars. Experimental results demonstrate the efficiency and effectiveness of our methods and their superiority over previous studies. However, one limitation of our current work is that our method is only applicable to upper-body videos. A full-body animation method should be our next work to cover more real-world scenarios. Another improvement to our system will be saving downlink bandwidth by reconstructing videos on mobile devices, which requires further research in GAN acceleration to meet real-time constraints on mobile devices.

References

[1] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. IEEE transactions on circuits and systems for video technology, 2012, 22(12): 1649 – 1668. DOI: 10.1109/TCSVT.2012.2221191

[2] KONUKO G, VALENZISE G, LATHUILIÈRE S. Ultra-low bitrate video conferencing using deep image animation [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 4210 – 4214. DOI: 10.1109/ICASSP39728.2021.9414731

[3] FENG D H, HUANG Y, ZHANG Y W, et al. A generative compression framework for low bandwidth video conference [C]//IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2021: 1 – 6. DOI: 10.1109/ICMEW53276.2021.9455985

[4] OQUAB M, STOCK P, GAFNI O, et al. Low bandwidth video-chat compression using deep generative models [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2021: 2388 – 2397. DOI: 10.1109/CVPRW53098.2021.00271

[5] WANG T C, MALLYA A, LIU M Y. One-shot free-view neural talking-head synthesis for video conferencing [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 10034 – 10044. DOI: 10.1109/CVPR46437.2021.00991

[6] SIAROHIN A, LATHUILIÈRE S, TULYAKOV S, et al. First order motion model for image animation [J]. Advances in neural information processing systems. 2019, 32: 7135 – 7145

[7] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets [J]. Advances in neural information processing systems. 2014, 27: 2672 – 2680

[8] VLASIC D, BRAND M, PFISTER H, et al. Face transfer with multilinear models [J]. ACM transactions on graphics, 2005, 24(3): 426 – 433. DOI: 10.1145/1073204.1073209

[9] DALE K, SUNKAVALLI K, JOHNSON M K, et al. Video face replacement [J]. ACM transactions on graphics, 2011, 30(6): 1 – 10. DOI: 10.1145/2070781.2024164

[10] THIES J, ZOLLHÖFER M, STAMMINGER M, et al. Face2Face: real-time face capture and reenactment of RGB videos [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 2387 – 2395. DOI: 10.1109/CVPR.2016.262

[11] NAGANO K, SEO J, XING J, et al. PaGAN: real-time avatars using dynamic textures [J]. ACM transactions on graphics, 2018, 37(6): 1 – 12. DOI: 10.1145/3272127.3275075

[12] KIM H, GARRIDO P, TEWARI A, et al. Deep video portraits [J]. ACM transactions on graphics (TOG), 2018, 37(4): 1 – 14. DOI: 10.1145/3197517.3201283

[13] BLANZ V, VETTER T. A morphable model for the synthesis of 3D faces [C]//26th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1999: 187 – 194. DOI: 10.1145/311535.311556

[14] BURKOV E, PASECHNIK I, GRIGOREV A, et al. Neural head reenactment with latent pose descriptors [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 13783 – 13792. DOI: 10.1109/CVPR42600.2020.01380

[15] OLSZEWSKI K, LI Z M, YANG C, et al. Realistic dynamic facial textures from a single image using GANs [C]//IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 5439 – 5448. DOI: 10.1109/ICCV.2017.580

[16] SONG Y, ZHU J W, LI D W, et al. Talking face generation by conditional recurrent adversarial network [C]//Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI, 2019: 919 – 925. DOI: 10.24963/ijcai.2019/129

[17] YU J H, LIN Z, YANG J M, et al. Generative image inpainting with contextual attention [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 5505 – 5514. DOI: 10.1109/CVPR.2018.00577

[18] ZAKHAROV E, SHYSHEYA A, BURKOV E, et al. Few-shot adversarial learning of realistic neural talking head models [C]//IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020: 9458 – 9467. DOI: 10.1109/ICCV.2019.00955

[19] ZAKHAROV E, IVAKHNENKO A, SHYSHEYA A, et al. Fast bi-layer neural synthesis of one-shot realistic head avatars [C]//European Conference on Computer Vision. Springer, 2020: 524 – 540. DOI: 10.1007/978-3-030-58610-2_31

[20] LIU J, CHEN P, LIANG T, et al. Li-Net: large-pose identity-preserving face reenactment network [C]//IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2021: 1 – 6. DOI: 10.1109/ICME51207.2021.9428233

[21] ZHAO R Q, WU T Y, GUO G D. Sparse to dense motion transfer for face image animation [C]//IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). IEEE, 2021: 1991 – 2000. DOI: 10.1109/ICCVW54120.2021.00226

[22] WILES O, KOEPKE A S, ZISSERMAN A. X2Face: a network for controlling face generation using images, audio, and pose codes [C]//European Conference on Computer Vision. Springer, 2018: 690 – 706. DOI: 10.1007/978-3-030-01261-8_41
[23] SIAROHIN A, LATHUILIÈRE S, TULYAKOV S, et al. Animating arbitrary objects via deep motion transfer [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 2372 – 2381. DOI: 10.1109/CVPR.2019.00248
[24] SIAROHIN A, WOODFORD O J, REN J, et al. Motion representations for articulated animation [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 13648 – 13657. DOI: 10.1109/CVPR46437.2021.01344
[25] ZHAO J, ZHANG H. Thin-plate spline motion model for image animation [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2022: 3647 – 3656. DOI: 10.1109/CVPR52688.2022.00364
[26] AGUSTSSON E, TSCHANNEN M, MENTZER F, et al. Generative adversarial networks for extreme learned image compression [C]//IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020: 221 – 231. DOI: 10.1109/ICCV.2019.00031
[27] KAPLANYAN A S, SOCHENOV A, LEIMKÜHLER T, et al. DeepFovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos [J]. ACM transactions on graphics, 2019, 38(6): 1 – 13. DOI: 10.1145/3355089.3356557
[28] LU G, OUYANG W L, XU D, et al. Deep kalman filtering network for video compression artifact reduction [C]//European Conference on Computer Vision. Springer, 2018: 591 – 608. DOI: 10.1007/978-3-030-01264-9_35
[29] GUO Y H, ZHANG X, WU X L. Deep multi-modality soft-decoding of very low bit-rate face videos [C]//28th ACM International Conference on Multimedia. ACM, 2020: 3947 – 3955. DOI: 10.1145/3394171.3413709
[30] PRABHAKAR R, CHANDAK S, CHIU C, et al. Reducing latency and bandwidth for video streaming using keypoint extraction and digital puppetry [C]//Data Compression Conference (DCC). IEEE, 2021: 360. DOI: 10.1109/DCC50243.2021.00057
[31] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 4510 – 4520. DOI: 10.1109/CVPR.2018.00474
[32] NAGRANI A, CHUNG J S, ZISSERMAN A. VoxCeleb: a large-scale speaker identification dataset [C]//18th Annual Conference of the International Speech Communication Association. ISCA, 2017: 2616 – 2620. DOI: 10.21437/interspeech.2017-950
[33] BULAT A, TZIMIROPOULOS G. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230 000 3D facial landmarks) [C]//IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 1021 – 1030. DOI: 10.1109/ICCV.2017.116
[34] AMOS B, LUDWICZUK B, SATYANARAYANAN M. Openface: a general-purpose face recognition library with mobile applications: CMU-CS-16-118 [R]. USA: School of Computer Science, Carnegie Mellon University, 2016. DOI:10.13140/RG.2.2.26719.07842
[35] JIANG X, WANG H, CHEN Y, et al. MNN: a universal and efficient inference engine [C]//Third Conference on Machine Learning and Systems. MLSys, 2020, 2: 1 – 13. DOI: 10.48550/arXiv.2002.12418
[36] NVIDIA. NVIDIA TensorRT [EB/OL]. [2022-02-22]. https://developer.nvidia.com/tensorrt

Biographies

LU Jianguo received his BS and MS degrees from Huazhong University of Science and Technology, China in 2017 and 2020 respectively. After graduation, he has been working at ZTE Corporation. His research interests include computer vision, artificial intelligence and augmented reality.

ZHENG Qingfang (zheng.qingfang@zte.com.cn) received his BS degree from Shanghai Jiao Tong University, China in 2002, and PhD degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences in 2008. He is now the chief scientist of cloud video product and deputy director of the Video Technology Committee at ZTE Corporation. His current research interests include video communication, computer vision and artificial intelligence.

Adaptive Load Balancing for Parameter Servers in Distributed Machine Learning over Heterogeneous Networks

CAI Weibo[1], YANG Shulin[1], SUN Gang[1], ZHANG Qiming[2], YU Hongfang[1]
(1. University of Electronic Science and Technology of China, Chengdu 611731, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202301009
https://kns.cnki.net/kcms/detail/34.1294.TN.20230301.1800.001.html, published online March 2, 2023
Manuscript received: 2022-04-11

Abstract: In distributed machine learning (DML) based on the parameter server (PS) architecture, unbalanced communication load distribution of PSs will lead to a significant slowdown of model synchronization in heterogeneous networks due to low utilization of bandwidth. To address this problem, a network-aware adaptive PS load distribution scheme is proposed, which accelerates model synchronization by proactively adjusting the communication load on PSs according to network states. We evaluate the proposed scheme on MXNet, known as a real-world distributed training platform, and results show that our scheme achieves up to 2.68 times speed-up of model training in the dynamic and heterogeneous network environment.

Keywords: distributed machine learning; network awareness; parameter server; load distribution; heterogeneous network

Citation (IEEE Format): W. B. Cai, S. L. Yang, G. Sun, et al., "Adaptive load balancing for parameter servers in distributed machine learning over heterogeneous networks," *ZTE Communications*, vol. 21, no. 1, pp. 72 – 80, Mar. 2023. doi: 10.12142/ZTECOM.202301009.

1 Introduction

Machine learning is widely used in many fields such as image classification[1], speech recognition[2], and natural language processing[3]. With the continuous increase in training data and the model size, the huge time cost of single-machine training is unacceptable to users. Therefore, distributed machine learning (DML) based on multi-machine parallelism has drawn more and more attention. Usually, distributed training is carried out within a single cluster, since it is considered that networks with limited bandwidth and complex and changeable states across clusters will seriously slow down the communication process of DML. However, due to the limitations of data privacy protection[4], data aggregation among clusters for model training is not allowed in some cases. In addition, with the proposal of Computing First Network[5-6], DML model training based on the integrated computing power of the whole network gradually shows great application prospects. Based on the consideration mentioned above, DML in heterogeneous networks across clusters has great research value.

There are mainly two communication architectures for DML: one is a centerless architecture, represented by AllReduce[7-8], and the other is a centered architecture, represented by a parameter server (PS) architecture[9-11]. In the PS architecture, there are usually two types of nodes in the DML system: the worker responsible for model training and the server for model aggregation and parameter update. During a typical training iteration of data parallelism and synchronous update mode, workers send model gradients uniformly after completing the training based on the local model and data, and the server receives the model gradient from workers. Thereafter, the model aggregation operation is performed to generate a global model, and the global model is sent to workers. Workers immediately replace the local model after receiving the global model from the server and start a new training iteration.

In this process, since the data from all workers need to be aggregated on the server, servers with limited bandwidth resources could become the bottleneck of transmission, which is also an inherent problem of the PS architecture[12]. In order to tackle this problem, a traditional solution[13] is to increase the number of servers and let multiple servers share the heavy communication load. Since the load distribution of each server usually follows the principle of fairness, this scheme has an

This research was partially supported by the computing power networks and new communication primitives project under Grant No. HC-CN-2020120001, the National Natural Science Foundation of China under Grant No. 62102066, and Open Research Projects of Zhejiang Lab under Grant No. 2022QA0AB02.

ideal effect on homogeneous networks. However, in networks with heterogeneous bandwidth resources, since the system is agnostic about networks, it is impossible to match the communication load undertaken by each server with its communication capability. This leads to a consequence that the servers with low communication capacity slow down the communication time during the entire iteration process due to excessive load.

To efficiently handle the problem, this paper proposes an adaptive load balancing scheme for network-aware-PS-based DML over heterogeneous networks. The scheme senses the throughput of each link in networks in real time through a designed network awareness mechanism, reasonably evaluates the communication capability of each server based on this, and then selects appropriate servers to undertake the appropriate model aggregation tasks according to their communication capabilities. Finally, each server is assigned with communication load that matches its communication capability. The main contributions of this paper are as follows:

• We achieve an effective estimation of the link throughput by the low-cost and high-precision statistics method of the data transmission time with a simple and ingenious design, so as to learn the global network state information;

• We conduct an in-depth theoretical analysis of fine-grained data transmission and find a method to solve the optimal granularity of data slices.

• We design a simple and effective aggregation node selection method and a specific data slice assignment method, which can achieve efficient slice assignment.

2 Related Work

Multiple servers are typically used to alleviate heavy traffic on a single server in the PS architecture. But the specific implementation of the traditional PS architecture is network unawareness (such as MXNet[14], TensorFlow[15], and Petuum[16]), making it impossible to distribute the communication load more reasonably according to the actual communication capabilities of each server. Therefore, it is generally assumed that their communication capabilities are basically the same and are distributed according to the principle of fairness[17]. This usually results in poor performance in heterogeneous networks.

The authors in Ref. [18] have proposed an elastic PS load distribution scheme, which mainly analyzes the performance of servers by calculating the transmission time of the parameters using the linear regression method, and finally distributes communication load accordingly. Considering that the load distribution is in a complex network environment, the primary problem is the awareness of the network state. However, the authors do not provide a statistical method of parameter transmission time to implement network awareness, which makes the engineering solution to this kind of problem practically impossible. In addition, this scheme fails to deeply consider the optimal granularity of fine-grained transmission, and only uses empirical values, which cannot make the transmission reach the optimal state.

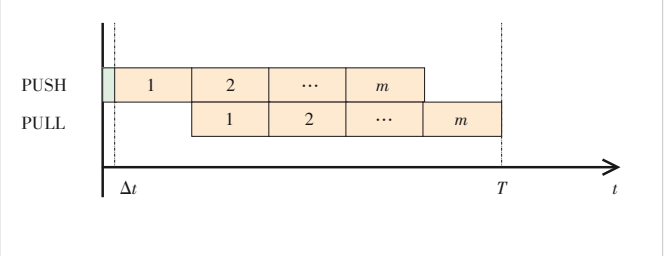
3 Proposed Approach

Based on the understanding of the related work about the PS load distribution of DML and the in-depth thinking of the problem, our approach is proposed as follows. First of all, the data are segmented according to the established slice granularity. The system in real time senses the network state through the cleverly designed network awareness mechanism, then evaluates the network communication capabilities of each node accordingly, and selects a part of the nodes as aggregation nodes. Finally, the complete distribution of fine-grained data is realized according to the PS load distribution and slice assignment algorithms.

3.1 Slice Granularity

During the model aggregation for DML, the process of workers sending data to the server to aggregate (PUSH) and the process of workers receiving the aggregated data returned from the server (PULL) are usually carried out synchronously, as shown in Fig. 1. The system performs the PULL process of data Slice 1 after all workers have completed the PUSH process of data Slice 1 (the time of data aggregation can be ignored), and the PUSH process of data Slice 2 is performed synchronously, thus overlapping PUSH and PULL. Theoretically, the smaller the data slice is, the better the overlapping of PUSH and PULL, ultimately making the aggregation quicker to complete. However, in practice, because there is a certain overhead in the data segmentation process, and there is also a certain additional network overhead in the transmission process of data slices, the granularity of slicing cannot be infinitely small.

Taking as many factors as possible into account, we analyze and solve this problem from a theoretical point of view. Considering the situation under a simple homogeneous network, in a complete data aggregation process under a single server, for a distributed system with a fixed data size M in every worker, the network bandwidth is W, and the number of nodes is N, where the slice granularity x that determines the times of the



▲ **Figure 1. Illustration of data transmission, where the green block is the additional synchronization delay, and the orange block is the transmission time of each slice**

data is sent separately by $m = M/x$ (the number of slices). In addition to the inherent transmission delay under the bandwidth limit, there are other network delays of data transmission during each data transmission. Hence, we compensate for the latency factor β. However, our study finds that the segmentation cost per slice is less than 1 ms, which can be ignored. We also find that $\beta \propto \frac{1}{W}$, thus let $\beta = \frac{1}{W}\alpha$. In addition, considering that the start time of the transmission of each node in practice is difficult to synchronize absolutely, there is an additional synchronization delay Δt in the total data transmission time. Eq. (1) shows the relationship between granularity x and the total time of data synchronization T.

$$T = \left(\frac{x}{W/N} + \frac{\alpha}{W}\right)\cdot\left(\frac{M}{x} + 1\right) + \Delta t, \tag{1}$$

where Δt denotes the delay of synchronization. We can expand the above equation to obtain:

$$T = \frac{NM}{W} + \frac{N}{W}x + \frac{M\alpha}{W}\cdot\frac{1}{x} + \frac{\alpha}{W} + \Delta t. \tag{2}$$

We simplify the above equation to the $y = x + \frac{a}{x}$ form and have:

$$\frac{W}{N}T = x + \frac{M\alpha}{N}\cdot\frac{1}{x} + M + \frac{\alpha}{N} + \frac{W}{N}\Delta t. \tag{3}$$

It can be found that when the left part of the equation takes the minimum value, the value of x is:

$$x = \sqrt{\frac{M\alpha}{N}}. \tag{4}$$

When M and N are determined, $\alpha = 1.2 \times 10^5$ can be obtained through actual testing. Obviously, at this point, the value x is only related to the data size M and the number of nodes N. It illustrates that during the distributed training of machine learning, when the training scale and the number of model parameters are determined, the value x is determined.

In a heterogeneous network, system performance is limited by the node with the smallest communication bandwidth (bottleneck node). If the bottleneck node is related to the server, W is calculated according to the bandwidth of the parameter server. If the bottleneck node is a worker, W can get the maximum value of T according to the bandwidth of the worker. However, in any case, the results are not related to W, so Eq. (4) is still of reference value for heterogeneous networks.

3.2 Network Awareness

In this scheme, the network state information that needs to be measured is only link throughput (available bandwidth). To avoid the large injection of probe traffic in the conventional network measurement technology[19-20] to occupy scarce network bandwidth resources, this scheme directly takes the model parameter data as the probe traffic. The granularity size_probe and the number probe_num of probe packets should be the minimum values that help the scheme to achieve an accurate measurement (the training iteration time remains stable in a stable heterogeneous network within a certain period of time), and they need to be determined in specific engineering implementation. Probe packets are segmented by each worker using the probability partition_rate to select the probe granularity size_probe to segment local data. In Eq. (5), where the coefficient γ is fixed at 0.6 in the experiment, the value of the probability partition_rate is necessary to ensure that the number of probe packets sent by the worker to each server is not less than probe_num, so as to realize the complete measurement of links between the worker and all servers.

$$\text{partition\_rate} = \frac{2N}{w/n}\gamma. \tag{5}$$

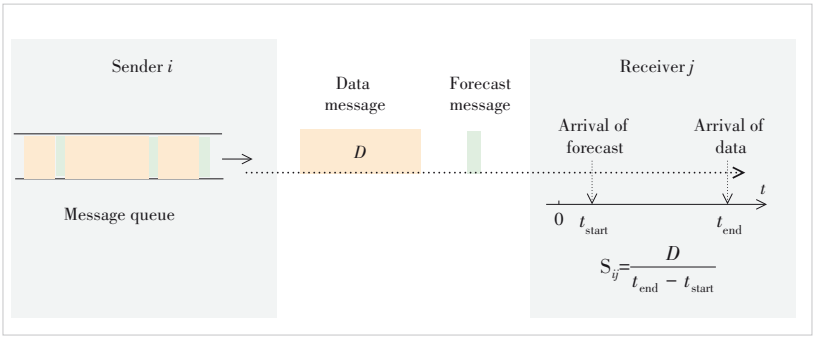
From the perspective of measurement implementation, the measurement of link throughput only needs to know the data size of the probe packet and the completion time of the probe packet transmission. Since the probe packet receiving node (receiver) has received the probe packet, the data size of the probe packet is known, but its transmission completion time is not easy to know. To calculate the transmission completion time, the start time and end time of transmission have to be figured out. When the probe packet is submitted to the upper layer, the receiver only knows the time at which the application layer received it, which is the end time of the probe packet transmission. But the receiver does not know the start time of the probe packet transmission. To obtain the start time of the probe packet transmission, the receiver can consider starting from the lower transport layer protocol and analyze the start time of the probe packet transmission in more detail, such as analyzing the Acknowledge Character (ACK) when the transmission is based on the Transmission Control Protocol (TCP). But in complex heterogeneous networks where different nodes may be deployed on different types of devices and use different network protocols, the scheme of obtaining the transmission start time of the probe packet based on the analysis of the underlying communication protocol is obviously not sufficiently pervasive.

In fact, without considering the underlying protocol analysis, it is also possible to obtain the start time of probe packet transmission. Although the application layer of the receiver does not directly know the start time of the probe packet transmission, the sending node (sender) knows. Therefore, it is only necessary to tell the receiver the start time of the probe packet transmission through the sender.

$$s_{ij} = \frac{\text{prob\_size}}{t_{\text{end}} - t_{\text{start}}}. \tag{6}$$

Specifically, before the probe packet needs to be sent, the sender i sends the forecast message to the receiver j. After receiving the forecast message, the receiver can assume that the end time of the forecast message transmission is the start time t_{start} of the following probe (packet) message transmission. Until the following probe message arrives, and the receiver obtains the end time of probe message transmission t_{end} and the data size of probe messages size_probe. Finally, according to Eq. (6), the average rate s_{ij} of the probe message transmission from node i to node j can be calculated. The process of link throughput measurement is shown in Fig. 2. We use s_{ij} as an estimate of the throughput of the link through which the probe message is transmitted, and then use the estimated throughput as a reference for the evaluation of the communication capabilities of the node associated with the link. In this process, although additional traffic (the forecast message) is also injected into the network, it is not probe traffic. It is just the signaling message which is responsible for state forecast, and the data size is very small. Thus, the overhead of transmission over the network is almost negligible.

From the overall perspective of the network awareness mechanism, the specific measurement of network awareness is distributed at each node. If the links are required for transmission, they all need to be measured. To further enhance the reliability and stability of the measurement, we not only use special probe messages but also take data messages as probe messages to measure networks. Although it leads to some overhead, considering that the final value of throughput between nodes is the average value of the throughput record, the design can further improve the measurement effect. These measurements are obtained by the receiver, and then summarized to the central scheduling node (scheduler) which is responsible for the evaluation of the communication capacity of nodes and the distribution of communication load. When each node reports the link throughput information, the scheduler will update its recorded throughput value, evaluate capacity, and make decisions under the new network state timely, so that the system has a strong adaptive ability.



▲Figure 2. Link throughput measurement

3.3 Load Distribution

Load distribution is decided by a scheduler, which mainly involves the distribution of communication load on each server and the assignment of data slices. For the distribution of communication load, system deployment needs to be considered first. As bandwidth resources are scarce in heterogeneous networks, more physical nodes are needed in networks and the utilization of link bandwidth between nodes will be lowered if servers and workers are placed separately. To avoid these problems, we attach a server to each worker to get higher network resource utilization. In such a deployment, each node not only receives and distributes aggregated data as a server but also sends and receives aggregated data as a worker. It is important to note that in such a deployment, the node acting as a worker does not need to actually send the communication load to itself acting as a server. As all nodes as servers need to bear the corresponding proportion of the communication load, and the part of the load undertaken by themselves does not need to be actually sent, it is equivalent to reducing the data transmission of a worker.

Specifically, when the number of nodes is N, the local data size of each node is M, and the communication load of server $i\,(i \in V)$ is assumed to be m_i, the communication load L_i of node i is:

$$L_i = M - m_i + (N-1)m_i. \tag{7}$$

Considering that the throughput information received by the scheduler is presented as s_{ij} from node i to node j, the actual throughput S_i of node i can be calculated by Eq. (8):

$$S_i = \sum\nolimits_{j \in Wi} s_{ij}. \tag{8}$$

Based on this, we can calculate the transmission time t_i for node i to complete communication load L_i under throughput S_i by Eq. (9):

$$t_i = \frac{L_i}{S_i} = \frac{M + (N-2)m_i}{S_i}. \tag{9}$$

In the model aggregation stage, the data transmission of each node is carried out simultaneously, so the total transmission completion time in the training iteration is the maximum of the transmission completion time of each node $\max_{i \in V} t_i$. The purpose of reasonable communication load distribution is to minimize $\max_{i \in V} t_i$. In other words, the current problem model can be determined as:

$$\min \max_{i \in V} \frac{M + (N-2)m_i}{S_i}$$
$$\text{s.t. } M = \sum_{i \in V} m_i. \tag{10}$$

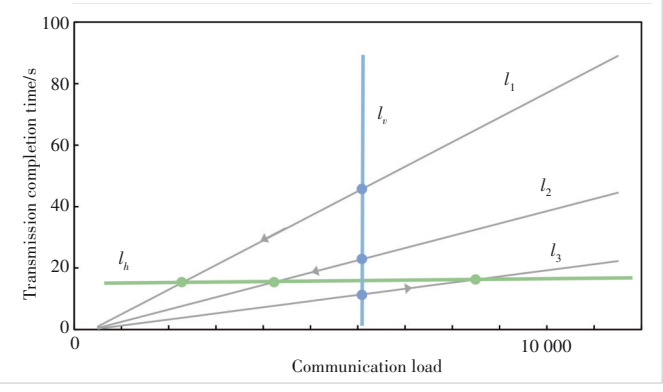
In Eq. (10), M, N and S_i are constants, and only m_i is variable. The objective function requires to minimize the maximum value of t_i. Under the strong constraint that the sum of all m_i is fixed, considering that adjusting the load of one node will inevitably affect the load of other nodes, it is intuitively difficult to determine the optimal value of m_i. However, we can write Eq. (10) as:

$$t_i = \frac{M}{S_i} + \frac{N-2}{S_i} m_i. \tag{11}$$

Eq. (11) is the linear function of t_i on m_i. For the training system with $V = \{1,2,3\}$, we draw the function curve of t_i on m_i of each node as shown in Fig. 3.

The problem of Eq. (10) can be approximately transformed to determine a point (m_i,t_i) on each line l_i in Fig. 3, and to minimize the maximum value in t_i on the premise that the sum of the abscissa of these points m_i is a constant value M. If the position of (m_i,t_i) is initialized randomly for each line and then moved gradually to minimize $\max_{i \in V} t_i$, the minimum value of $\max_{i \in V} t_i$ can be achieved if and only if all points are on the same horizontal line l_h. Otherwise, there must be a line $l_{h'}$, above and below which there are at least one point respectively. Thus, we can still get all the points closer to each other by moving the point above $l_{h'}$ down its line and moving the point below $l_{h'}$ up its line, until they are on the same horizontal line.

We distribute the communication load of each node according to the principle of equalitarianism in advance. Positions of (m_i,t_i) are initialized at the intersections of line $l_v = \frac{M}{N}$ and each line l_i. Then each point (m_i,t_i) is moved by means of iterative forced equalization of $\max_{i \in V} t_i$ and $\min_{i \in V} t_i$. Specifically, in a moving iteration, it is assumed that $i = \max$, when $t_{\max} = \max_{i \in V} t_i$, and $i = \min$, when $t_{\min} = \min_{i \in V} t_i$. When



▲**Figure 3. Geometrization of the load distribution problem**

$t_{\max} = t_{\min}$, the x-coordinates $m'_{\max}$ and $m'_{\min}$ of the moved points $(m_{\max},t_{\max})$ and $(m_{\min},t_{\min})$ have the relationship as shown in Eqs. (12) and (13).

$$\frac{M + (N-2)m'_{\max}}{S_{\max}} = \frac{M + (N-2)m'_{\min}}{S_{\min}}. \tag{12}$$

$$m'_{\max} + m'_{\min} = m_{\max} + m_{\min}. \tag{13}$$

Therefore,

$$m'_{\max} = m_{\max} + m_{\min} - m'_{\min}$$
$$x'_{\min} = \frac{(N-2)(m_{\max} + m_{\min})S_{\min} - M(S_{\max} - S_{\min})}{(N-2)(S_{\min} - S_{\max})}. \tag{14}$$

Now, $(m_{\max}, t_{\max})$ and $(m_{\min}, t_{\min})$ move to the same ordinate position and the next iteration can be started until $\max_{i \in V} y_i = \min_{i \in V} y_i$. Algorithm 1 shows the detailed steps of the process.

Algorithm 1: Load distribution

Input: The local data size of each node M, the number of nodes N, the throughput S_i of node i, and the similarity threshold similarity_threshold of t_i, where $i \in V$.
Output: The load distribution m_i of node i.

1) **Initialization:** $m_i = \frac{M}{N}$, $t_{\max} = -\infty$, $t_{\min} = \infty$
2) **for** i in V **do**
3) $t = \frac{M + (N-2)m_i}{S_i}$
4) **if** $t_{\max} < t$ **do**
5) $t_{\max} = t$
6) $\text{node}_{\max} = i$
7) **if** $t_{\min} > t$ **do**
8) $t_{\min} = t$
9) $\text{node}_{\min} = i$
10) **while** $t_{\max} - t_{\min} \geqslant$ similarity_threshold **do**
11) $m_{\text{sum}} = m_{\text{node}_{\max}} + m_{\text{node}_{\min}}$
12) $m_{\text{node}_{\min}} = \frac{(N-2)(m_{\max} + m_{\min})S_{\min} - M(S_{\max} - S_{\min})}{(N-2)(S_{\min} - S_{\max})}$
13) $m_{\text{node}_{\max}} = m_{\text{sum}} - m_{\text{node}_{\min}}$
14) $t_{\max} = -\infty$, $t_{\min} = \infty$
15) **for** i in V **do**
16) $t = \frac{M + (N-2)m_i}{S_i}$
17) **if** $t_{\max} < t$ **do**
18) $t_{\max} = t$
19) $\text{node}_{\max} = i$
20) **if** $t_{\min} > t$ **do**
21) $t_{\min} = t$
22) $\text{node}_{\min} = i$

The first line of Algorithm 1 distributes the communication

load of each node according to the principle of fairness in advance. Lines 2 – 9 determine the maximum and minimum transmission time of the nodes in the current communication load distribution, as well as the corresponding node. At Line 10, we judge whether the moving iteration needs to be stopped. In order to reduce the number of iterations, we define the difference between the maximum and minimum values of node transmission time as approximately equal if the difference is no more than similarity_threshold (the experience value is 1 s in our experiment). Lines 11 – 13 adjust the communication load of the nodes with the maximum and minimum transmission time. Lines 14 – 22 determine the maximum and minimum values of the transmission time of nodes after adjusting the communication load distribution, which is used for judgment in Line 10. Based on the above process, Algorithm 1 has a $\Theta\left(N+\frac{N}{2}\times N\right)=\Theta(N^2)$ time complexity when t_i has a uniform initial distribution on the timeline and similarity_threshold isn't too small.

In the specific process of slice assignment, data are transmitted as the slice, just like the basic granularity, thus the final work of load distribution is the assignment of data slices. Algorithm 2 shows the data slice assignment.

Algorithm 2: Slice assignment

Input: The load distribution m_i of node i, the data size paras_j of slice j, the number of slices num_slice, the granularity of probe slice size_probe, and the number of probe slices that each node sends to other nodes, where $i \in V$, $j \in (0, \text{num\_slice})$.

Output: The assignment result assign_j of slice j, where $j \in (0, \text{num\_slice})$.

1) **Initialization:** Initialize index variables index = 0.
2) **for** i **in** V
3) **for** h **in** (0, num_probe) **do**
4) **while** index < num_slice and $\text{paras}_{\text{index}} \neq$ size_probe **do**
5) index = index + 1
6) **if** index > index_end **do**
7) **break**
8) $\text{assign}_{\text{index}} = i$
9) $m_i = m_i - \text{paras}_{\text{index}}$
10) index = index + 1
11) **for** index **in** (0, num_slice) **do**
12) **if** $\text{assign}_{\text{index}}$ == NULL **do**
13) max_m = $-\infty$
14) receiver = 0
15) **for** i **in** V **do**
16) **if** max_m < $m_i - \text{paras}_{\text{index}}$ **do**
17) max_m = $m_i - \text{paras}_{\text{index}}$
18) receiver = i
19) $\text{assign}_{\text{index}}$ = receiver
20) m_{receiver} = max_m

At Lines 2 – 10 in Algorithm 2, the number of probe slices that the servers are distributed with is defined as num_probe, which is generated by segmentation probability partition_rate during data segmentation, mainly to maintain the awareness of the network state of idle nodes that are not distributed any slices. Lines 11 – 20 are used to achieve the assignment of the remaining slices. Specifically, for each slice, we traverse all current aggregation nodes and select the node with the largest remaining load as the receiving node of this slice. In this way, the receiving node with the best network state can be arranged for each slice as much as possible, and the excess load that the node needs to bear when the slice granularity is larger than the remaining load of nodes can be reduced as much as possible. Based on the above process, Algorithm 2 has a $O(\min(N \times \text{num\_probe}, \text{num\_slice}) + N \times \text{num\_slice})$ time complexity, which shows the execution time of the algorithm is mainly related to the number of nodes and data slices.

The scheme provides a standard execution process in order to make the system adaptive. In each iteration, specifically, at the beginning of the communication process, each node first reports to the scheduler the link throughput information measured in the communication process of the previous iteration, then waits for the scheduler to make the latest distribution strategy according to the link throughput information, and sends it to each node. After receiving the latest strategy information, each node updates its local strategy, transmits data according to the new strategy, and records the link throughput information measured during transmission. Based on such an interactive process, the training system can realize adaptability almost in real time.

4 Experiment

4.1 Environment and Deployment

We simulate a 12-node cluster with Intel(R) Xeon(R) E5-2678 v3 CPUs and NVIDIA 2080TI GPUs and use MXNet as a DML training platform. We have implemented our scheme by modifying the source code of MXNet and deployed the server and the worker in a 1:1 ratio, which means placing one server and one worker on each physical node in the cluster. The bandwidth limit between nodes is below the typical Wide Area Network (WAN) bandwidth of 220 M/bits with a TC-Tool[21]. The specific value of bandwidth is randomly determined and randomly adjusted periodically (300 s) to simulate the dynamic heterogeneous network environment. In addition, the hyperparameter configuration of the training system is shown in Table 1.

4.2 Experiment Design

We set up two related schemes to compare with our scheme (Aware). One scheme is Average[17], which is based on the equal distribution principle and network agnosticism,

▼Table 1. Key hyperparameters

Parameter	Value
Dataset	Fashion MNIST
Mini-batch	32
Optimizer	Adam
Learning rate	0.001

and the other is the elastic parameter distribution scheme named Elastic[18]. Since the network awareness mechanism of Elastic is unknown, we directly test Elastic based on our network awareness mechanism in experiments. For these three schemes, we test their performance on AleNet (228 MB), ResNet50 (93 MB), and MobileNet (21 MB) models respectively.

4.3 Performance Metrics

In our experiments, we use the training speed, namely the number of images per minute trained by the system, as the main performance evaluation metric. The higher the speed, the better the performance of the scheme. Eq. (16) shows the definition of speed, where num_iters is the number of iterations in t_iters time.

$$\text{speed} = \frac{\text{num\_iters}*\text{MiniBatch}*N}{t\_\text{iters}}. \tag{16}$$

In addition, single-round iteration time (SRIT) and average single-round iteration time (ASRIT) are used in the verifications of network awareness validity, verifications of segmentation granularity rationality, and cost analysis. SRIT is the time to complete a model training iteration, which is directly measured in tests. The shorter SRIT is, the better performance the scheme has.

5 Results and Analysis

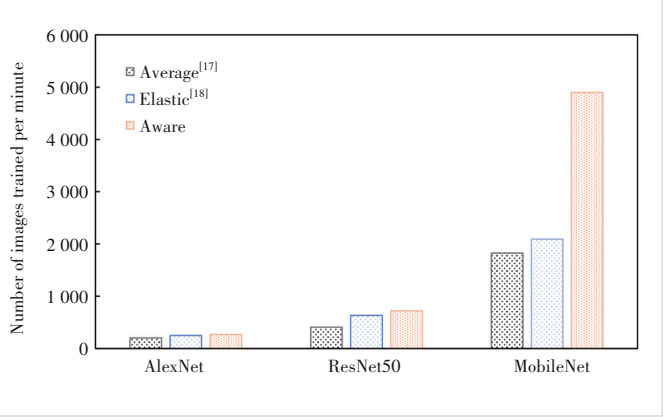
5.1 Training Speed

Fig. 4 shows the training speed of the compared schemes in different models. As we can see that network-aware Elastic and Aware schemes significantly improves performance: 1.14 times and 2.68 times for MobileNet, 1.56 times and 1.76 times for ResNet50, and 1.23 times and 1.32 times for AlexNet, compared with the Average scheme which is agnostic to network states. This shows that the PS adaptive load balancing is feasible and effective based on the network awareness. Compared with Elastic, Aware has achieved better performance improvement, 2.34 times for MobileNet, 1.13x for ResNet50, and 1.08 times for Alexnet, especially on the MobileNet model, which achieved over 2 times acceleration. This suggests that the load distribution strategy of Aware is indeed better than that of Elastic.

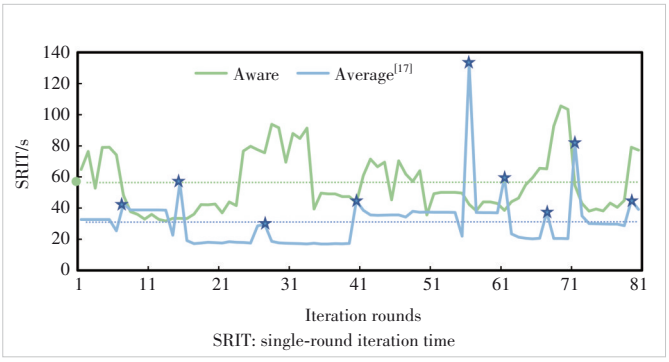
In addition, by comparing the speed gain on different models, it can be found that the gain achieved by Aware is more obvious on the smaller model (MobileNet). This is because the network load of the small model is small, the iteration time of model training is short, and the optimization effect of Aware is more significant in the same experimental network, which is finally shown as a significant increase in the training speed. On the larger model (AlexNet), Aware has almost no gain compared with Elastic. The reason is that there is no obvious room for optimization of the data aggregation process in the experiment network with limited bandwidth under the excessively large communication load.

5.2 Effectiveness Verification of Network Awareness

Fig. 5 shows the changes of SRIT of Aware and Average schemes with iteration rounds in the same dynamic network. The system parameters num_probe and size_probe are set to the best values of 2 and 10 000, respectively, which are determined by actual tests in the experiment. Due to space limitation of the paper, the details are omitted. In the figure, the curve of Average which is agnostic about the network is above the curve of Aware, which indicates that the optimization effect of Aware scheme is significant and lasting. Additionally, the curve of Aware exhibits periodic shock wave characteristics, which can be attributed to its poor performance in response to abrupt changes in network states at the crest and the end of the strategy. However, with the release of a new round



▲Figure 4. Training speed of different schemes on different models



▲Figure 5. SRIT comparison of Aware and Average schemes in dynamic networks

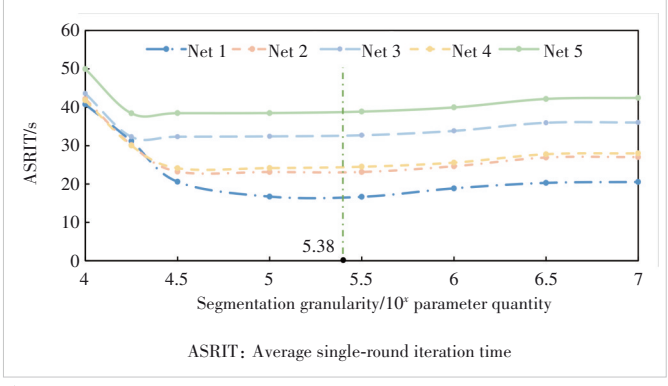
of strategy based on the latest network state, the performance of Aware improves rapidly. That also verifies the effectiveness and reliability of the network awareness mechanism of our scheme.

5.3 Reasonableness Verification of Segmentation Granularity

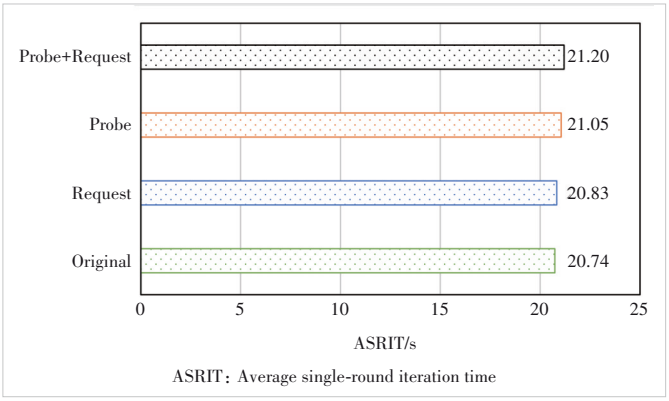
In order to verify the rationality of the theoretical analysis conclusion of slice granularity, we take the Resnet50 model as an example to test the change of ASRIT with the slice granularity under multiple network states. As shown in Fig. 6, in different network states, ASRIT remains almost unchanged within the logarithmic range of 5 – 5.5 (quantity of 10^5 – 3.16×10^5 parameters) of the slice granularity, while our theoretical value of 5.38 is exactly within this range. This indicates that our theoretical value of slice granularity can indeed achieve almost the lowest ASRIT in different network states.

5.4 Overhead Analysis

The overhead of the Aware scheme is likely to be concentrated in frequent forecast messages and synchronization of strategy requests with each round. As for the former, there should be no significant overhead because the preview message only contains extremely short header fields with a fixed length. As for the latter, because the experiments are based on the synchronous training mode and the synchronization of each round has already existed, there should be no obvious overhead. In order to verify this analysis, in a stable (static and isomorphic) network environment, we have tested ASRIT of the Average scheme under four conditions: requiring probe and strategy request synchronization (Probe + Request), only requiring probe (Probe), only requiring strategy request synchronization (Request) and neither requiring probe nor strategy request synchronization (Original). The ASRIT over dozens of iterations is shown in Fig. 7. Adding probe or strategy request synchronization does incur some overhead, but even with Probe + Request having the largest overhead, only 0.44 s (2.12%) overhead is added to Original, which is negligible compared with the huge gain shown in Fig. 5.



▲Figure 6. ASRIT of Aware scheme in different network states



▲Figure 7. ASRIT of Average scheme in different conditions

6 Conclusions

In this paper, we study the problem of PS load distribution in DML in heterogeneous networks. The state-of-the-art schemes cannot match the communication load with the communication capacity of PSs to achieve load balancing due to the lack of network awareness. The existing schemes with network awareness have not given specific network measurement methods, which makes them difficult to be realized in practice. This paper proposes a well-designed network awareness mechanism, which can realize low cost and high precision network measurement. In addition, the slice granularity determination and slice assignment of fine-grained transmission is studied. We have implemented the scheme in MXNet, and completed the function verification and performance measurement based on the experiment cluster. The results show that the proposed scheme can significantly accelerate DML.

References

[1] YU J, TAN M, ZHANG H Y, et al. Hierarchical deep click feature prediction for fine-grained image recognition [J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 44(2): 563 – 578. DOI: 10.1109/TPAMI.2019.2932058
[2] KRIMAN S, BELIAEV S, GINSBURG B, et al. Quartznet: deep automatic speech recognition with 1D time-channel separable convolutions [C]//2020 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2020: 6124 – 6128. DOI: 10.1109/ICASSP40776.2020.9053889
[3] AHMAD F, ABBASI A, LI J J, et al. A deep learning architecture for psychometric natural language processing [J]. ACM transactions on information systems, 2020, 38(1): 1 – 29. DOI: 10.1145/3365211
[4] HONG R, CHANDRA A. DLion: Decentralized distributed deep learning in micro-clouds [C]//Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing. ACM, 2021: 227 – 238. DOI: 10.1145/3431379.3460643
[5] TIAN L, YANG M Z, WANG S G. An overview of compute first networking [J]. International journal of web and grid services, 2021, 17(2): 81 – 97. DOI: 10.1504/ijwgs.2021.114566
[6] KRÓL M, MASTORAKIS S, ORAN D, et al. Compute first networking: distributed computing meets ICN [C]//The 6th ACM Conference on Information-Centric Networking. ACM, 2019: 67 – 77. DOI: 10.1145/3357150.3357395

[7] AWAN A A, HAMIDOUCHE K, HASHMI J M, et al. Scaffe: co-designing MPI runtimes and caffe for scalable deep learning on modern GPU clusters [J]. ACM sigplan notices, 2017, 52(8): 193 – 205
[8] WANG S, LI D, GENG J K, et al. Impact of network topology on the performance of DML: Theoretical analysis and practical factors [C]//IEEE Conference on Computer Communications. IEEE, 2019: 1729 – 1737. DOI: 10.1109/INFOCOM.2019.8737595
[9] LI M, ZHOU L, YANG Z, et al. Parameter server for distributed machine learning [J]. Big learning NIPS workshop, 2013, 6: 2 – 12
[10] LI M, ANDERSEN D G, PARK J W, et al. Scaling distributed machine learning with the parameter server [C]//The 11th USENIX conference on Operating Systems Design and Implementation. ACM, 2014: 583 – 598. DOI: 10.5555/2685048.2685095
[11] LI M, ANDERSEN D G, SMOLA A, et al. Communication efficient distributed machine learning with the parameter server [C]//The 27th International Conference on Neural Information Processing Systems. ACM, 2014: 19 – 27
[12] ZHANG S, CHOROMANSKA A, LECUN Y. Deep learning with elastic averaging SGD [C]//The 28th International Conference on Neural Information Processing Systems. ACM, 2015: 685 – 693
[13] DEAN J, CORRADO G S, MONGA R, et al. Large scale distributed deep networks [J]. Advances in neural information processing systems, 2012, 1: 1223 – 1231
[14] CHEN T Q, LI M, LI Y T, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems [EB/OL]. [2022-10-10]. https://arxiv.org/abs/1512.01274
[15] ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems [EB/OL]. [2022-10-10]. https://arxiv.org/abs/1603.04467
[16] XING E P, HO Q, DAI W, et al. Petuum: a new platform for distributed machine learning on big data [J]. IEEE transactions on big data, 2015, 1(2): 49 – 67. DOI: 10.1109/tbdata.2015.2472014
[17] MXNET. Distributed training in MXNet. [EB/OL]. [2022-10-10]. https://mxnet.apache.org/versions/1.7.0/api/faq/distributed_training
[18] CHEN Y R, PENG Y H, BAO Y X, et al. Elastic parameter server load distribution in deep learning clusters [C]//Proceedings of the 11th ACM Symposium on Cloud Computing. ACM, 2020: 507 – 521. DOI: 10.1145/3419111.3421307
[19] MOHAN V, REDDY Y J, KALPANA K. Active and passive network measurements: a survey [J]. International journal of computer science and information technologies, 2011, 2(4): 1372 – 1385
[20] GOEL U, WITTIE M P, CLAFFY K C, et al. Survey of end-to-end mobile network measurement testbeds, tools, and services [J]. IEEE communications surveys & tutorials, 2016, 18(1): 105 – 123. DOI: 10.1109/COMST.2015.2485979
[21] TC(8). Linux tc. [EB/OL]. [2022-10-10]. https://linux.die.net/man/8/tc

Biographies

CAI Weibo is pursuing his master's degree in communication and information system at University of Electronic Science and Technology of China. His research focuses on distributed machine learning.

YANG Shulin is pursuing his master's degree in communication and information system at University of Electronic Science and Technology of China. His research focuses on distributed machine learning.

SUN Gang (gangsun@uestc.edu.cn) is a professor of computer science at University of Electronic Science and Technology of China. His research interests include machine learning, cloud computing, high performance computing, parallel and distributed systems, ubiquitous/pervasive computing and intelligence and cyber security.

ZHANG Qiming is a senior system architect of ZTE Corporation. He received his bachelor's degree from Zhejiang University, China in 1992. His research interests include MEC and heterogeneous computing.

YU Hongfang is a professor of University of Electronic Science and Technology of China. Her research interests include network virtualization, cloud computing and next generation network.

Scene Visual Perception and AR Navigation Applications

LU Ping[1,2], SHENG Bin[2], SHI Wenzhe[1,2]
(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Shanghai Jiao Tong University, Shanghai 200240, China)

DOI: 10.12142/ZTECOM.202301010
https://kns.cnki.net/kcms/detail/34.1294.TN.20230215.1801.002.html, published online February 16, 2023
Manuscript received: 2022-11-01

Abstract: With the rapid popularization of mobile devices and the wide application of various sensors, scene perception methods applied to mobile devices occupy an important position in location-based services such as navigation and augmented reality (AR). The development of deep learning technologies has greatly improved the visual perception ability of machines to scenes. The basic framework of scene visual perception, related technologies and the specific process applied to AR navigation are introduced, and future technology development is proposed. An application (APP) is designed to improve the application effect of AR navigation. The APP includes three modules: navigation map generation, cloud navigation algorithm, and client design. The navigation map generation tool works offline. The cloud saves the navigation map and provides navigation algorithms for the terminal. The terminal realizes local real-time positioning and AR path rendering.

Keywords: 3D reconstruction; image matching; visual localization; AR navigation; deep learning

Citation (IEEE Format): P. Lu, B. Sheng, and W. Z. Shi, "Scene visual perception and AR navigation applications," *ZTE Communications*, vol. 21, no. 1, pp. 81 – 88, Mar. 2023. doi: 10.12142/ZTECOM.202301010.

1 Introduction

Navigation services applied to mobile devices are an indispensable part of modern society. At present, the outdoor positioning and navigating service technology has become mature, and the Global Positioning System (GPS) can provide relatively accurate position information and related supporting navigation services for outdoor pedestrians. For example, the navigation products of Baidu, Amap, Tencent and other companies can meet the location information and navigation service needs of outdoor pedestrians in terms of location services. However, once pedestrians go indoors, e.g., in shopping malls, airports, underground parking lots and other sheltered places, the positioning signal is greatly attenuated by factors like walls, and the GPS-based outdoor navigation technology becomes insufficient. The existing indoor localization methods have many constraints in localization accuracy, deployment overhead, and resource consumption, which limits their promotion in real-world navigation applications.

In recent years, researchers have designed a variety of indoor and outdoor positioning solutions for various types of information such as visible light communication (VLC), built-in sensors, QR codes, and WIFI. However, these solutions have many shortcomings in terms of localization accuracy, deployment difficulty, and equipment overhead. For example, the VLC-based methods require indoor LED lights to be upgraded on a large scale, which greatly increases deployment costs. Meanwhile, the WIFI-based methods cannot provide accurate direction information, which is difficult to meet the needs of precise localization.

However, in a visual scenario perception method, target recognition and position calculation are performed by means of image processing, so that relatively high positioning precision can be provided, and deployment of an additional device is not required, which is widely researched and applied in recent years.

The main application of scene perception is visual localization, which is a method of determining the position of 6-degree of freedom (6-DoF) from the image. The initialization conditions of visual localization usually require a sparse model of the scene and the estimated pose of the query image. Augmented reality (AR) navigation is an important application scenario of visual localization technologies, which can interact with the real world in a virtual environment through localization. The application of AR navigation technologies has great prospects in the future. Shopping malls have the most demand for localization and navigation technologies, and users are very interested in store discount information, personalized advertisements, store ratings, store locations, and indoor road

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-20210707004.

guidance. The application of scene visual perception and AR navigation can solve most of the above problems well, and has vast potential in future development in the expansion of added value.

This paper introduces the design and implementation of AR navigation applications (APPs) and the cloud algorithm in detail, and starts from three aspects: navigation map generation, the cloud navigation algorithm, and the client design. Combined with specific cases, this paper introduces in detail the process of panoramic data acquisition and processing, point cloud map[1] and computer aided design (CAD) map alignment in the navigation map generation tool, and introduces the path planning algorithm and path correction algorithm in the cloud navigation algorithm. In terms of localization and AR path rendering, the client design method is introduced in detail, and finally, the running example of an AR navigation APP is given.

2 Basic Framework of Scene Visual Perception

Similar to humans, machines perceive and understand the environment mostly through visual information. In recent years, the development of 3D visual perception methods has provided great help for building models of the real physical world. For various application scenarios, there are currently some vision algorithms with commercial application capabilities, including face recognition, living body detection, 3D reconstruction, simultaneous localization and mapping (SLAM), gesture recognition, behavior analysis, augmented reality, virtual reality, etc.

Scene visual perception applied to navigation mainly includes 3D reconstruction and SLAM. The above steps can be regarded as the process of building a visual map. Visual map-based localization usually includes steps such as visual map construction and update, image retrieval, and fine localization, among which the visual map is the core of the method. According to the condition that the image frame has accurate prior pose information or not, the process of constructing a visual map can be divided into prior pose-based construction methods and non-prior pose methods. In the prior pose-based construction methods, the prior pose of the image frame can be derived from the high-precision LiDAR data synchronized and calibrated with the camera, which is common in high-precision acquisition vehicles in the field of autonomous driving. In small-scale scenes, especially indoors, the prior pose can also be obtained from visual motion capture systems such as Vicon and OptiTrack. The non-prior pose methods adopt offline extraction of feature points and offline optimization of pose and scene structures, which is similar to structure-from-motion (SfM). The constructed geometric visual map generally includes image frames, feature points and descriptors, 3D points, the correspondence between image frames, and the correspondence between 2D points and 3D points. During the process, due to changes in the real scene, the constructed visual map also needs to be updated synchronously to detect new and expired changes in time, and then update the corresponding changes to the visual map. When the prior visual map is obtained, the image retrieval and fine localization steps can usually be performed on the newly acquired image frame to complete localization. In the visual map-based localization framework, sensor information such as inertial measurement unit (IMU), GPS, and wheel odometer can also be fused.

3 Introduction to Key Technologies of Scene Visual Perception

3.1 3D Reconstruction

Accurate and robust 3D reconstruction methods are crucial to visual localization. The purpose of 3D reconstruction is to obtain the geometry and structure of an object or a scene from a set of images. SfM is a way to achieve 3D reconstruction, which is mainly used in the stage of building sparse point cloud in 3D reconstruction. A complete 3D reconstruction process usually also includes a multi-view stereo (MVS) step to achieve dense reconstruction. SfM is mainly used for mapping and restoring the structure of the scene. According to the difference in the image data processing flow, SfM can usually include four categories: incremental SfM, global SfM, distributed SfM, and hybrid SfM. Among them, distributed SfM and hybrid SfM are usually used to solve large-scale reconstruction and are based on incremental SfM and global SfM. Incremental SfM mainly includes two steps. The first step is to find the initial correspondence, and the second step is to achieve incremental reconstruction. The former aims to extract robust and well-distributed features to match image pairs, and the latter is used to estimate the image pose and 3D structure through image registration, triangulation, bundle adjustment (BA), and outlier removal. The initial corresponding outliers usually need to be removed by geometric verification methods. Generally, when the number of recovered image frames accounts for a certain proportion, global BA is required. Due to the incremental BA processing, incremental SfM usually has higher accuracy and better robustness. As the number of images increases, the scale of BA processing becomes larger, leading to disadvantages such as low efficiency and large memory usage. Additionally, incremental SfM suffers from cumulative drift as images are incrementally added. Typical SfM frameworks include Bundler and COLMAP.

CAO et al.[2] proposed a fast and robust feature tracking method for 3D reconstruction using SfM. First, to save computational costs, a feature clustering method was used to cluster a large set of images into small ones to avoid some wrong feature matching. Second, the joint search set method was used to achieve fast feature matching, which could further save the computational time of feature tracking. Third, a geometric constraint method was proposed to remove outliers in trajectories produced by feature tracking methods. The method could cope with the effects of image distortion, scale changes, and illumi-

nation changes. LINDENBERGER et al.[3] directly aligned low-level image information from multiple views, optimized feature point locations using depth feature metrics after feature matching, and performed BA through similar depth feature metrics during incremental reconstruction. In this process, the convolutional network was used to extract the dense feature map from the image, then the position of the feature points in the image was adjusted according to the sparse feature matching to obtain the two-dimensional observation of the same 3D point in different images, and the SfM reconstruction was completed according to the adjustment. The BA optimization residual in the reconstruction process changes from reprojection error to feature metric error. This improvement is robust to large detection noise and appearance changes, as it optimizes feature metric errors based on dense features predicted by neural networks.

The cumulative drift problem can be solved by global SfM. For the fundamental and essential matrix between images obtained in the image matching process, the relative rotation and relative translation can be obtained through decomposition. Using the relative rotation as a constraint, the global rotation can be recovered, and then the global translation can be recovered using the global rotation and relative translation constraints. Since the construction of the global BA does not require multiple optimizations, the global SfM is more efficient. However, since the relative translation constraints only constrain the translation direction and the scale is unknown, the translation averaging is difficult to solve. In addition, the translational average solution process is sensitive to outliers, so the global SfM is limited in practical applications.

3.2 Image Matching

How to extract robust, accurate, and sufficient image correspondences is a key issue in 3D reconstruction. With the development of deep learning, learning-based image matching methods have achieved excellent performance. A typical image matching process usually includes three steps: feature extraction, feature description, and feature matching.

Detection methods based on deep convolutional networks search for interest points by constructing response graphs, including supervised methods[4-5], self-supervised methods[6-7], and unsupervised methods[8-9]. Supervised methods use anchors to guide the training process of the model, but the performance of the model is likely to be limited by the anchor construction method. Self-supervised and unsupervised methods do not require human-annotated data, while they focus on geometric constraints between image pairs. Feature descriptors use local information around interest points to establish the correct correspondence of image features. Due to the information extraction and representation capabilities, deep learning techniques have also achieved good performance in feature descriptions. The deep learning-based feature description problem is usually a supervised learning problem, that is, learning a representation so that the matched features in the measurement space are as close as possible, and the unmatched features are as far as possible[10]. Learning-based descriptors largely avoid the requirement of human experience and prior knowledge. Existing learning-based feature description methods include two categories, namely metric learning[11-12] and descriptor learning[13-14], and the difference lies in the output content of the descriptor. Metric learning methods learn metric discriminants for similarity measurement, while descriptor learning generates descriptor representations from raw images or image patches.

Among these methods, SuperGlue[14] proposed a network capable of feature matching and filtering outliers simultaneously, whose feature matching was achieved by solving a differentiable optimization transfer problem. The loss function was constructed by a graph neural network, and a flexible content aggregation mechanism was proposed based on the attention mechanism, which enabled SuperGlue to simultaneously perceive potential 3D scenes and perform feature matching. LoFTR[15] used a transformer module with self-attention and cross-attention layers to process dense local features extracted from convolutional networks. Dense matches were first extracted at a low feature resolution (1/8 of the image dimension), from which high-confidence matches were selected and refined to high-resolution sub-pixel levels using correlation-based methods. In this way, the large receptive field of the model enabled the transformed features to reflect context and location information, and the prior matching was achieved through multiple self-attention and cross-attention layers. Many methods integrate feature detection, feature description, and feature matching into matching pipelines in an end-to-end manner, which is beneficial for improving matching performance.

3.3 Visual Localization

Visual localization is a problem of estimating the pose of a 6-DoF camera, from which a given image is obtained relative to a reference scene representation. Classical approaches to visual localization are structure-based, which means that they rely on 3D reconstructions of the environment (e. g. point clouds) and use local feature matching to establish correspondences between query images and 3D maps. Image retrieval can be used to reduce the search space by considering only the most similar reference images instead of all possibilities. Another approach is to directly interpolate the pose from the reference image or estimate the relative pose between the query and the retrieved reference image, which does not rely on the 3D reconstruction results. Scene point regression methods can directly obtain the correspondence between 2D pixel positions and 3D points using a deep neural network (DNN), and compute camera poses similar to structure-based methods. Modern scene point regression methods benefit from 3D reconstruction during training but do not rely on it. Absolute pose regression methods use a DNN to estimate poses end-to-

end. These methods differ in generalization ability and localization accuracy. Furthermore, some methods rely on 3D reconstruction, while others only require pose-labeled reference images. The advantage of using 3D reconstructions is that the generated poses can be very accurate, while the disadvantage is that these 3D reconstructions are sometimes difficult to obtain and even more difficult to maintain. For example, if the environment changes, they need to be updated.

The typical work of the structure-based approach can refer to a general visual localization pipeline proposed in Ref. [17]. Through a hierarchical localization approach, the pipeline can simultaneously predict local features and global descriptors for accurate 6-DoF localization, which utilizes a coarse-to-fine localization paradigm, first performing global retrieval to obtain location hypotheses and then matching local features in these candidate locations. This hierarchical approach saves runtime for real-time operations and proposes a hierarchical feature network (HF-Net) that jointly estimates local and global features, thereby maximizing shared computation, and compresses the model through multi-task distillation.

4 AR Navigation Based on Scene Visual Perception

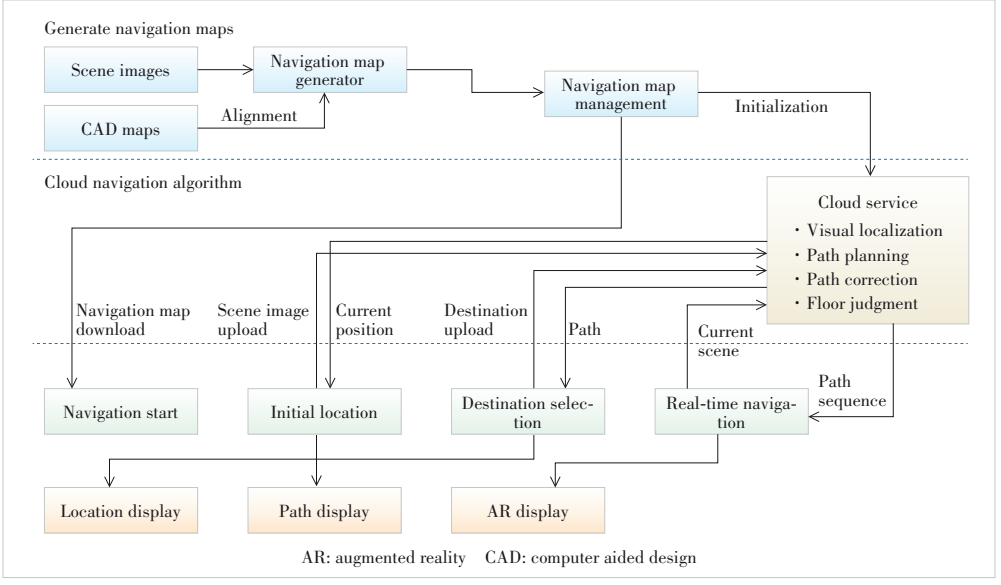
AR navigation usually works in the following process: 1) The real-world view is got from the user's point of view; 2) the location information is obtained and used to track the user; 3) virtual-world information is generated based on the real-world view and location information; 4) the generated virtual world information is registered into the real-world view and displayed to the user, creating augmented reality. The main challenge of AR navigation is how to integrate the virtual and real worlds, and design and present the navigation interface. Registration is the process of correctly aligning virtual information with the real world, which gives the user the illusion of keeping the virtual and the real coexisting. For AR in navigation, accurate registration is critical, and AR navigation systems can cause confusion when orientation changes rapidly due to registration errors. So even small offsets of registering dummy information can be harmful. In an AR navigation system, the display should not interfere with the user's movement. The augmented reality display technology is also known as video see-through. Video see-through display refers to placing a digital screen between the real world and the user, where the user can see the real world and augmented information, use a camera to capture the real-world view, and then combine it with the augmented information and display it on the screen superior. Typical examples of displays include head-mounted displays with cameras and smartphone displays.

On the basis of scene visual perception, this paper designs an AR navigation APP developed based on Unity and ARCore. Its overall framework is shown in Fig. 1. The system consists of three parts, namely, the navigation map generation tool, the cloud navigation algorithm, and the terminal navigation APP design.

The navigation map generation tool works offline, including scene panoramic video capture, dense point cloud generation, point cloud and plane CAD map alignment, navigation map management and other functions. The map generated by the navigation map generation tool is stored in the cloud. In addition, the cloud is also responsible for providing navigation algorithms to the terminal, including visual localization methods, path planning algorithms, path correction algorithms, floor judgment algorithms and cross-layer guidance algorithms. When users request a navigation activity with the terminal APP, they first select the current location map, and the cloud issues the corresponding navigation map according to the user's selection. After selecting the starting point and ending point, the user requests the navigation service from the cloud, and realizes local real-time localization, global path and current position display, and AR path rendering in the local APP.

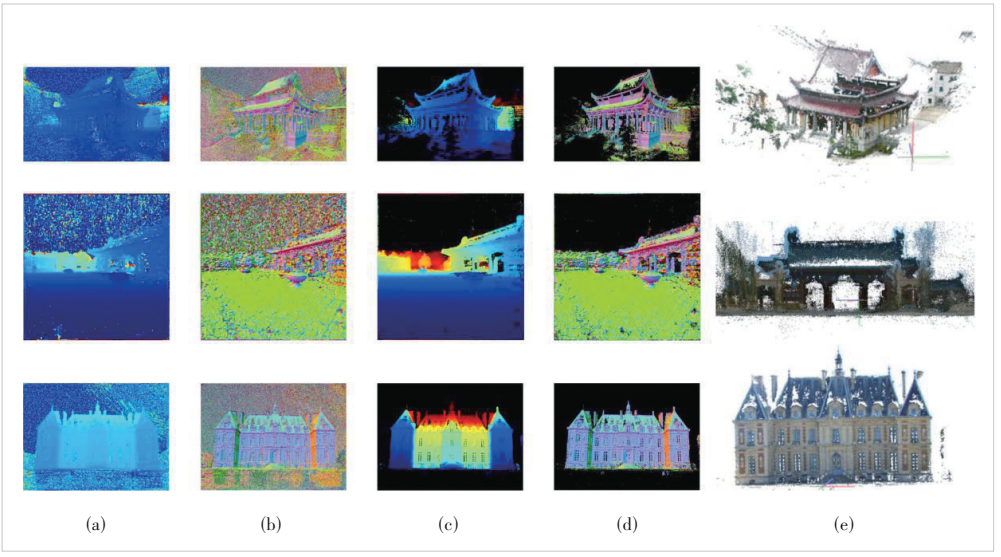
4.1 Panoramic Data Collection and Processing

This paper uses a panoramic camera to capture video to collect mapping data. Instead of rotating the camera around its optical center, this panoramic camera can be used to capture



▲Figure 1. Overall framework of an AR navigation application (APP)

multiple images of a scene from different viewpoints, from which stereoscopic information about the scene can be calculated. The stereo information is then be used to create a 3D model of the scene, and arbitrary views can be computed. This approach is beneficial for 3D reconstruction of large-scale scenes. The dense reconstruction results of the proposed approach on the building dataset are shown in Fig. 2.



▲**Figure 2. Result of dense reconstruction: (a) photometric depth map, (b) photometric normal map, (c) geometric depth map, (d) geometric normal map, and (e) dense reconstruction effect**

Taking a large shopping mall as an example, for the processing and 3D reconstruction of the data collected from the panoramic video, this paper goes through the following steps:

1) Shoot a panoramic video of the scene, and the shooting area should be covered as much as possible;

2) Frame the obtained panoramic video to obtain a panoramic image and segment the panoramic image according to the field of view (FOV);

3) Realize sparse point cloud reconstruction for each floor and finally output all camera parameters and sparse 3D point cloud;

4) Complete the single-layer dense point cloud reconstruction;

5) Integrate multiple layers of dense point clouds to obtain a complete 3D structure of the scene.

4.2 Alignment of Point Cloud Map and CAD

The point cloud obtained in Section 3.1 is based on the camera coordinate system, which must be aligned with the world coordinate system if it is to be used for navigation tasks. This paper takes the CAD map as the world coordinate system, because CAD can provide accurate position information and scale information. The problem is transformed into the alignment of the point cloud map and the plane CAD. The specific process of its realization is as follows:

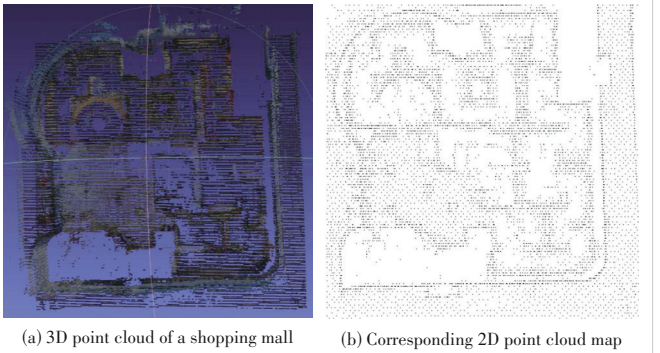
1) The point cloud is dimensionally reduced and projected to the XoY plane to form a plane point cloud map, as shown in Fig. 3.

2) Marker points (such as walls and other points that are easy to be distinguished) and the corresponding points are found on the plane point cloud map and the CAD map, respectively.

3) Alignment is completed through the scale information provided by the CAD map, output rotation and the displacement matrix.

Once the point cloud $\boldsymbol{X}$ is sampled, it can be mapped to a 2D plane by simply removing the z coordinates. The problem is transformed into finding the mapping between $(\boldsymbol{X}_x, \boldsymbol{X}_y)$ and pixels (u, v), where $(\boldsymbol{X}_x, \boldsymbol{X}_y)$ is the set of 2D coordinates (x, y) extracted from the point cloud X. It is worth noting that (x, y) are usually float values, while pixel coordinates (u, v) are usually positive integer values. Therefore, (x, y) needs to go through a certain scale, rotation and rounding transformation.

Once the plane point cloud map is obtained, it can be aligned with the CAD map through the affine transformation. To determine the affine matrix, at least three pairs of corresponding points are usually required. Considering the need to reduce errors, this paper selects multiple pairs of corresponding points in the point cloud map and CAD map respectively, and uses the least square method to achieve alignment. It is worth noting that the selection of corresponding points should try to select parts that are easy to identify, such as walls and other fixed objects with clear structural characteristics. Fig. 3 shows the process of aligning a point cloud map with a CAD map. After the alignment, the position coordinates of the point cloud in the world coordinate system can be obtained, which



▲**Figure 3. An example of a 2D point cloud map generation**

is beneficial to the subsequent localization and navigation tasks. The obtained results can be saved separately according to the scene, and the saved content includes the scene pose, corresponding geographic information, camera model, and other information to form a navigation digital map.

4.3 Cloud Navigation Algorithm

When a user requests a navigation activity with the terminal APP, he first selects the map corresponding to the current location, and the cloud issues the corresponding navigation map according to the user's selection. After the user selects the destination, the user requests the navigation service from the cloud, and at the same time uploads the current scene graph to the cloud. At this time, the cloud needs to invoke the visual localization algorithm to determine the current initial position of the user as a starting point. After obtaining the coordinates of the starting point and the ending point, the cloud calls the path planning algorithm to obtain the navigation path point sequence and sends it to the terminal APP for AR rendering. The user is actually positioned through ARCore during the process of traveling. However, this method will generate accumulated errors after traveling for a certain distance, and since the user may deviate from the recommended path, the path correction algorithm needs to be implemented through the cloud, and the user is directed to the correct path.

According to common practice in the industry, the path planning algorithm designed in this paper does not need to provide a path from any point to any point. The path planning involved in this paper only needs to provide a path from any point (user location or user-selected location) to a specific point (specified end-point set). Therefore, the path planning problem in this paper can be regarded as solving the shortest path problem between the vertices of a directed graph. The basic flow of the path planning algorithm proposed in this paper is as follows:

1) The passable area is determined through the point cloud map, and the waypoint is selected in the passable area.

2) The route point and the destination point (the selected end-point) form a graph structure.

3) The shortest path is found among all vertices in the graph through a search algorithm.

The process of building route points and destination points into a graph structure forms a road network. In this process, it is necessary to clarify the world coordinates of the waypoint and the destination point, and mark the connection relationship between points to form a graph structure of the road network, which is stored in the form of an adjacency list. Since the purpose of this paper is to find the shortest path among all vertices in the graph, it constitutes an all pairs shortest paths (APSP) problem. The general solution to the APSP problem is the Floyd-Warshall algorithm. After the shortest path among all points is obtained, the result is saved in the cloud according to the scene, so that in practical applications, there is no need to calculate the planned path online, and only the retrieval function will be implemented, which is time-consuming.

During the user's journey, the local positioning provided by ARCore will gradually produce errors with the advancing distance. At the same time, the user may deviate from the recommended navigation path due to internal or external reasons. Therefore, the cloud needs to provide a path correction algorithm to guide the user back to the navigation path (the correct path). The specific workflow of the path correction algorithm is as follows:

1) The user uploads the current scene image while traveling.

2) The cloud determines whether it deviates from the navigation path recommended by the algorithm according to the positioning algorithm.

3) If the user's deviation is small, the user will be guided to the recommended navigation path through the navigation arrows of the terminal APP. If the user's deviation is too large, the path planning will be re-planned based on the user's current position.

The path correction process is actually a verification process of the real-time local positioning information fed back by the terminal. When the error exceeds the distance threshold τ, the path correction function can be activated. In practical applications, the selection of the distance threshold τ is usually between 50 cm and 200 cm. If the threshold is too small, it will increase the influence of visual positioning errors. If the threshold is too large, it will not only lose the accuracy of navigation, but also bring inconvenience to users.

4.4 AR Systems

AR systems contain three basic features: the combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects. In this way, AR changes people's continuous perception of the real environment and obtains an immersive experience by integrating the composition of the virtual world into people's perception of the real environment. Specific to AR navigation APPs, users can obtain real-world information from smartphones (through the phone camera), and by applying the AR technology, virtual navigation paths can be added to the smartphone's interface, enhancing the user's perception of the real environment for a better navigation experience. From the user's point of view, a complete AR navigation includes the following process: 1) The user selects the current scene and obtains the navigation map delivered by the cloud; 2) the user selects the destination according to the navigation map and requests the cloud navigation service; 3) the user follows the terminal interface rendering AR path to the end. Due to network bandwidth limitations, users cannot obtain real-time localization by sending the current scene image to the cloud in real time. Therefore, the ARCore-based method is used to provide real-time localization. However, this method will generate accumulated er-

rors after traveling for a certain distance. And since users may deviate from the recommended path, path correction needs to be implemented through a correction algorithm to guide users to the correct path. Fig. 4 shows the flow of the AR navigation APP and AR rendering.

ARCore is an AR application platform provided by Google, which can be easily combined with 3D engines such as Unreal and Unity. ARCore provides three main applications for motion tracking, environment understanding, and lighting estimation. Among them, motion tracking enables the phone to know and track its position relative to the world, environment understanding enables the phone to perceive the environment, such as the size and location of detectable surfaces, and light estimation allows the phone to obtain the current lighting conditions of the environment. Localization can be achieved using ARCore's motion-tracking capabilities.

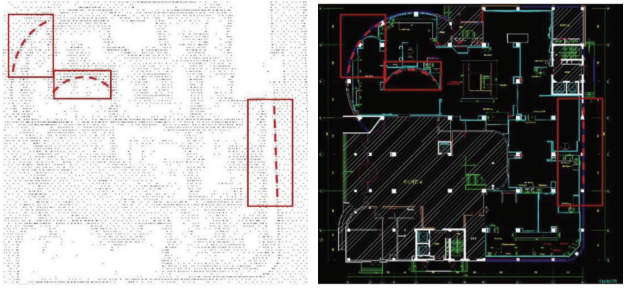
The motion-tracking function of ARCore is actually realized by visual inertial odometry (VIO). VIO includes two parts: a visual tracking system and an inertial navigation system. The camera obtains a frame of pixel matching to track the user's pose. The inertial navigation system realizes position and attitude tracking through an IMU, which usually consists of an accelerometer and a gyroscope. The outputs of the two systems are combined through a Kalman filter to determine the final pose of the user. The local positioning function provided by ARCore can track the user's position in real time, but the error in the inertial navigation system of ARCore will accumulate over time. As the user's advancing distance increases and time passes, tracking of the user's position will be offset. In practice, we find that after a user travels about 50 m, the localization provided by ARCore will begin to deviate. At this time, it is necessary to relocate through the visual localization algorithm and correct the path.

On the basis of the previous work, the AR navigation APP can obtain the current position of the user and the path point sequence of the path planning from the cloud. Then the next question is how to realize AR rendering of the path point sequence on the mobile phone interface. From the perspective of user experience, the AR markers cannot block the user's line of sight and must provide an obvious guiding role. Therefore, in the actual rendering process, this paper chooses to render the AR markers close to the ground. The environment understanding section in ARCore provides plane detection capabilities. In fact, ARCore stipulates that all virtual objects need to rely on planes for rendering. After ARCore implements plane detection, the AR markers can be placed on the ground. The placement of AR markers can be achieved by radiographic inspection. The principle of ray detection is to judge whether there is a collision with an object through the ray emitted from the camera position to any position in the 3D world. In this way, the collision object and its position can be detected. By performing collision detection on the planes in the scene, the planes can be judged and AR signs can be placed. Here, this paper adopts two kinds of AR markers, one is the navigation guidance arrow, which is responsible for indicating the forward direction, and the other is the end prompt sign, which reminds the user to reach the end-point. Fig. 4 shows the actual workflow of the AR navigation APP and the rendering effect of the AR markers. In the figure, from left to right, the user selects the destination (elevator entrance), the navigation guide arrow is rendered, the user follows the navigation guide arrow, and the navigation ends at the end prompt sign.

5 Conclusions and Outlook

This paper analyzes and introduces related technologies in the field of scene visual perception, based on which we implement AR navigation. In practical application, there are still some problems to be solved[18-19]. For example, this paper adopts a structure-based localization framework, with an advantage that it can effectively handle large-scale scenes and has high localization accuracy. However, if the environment changes, the 3D structure needs to be re-adjusted to achieve re-registration of point clouds. The alignment method of point cloud map and plane CAD shown in Fig. 5 still requires manual selection of corresponding points, which is not conducive to large-scale applications, so it needs to be studied in

▲Figure 4. Augmented reality (AR) navigation application (APP) and AR rendering result



(a) Corresponding points in point cloud map

(b) Corresponding points in CAD map

CAD: computer aided design

▲Figure 5. An example of a 2D point cloud map aligned with CAD map

the follow-up work to realize the automatic process. The proposed localization method in this paper adopts a pure vision solution. In the future, it can also be considered to combine other sensor data such as IMU, depth camera or LiDAR to further improve the localization and navigation performance. In addition, most of the current visual localization algorithms cannot be independent of the scene, and usually need to train different models on different datasets (such as training models on indoor and outdoor datasets), which brings difficulties to practical applications. For example, in the AR navigation process, image feature matching is usually performed in the cloud. Due to the diversity of the user's scene, if a scene-related localization algorithm is used, the generalization ability of the model will be insufficient, which will lead to poor localization performance. Therefore, for AR navigation, it is particularly important to enhance the generalization performance of localization algorithms and achieve scene-independent visual localization.

References

[1] LI H Q, LI L, LI Z. A review of point cloud compression [J]. ZTE technology journal, 2021, 27(1): 5 – 9. DOI: 10.12142/ZTETJ.202101003
[2] CAO M, WEI J, LYU Z, et al. Fast and robust feature tracking for 3D reconstruction [J]. Optics & laser technology, 2019, 110: 120 – 128. DOI: 10.1016/j.optlastec.2018.05.036
[3] LINDENBERGER P, SARLIN P E, LARSSON V, et al. Pixel-perfect structure-from-motion with featuremetric refinement [C]//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. IEEE, 2022: 5967 – 5977. DOI: 10.1109/ICCV48922.2021.00593
[4] YI K M, TRULLS E, LEPETIT V, et al. LIFT: learned invariant feature transform [C]//European Conference on Computer Vision. ECCV, 2016: 467 – 483. DOI: 10.1007/978-3-319-46466-4_28
[5] ZHANG X, YU F X, KARAMAN S, et al. Learning discriminative and transformation covariant local feature detectors [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 4923 – 4931. DOI: 10.1109/CVPR.2017.523
[6] ZHANG L G, RUSINKIEWICZ S. Learning to detect features in texture images [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 6325 – 6333. DOI: 10.1109/CVPR.2018.00662
[7] DETONE D, MALISIEWICZ T, RABINOVICH A. SuperPoint: self-supervised interest point detection and description [C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2018: 224 – 236. DOI: 10.1109/CVPRW.2018.00060
[8] LAGUNA A B, RIBA E, PONSA D, et al. Key.Net: keypoint detection by handcrafted and learned CNN filters [C]//International Conference on Computer Vision (ICCV). IEEE, 2020: 5835 – 5843. DOI: 10.1109/ICCV.2019.00593
[9] ONO Y, TRULLS E, FUA P, et al. LF-Net: Learning local features from images [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. ACM, 2018: 6237 – 6247. DOI: 10.5555/3327345.3327521
[10] SCHÖNBERGER J L, HARDMEIER H, SATTLER T, et al. Comparative evaluation of hand-crafted and learned local features [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 6959 – 6968. DOI: 10.1109/CVPR.2017.736
[11] WANG J, ZHOU F, WEN S L, et al. Deep metric learning with angular loss [C]//International Conference on Computer Vision. IEEE, 2017: 2612 – 2620. DOI: 10.1109/ICCV.2017.283
[12] ZAGORUYKO S, KOMODAKIS N. Learning to compare image patches via convolutional neural networks [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2015: 4353 – 4361. DOI: 10.1109/CVPR.2015.7299064
[13] LUO Z X, SHEN T W, ZHOU L, et al. ContextDesc: local descriptor augmentation with cross-modality context [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 2522 – 2531. DOI: 10.1109/CVPR.2019.00263
[14] TIAN Y R, YU X, FAN B, et al. SOSNet: second order similarity regularization for local descriptor learning [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 11008 – 11017. DOI: 10.1109/CVPR.2019.01127
[15] SARLIN P E, DETONE D, MALISIEWICZ T, et al. SuperGlue: learning feature matching with graph neural networks [C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 4937 – 4946. DOI: 10.1109/CVPR42600.2020.00499
[16] SUN J M, SHEN Z H, WANG Y A, et al. LoFTR: detector-free local feature matching with transformers [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2021: 8918 – 8927. DOI: 10.1109/CVPR46437.2021.00881
[17] SARLIN P E, CADENA C, SIEGWART R, et al. From coarse to fine: robust hierarchical localization at large scale [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 12708 – 12717. DOI: 10.1109/CVPR.2019.01300
[18] LU P, SHENG B, ZHU F. Next-generation communications technology facilitates real-time distributed cloud rendering [J]. ZTE technology journal, 2021, 27(1): 17 – 20. DOI: 10.12142/ZTETJ.202101005
[19] LU P, OUYANG X Z, GAO W W. Capacity improvement and practice of 5G industry virtual private network [J]. ZTE technology journal, 2022, 28(2): 68 – 74. DOI: 10.12142/ZTETJ.202202011

Biographies

LU Ping is the Vice President and general manager of the Industrial Digitalization Solution Department of ZTE Corporation, and Executive Deputy Director of the National Key Laboratory of Mobile Network and Mobile Multimedia Technology. His research directions include cloud computing, big data, augmented reality, and multimedia service-based technologies. He has supported and participated in major national science and technology projects and national science and technology support projects. He has published multiple papers, and authored two books.

SHENG Bin (shengbin@cs.sjtu.edu.cn) is a professor of computer science and engineering from Shanghai Jiao Tong University, China. His research directions include virtual reality and computer graphics. He has presided over two projects on the National Natural Science Foundation of China, one youth project of the National Natural Science Foundation of China, and participates in one high-technology research and development plan (the "863" plan) and one key project of the National Natural Science Foundation of China. He has published 121 papers in different journals.

SHI Wenzhe is a strategy planning engineer with ZTE Corporation, a member of the National Key Laboratory for Mobile Network and Mobile Multimedia Technology, and an engineer of XRExplore Platform Product Planning. His research interests include indoor visual AR navigation, SFM 3D reconstruction, visual SLAM, real-time cloud rendering, VR, and spatial perception.

RCache: A Read-Intensive Workload-Aware Page Cache for NVM Filesystem

TU Yaofeng[1,2], ZHU Bohong[3], YANG Hongzhang[1,2], HAN Yinjun[2], SHU Jiwu[3]

(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Tsinghua University, Beijing 100084, China)

DOI: 10.12142/ZTECOM.202301011

https://kns.cnki.net/kcms/detail/34.1294.TN.20230302.1104.002.html, published online March 2, 2023

Manuscript received: 2022-11-01

Abstract: Byte-addressable non-volatile memory (NVM), as a new participant in the storage hierarchy, gives extremely high performance in storage, which forces changes to be made on current filesystem designs. Page cache, once a significant mechanism filling the performance gap between Dynamic Random Access Memory (DRAM) and block devices, is now a liability that heavily hinders the writing performance of NVM filesystems. Therefore state-of-the-art NVM filesystems leverage the direct access (DAX) technology to bypass the page cache entirely. However, the DRAM still provides higher bandwidth than NVM, which prevents skewed read workloads from benefiting from a higher bandwidth of the DRAM and leads to sub-optimal performance for the system. In this paper, we propose RCache, a read-intensive workload-aware page cache for NVM filesystems. Different from traditional caching mechanisms where all reads go through DRAM, RCache uses a tiered page cache design, including assigning DRAM and NVM to hot and cold data separately, and reading data from both sides. To avoid copying data to DRAM in a critical path, RCache migrates data from NVM to DRAM in a background thread. Additionally, RCache manages data in DRAM in a lock-free manner for better latency and scalability. Evaluations on Intel Optane Data Center (DC) Persistent Memory Modules show that, compared with NOVA, RCache achieves 3 times higher bandwidth for read-intensive workloads and introduces little performance loss for write operations.

Keywords: storage system; file system; persistent memory

Citation (IEEE Format): Y. F. Tu, B. H. Zhu, H. Z. Yang, et al., "RCache: a read-intensive workload-aware page cache for NVM filesystem," *ZTE Communications*, vol. 21, no. 1, pp. 89–94, Mar. 2023. doi: 10.12142/ZTECOM.202301011.

1 Introduction

In 2019, Intel released the first commercially available non-volatile memory (NVM) device called Intel DC Optane Persistent Memory[1]. Compared with Dynamic Random Access Memory (DRAM), byte-addressable non-volatile memory provides comparable performance and similar interfaces (e.g., Load/Store) along with data persistence at the same time. Because of a unique combination of features, NVM has a great advantage of performance on storage systems and posts the urgent necessity of reforming the old architecture of storage systems. Refs. [2–11] re-architected the old storage systems to better accommodate NVM and significant performance boost that endorsed these design choices.

Among these novel designs, bypassing the page cache in kernel space is a popular choice. The page cache in Linux is used to be an effective mechanism to shorten the performance gap between DRAM and block devices. Since NVM has a close performance to the DRAM, the page cache itself posts severe performance loss to the NVM filesystem, because the page cache introduces extra data copy at every file operation and leads to write amplification on NVM. Therefore, the legacy page cache in the Linux kernel has become a liability for the NVM system. For the above reasons, recent work simply deployed the DAX[12] technology to bypass the page cache entirely[12–17]. With the DAX technology, NVM filesystems access the address space of NVM directly, without the necessity of filling the page cache first, which reduces the latency of filesystem operations significantly.

However, although NVM achieves bandwidth and latency at the same order of magnitude as DRAM, DRAM still provides bandwidth several times higher than NVM and fairly lower latency than NVM. Therefore, the DAX approach reduces extra data copy and achieves fast write performance at the cost of cached read, especially for read-intensive workloads[18–20]. The page cache provides benefits for reading but has severe performance impacts on writing because of the extra data copy and write amplification. And the DAX approach is efficient for writing due to direct access to NVM but fails to utilize DRAM bandwidth for reading. Therefore, in order to utilize DRAM

This paper was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-20181128026.

bandwidth and avoid extra data copy and write amplifications, the page cache should be redesigned to allow both direct access and cached read.

In this paper, we propose RCache, a read-intensive workload-aware page cache for the NVM filesystem. RCache aims to provide fast read performance for read-intensive workloads and avoid introducing significant performance loss for write operations at the same time. To achieve this, RCache assigns DRAM and NVM to hot and cold data separately, and reads data from both sides. Our major contributions are summarized as follows.

• We propose a read-intensive workload-aware page cache design for the NVM filesystem. RCache uses a tired page cache design, including reading hot data from DRAM and accessing cold data directly from NVM to utilize DRAM bandwidth for reading and preserving fast write performance. In addition, RCache offloads data copy from NVM to DRAM and to a background thread, in order to remove a major setback of caching mechanism from the critical path.

• RCache introduces a hash-based page cache design to manage the page cache in a lock-free manner using atomic instructions for better scalability.

• We implement RCache and evaluate it on servers with Intel DC Persistent Memory Modules. Experimental results show that RCache effectively utilizes the bandwidth of DRAM with few performance cost to manage the page cache and outperforms the state-of-the-art DAX filesystem under read-intensive workloads.

2 Background and Motivation

2.1 Non-Volatile Memory

Byte-addressable NVM technologies, including Phase-change Memory (PCM) [22-24], ReRAM, and Memristor[21], have been intensively studied in recent years. These NVMs provide comparable performance and a similar interface as the DRAM, while persisting data after power is off like block devices. Therefore, NVMs are promising candidates for providing persistent storage ability at the main memory level. Recently, Intel has released Optane DC Persistent Memory Modules (DCPMM)[1], which is the first commercially available persistent memory product. Currently, new products come in three capacities: 128 GB, 256 GB, and 512 GB. Previous studies show that a single DCPMM provides bandwidths at 6.6 GB/s and 2.3 GB/s at most for read/write. Note that these bandwidth have the same order of magnitudes comparable to the DRAM but is a lot lower than the DRAM[25].

2.2 Page Cache and DAX Filesystem

Page cache is an important component in a Linux kernel filesystem. In brief, the page cache consists of a bunch of pages in DRAM and the corresponding metadata structures. The page cache is only accessed by the operating system in the context of a filesystem call and acts as a transparent layer to user applications. For a write system call, the operating system writes data on pages in the page cache, which cannot guarantee the persistence of the data. To guarantee the persistence of the data, the operating system needs to flush all data pages in the page cache to the storage devices, probably within an fsync system call. For a read system call, the operating system first reads data from the page cache; if not present, the operating system further reads data from the storage devices. Note that this may involve loading data into the page cache depending on the implementation. In the current implementation, the operating system maintains an individual radix tree for each opened file.

As for the DAX filesystem, note that the page cache is extremely useful for block devices with much higher access latency than DRAM, but not suitable for the NVM devices with comparable access latency to DRAM. As mentioned before, to ensure data persistence, the user must issue an fsync system call after a write system call. This brings substantial access latency to persisting data in an NVM filesystem. Therefore, the state-of-the-art NVM filesystems leverage the DAX technology to bypass the page cache entirely and achieve instant persistence immediately when the write system call returns. In a DAX filesystem, read/write system call does not access the page cache at all, instead, data are loaded/stored from/to the NVM respectively using a memory interface. The DAX technology reduces extra data copy and accomplishes lower-cost data persistence.

2.3 Issue of DAX and Page Cache

The performance of NVM is close to that of DRAM but not equal to it. We measure the read and write latency of two different filesystems (NOVA[17] and EXT4[26]) representing two different mechanisms (DAX and Page Cache). Fig. 1(a) shows that the read latency of the DAX is much higher than the page cache (4 kB sequential read). Fig. 1(b) shows that the write latency of the DAX is much lower than the page cache (4 kB sequential write).

To sum up, the DAX technology prevents the read operations from benefiting a much higher bandwidth of DRAM in the NVM filesystem, and the presence of the page cache significantly increases the latency of write operations with immediate data persistence. To overcome this, the page cache mechanism needs to be redesigned.

3 Rcache Design

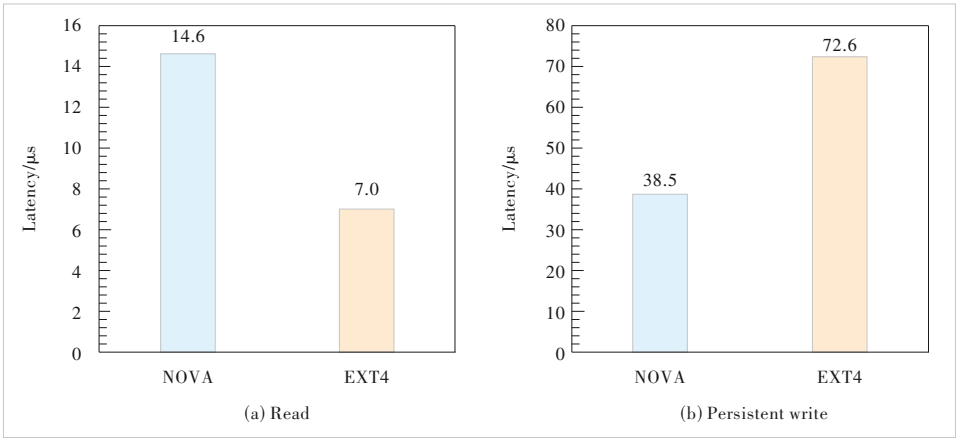
3.1 Overview

We build RCache for servers with non-volatile memory to accelerate read-intensive workloads. In order to benefit from the DRAM bandwidth for read operations but not to induce notable latency for data persistence, we build RCache, a read-intensive workload-aware page cache for the NVM filesystem.

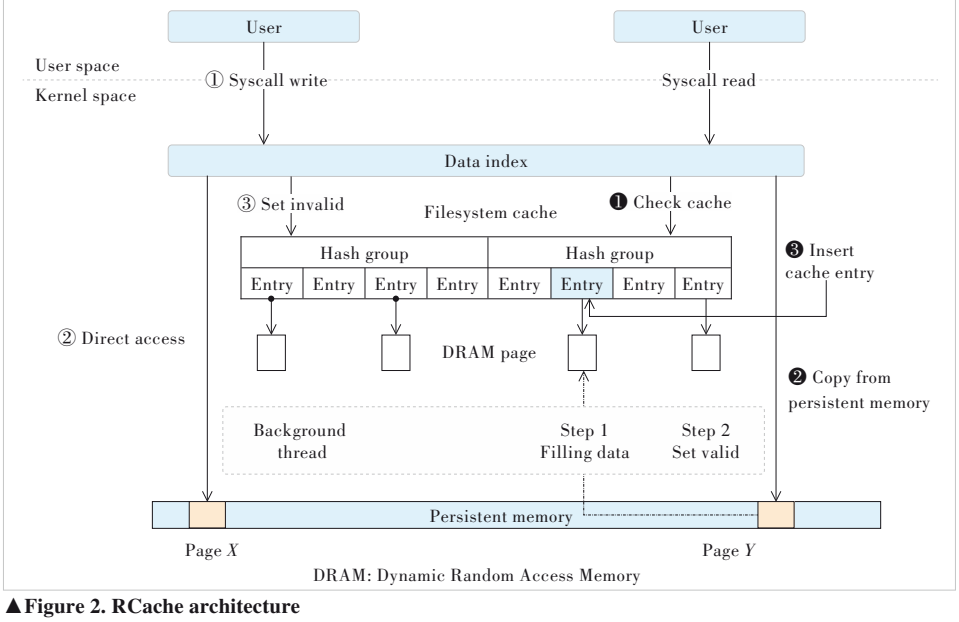
1) RCache assigns DRAM and NVM to hot and cold data separately, and allows cached read and direct read from NVM to coexist. Furthermore, RCache offloads data copy to a background thread to alleviate the pressure of the critical path.

2) In addition, RCache deploys a lock-free page cache using hash-table to further reduce the performance cost of cache coherence management.

The architecture of RCache is described in Fig. 2. RCache keeps an individual cache structure for each opened file. The page cache consists of a bunch of DRAM pages and a cache entry table containing a certain number of cache entries in the DRAM. A cache entry represents a DRAM page. It carries necessary information for RCache to manage the cache and navigate data given a logical block number. As shown in Fig. 3, a cache entry carries a validation flag to indicate the status of this cache entry, a timestamp for the least recently used (LRU) algorithm, a Blocknr to indicate the logical block number that the entry represents, a DRAM page that is a pointer points to the actual cache page in DRAM, and an NVM page that is a pointer points to the actual data page in NVM.



▲Figure 1. Performance comparison between different hardware and different filesystem settings



▲Figure 2. RCache architecture

3.2 Tiered Page Cache Design

As shown in Fig. 2, the page cache is accessed in two contexts: a read/write system call and a background thread.

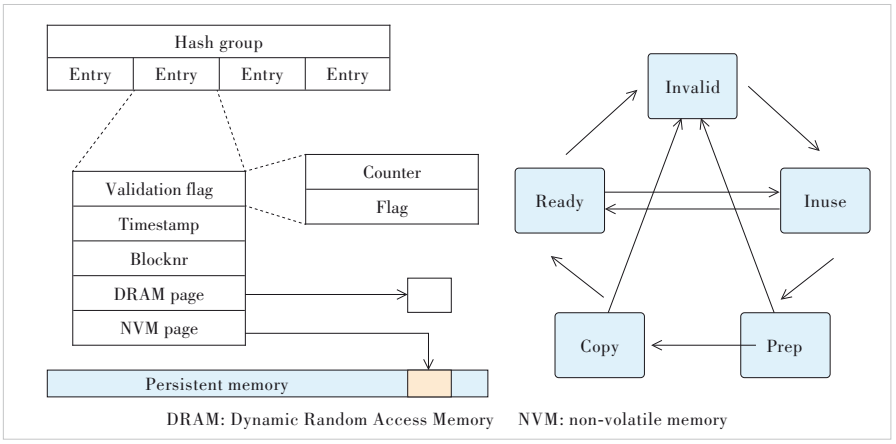
For a read operation, the operating system accesses the page cache first. If the data required by the user are present and valid in the page cache, the operating system copies data directly from the cached page in the DRAM to the user's buffer; if a cache miss happens, the operating system falls back to the legacy procedure where the operating system reads data directly from the NVM and inserts the newly read data to the page cache. For cache insertion, since reading all the data blocks into the page cache introduces extra data copy and then leads to higher latency, RCache only inserts a small cache entry carrying a pointer to the physical block to the page cache instead of the actual data blocks.

For a write operation, the operating system needs to invalidate all cached pages affected by this write operation before returned to users. We further explain why the invalidation procedure is light weight in Section 3.3.

RCache depends on a background kernel thread to finish the management of the cache. As described above, in the read operation, RCache only inserts cache entries to the page cache. In the context background thread, once a pending cache entry is discovered, RCache first allocates a DRAM page to cache data, and then copies data from the NVM block to the DRAM page according to the cache entry. At last, RCache declares the validity of the cache entry by switching the validation flag atomically. Note that only when RCache updates the validation flag in the cache entry to validation, the cache entry is available for read/write context.

3.3 Lock-Free Cache Management

The decoupled cache mechanism splits the cache management into two separating and concurrent contexts, which makes coordinating across all units more expensive since it leads to more cross-core communications. Therefore, RCache



▲Figure 3. Cache structure and status shifting paradigm

deploys a lock-free cache management procedure to minimize the impact. First, RCache operates cache entries by manipulating the validation flag atomicity using Compare-and-Swap (CAS) instructions. In the current implementation, a cache entry switches among five states using the Compare-and-Swap instruction. Fig. 3 depicts the transition diagram among these five states. At the initial point, all cache entries are invalid. To insert a cache entry, RCache first acquires control of a candidate entry by setting the validation flag of this entry to "In use" atomically using CAS, which prevents other threads from operating on this entry. Then, RCache fills necessary information (e.g. the block number and the NVM page pointer) and changes the status to "Prep", which tells the background thread that this entry has all information needed and is ready for data copy. From the background thread view, before copying data from persistent memory to DRAM, the background thread first sets the status of a cache entry to "Copy", then the background thread initiates a data copy procedure. When the data copy completes, the background thread sets the status of a cache entry to "Ready" by using CAS instruction operating on the validation flag, and, only at this point, the cache is available for read operations. To write data into a certain page, if cache hits, RCache needs to invalidate the cache entry representing this page by switching the status to "Invalid" by CAS, and the validation flag of the entry to "Invalid". Note that RCache never invalidates an "In use" cache entry, because the "In use" status only exists in the context of a read syscall. Since the file is locked up in write operations, this situation never happens. To read data from a cache entry, RCache first switches the status from "Ready" to "In use" using CAS, then copies data from the DRAM page to user buffer, and at last, changes the status back to "Ready". However, this leads to an inconsistent status where users might be given wrong data, since there might be several threads reading data from the cache entry concurrently. Therefore, RCache incarnates an additional counter in the validation flag, when a reader wants to read this cache, it must increase this counter; and when a reader finishes reading, it must decrease the counter. Therefore, only the last reader can switch the status back to "Ready".

3.4 Implementation

We implement RCache on NOVA, a state-of-the-art NVM filesystem developed with the DAX technology. We keep the metadata and data layout in NOVA intact, and add extra logic for managing the cache in the context of read/write procedure. We launch the background thread in kernel at the mount phase, and reclaim this thread during the unmount phase. To tackle the hotness of a block, we extend the block index in NOVA, and add an extra counter to each leaf node of the radix tree. We insert a block into the cache only when it is accessed more times than a threshold in a time window. The threshold and the time window are predefined.

4 Evaluation

In this section, we first evaluate RCache's read/write latency, then we evaluate the read performance under read-intensive workload, and at last, we evaluate the read performance under a skewed read-intensive workload.

4.1 Experimental Setup

We implement RCache and evaluate the performance of RCache on the server with Intel Optane DCPMM. The server has 192 GB DRAM and two Intel Xeon Gold 6 240 M processors (2.6 GHz, 36 cores per processor) and 1 536 GB Intel Optane DC Persistent Memory Modules (6×256 GB). Because cross-non-uniform memory access (NUMA) traffic has a huge impact on performance[27], throughout the entire evaluation, we only utilize NVMs on one NUMA node to deploy RCache and other file systems (e.g., only 768 GB NVMs on this server). The server is running Ubuntu18.04 with Linux Kernel 4.15.

Table 1 lists file systems for comparison. We build all filesystems on the same NVM device with a PMem driver. For EXT4, we build it following the traditional procedure with a page cache involved. For both NOVA and RCache, since RCache shares most of the filesystem routines with NOVA, we deploy both of them on an NVM device with a PMem driver and DAX enabled.

For a latency test, we use custom micro benchmarks and Fxmark[28] for bandwidth evaluation. Fxmark is a benchmark de-

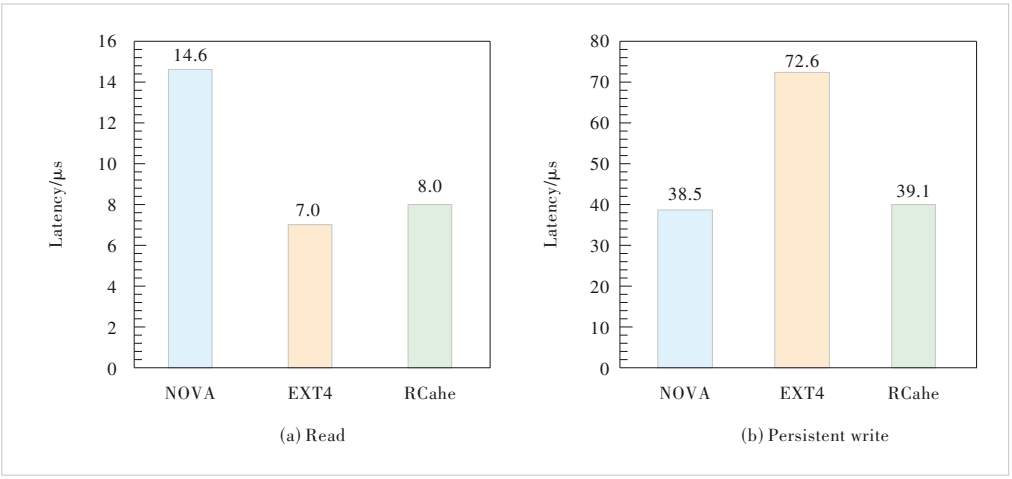
▼Table 1. Evaluated file systems

File System	Description
NOVA[17]	A state-of-the-art NVM filesystem in the kernel. NOVA adopts conventional log-structured file system techniques and optimizes file systems for hybrid memory systems to maximize performance
EXT-4[26]	A well-known kernel file system in Linux

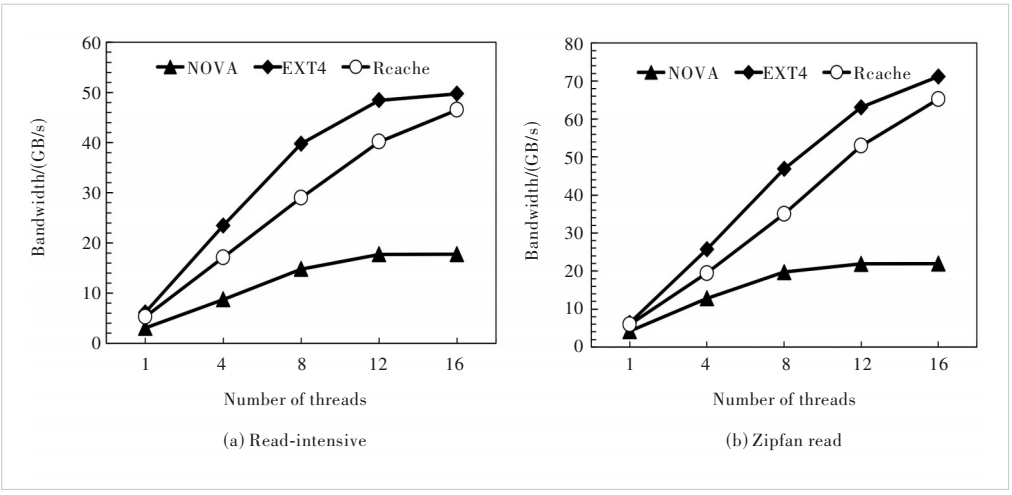
signed to evaluate the scalability of file systems. In this evaluation, we use three sub-benchmarks, namely DRBL, DRBM and DWAL, in Fxmark.

4.2 Overall Performance

To evaluate the read/write performance, we use a custom micro-benchmark. All evaluation on each filesystem spawns only one thread. We first create a file with 64 MB, then issue 4 kB read/write data with 100 000 requests, and finally calculate the average latency. Since EXT4 does not ensure data persistency in the write system call, we issue another fsync after each write system call to preserve data persistency. Fig. 4 shows the read/write latency for three evaluated filesystems.



▲Figure 4. Read and write latency of different filesystems



▲Figure 5. Read bandwidth under the read-intensive workload of different filesystems

For read operations, EXT4 shows the lowest latency, and the latency of RCache is close to that of EXT4 and much lower than that of NOVA. This is because RCache utilizes the DRAM bandwidth to accelerate read.

To evaluate the read bandwidth under a read-intensive workload, we use sub-benchmark DRBL from Fxmark. DRBL first creates a 64 MB file for each thread and then issues sequence read operation to the filesystem. We conduct the evaluation for 20 s. If a read operation reaches the tail of the file, the next read operation is set at the beginning of the file. From Fig. 5(a) we can see that the RCache shows much better read performance than NOVA and close to that of EXT4.

4.3 Read Performance Under Skewness

We evaluate the read performance under the skewed workload. We modify the DRBL benchmark instead of reading files sequentially, where each thread post-read request at an offset is controlled by a random variable that follows the normal distribution. Fig. 5(b) shows that, both EXT4 and RCache achieve even better performance than that in Fig. 5(a). This is because under the skewed workload, the hot pages are more likely to be stored in the L3 cache and therefore end up with better performance. On the other hand, since NOVA does not utilize DRAM for better read performance, the read bandwidth achieved is much lower than that of EXT4 or RCache.

5 Conclusions

Traditional page cache in the Linux kernel can benefit read workload but cannot fit into an NVM filesystem because it causes extra data copy and write amplification. By bypassing the page cache, the DAX filesystem achieves better write performance but gives up the opportunity of cached read. Therefore, in this paper, we propose a read-intensive workload-aware page cache for NVM filesystems. RCache uses a tiered page cache design, including assigning DRAM and NVM to hot and cold data separately, and reading data from both sides. Therefore, cached read and direct access can coexist. In addition, to avoid copying data to DRAM in a critical path, RCache migrates data from NVM to DRAM in a background thread. Furthermore, RCache manages data in DRAM in a lock-free manner for better latency and scalability. Evaluations on Intel Optane DC Persistent Memory Modules show that compared with NOVA, RCache has 3 times higher bandwidth for read-intensive workloads and introduces little performance loss to write operations.

References

[1] Intel. Intel Optane DC Persistent Memory [EB/OL]. [2022-11-01]. https://www.intel. com/content/www/us/en/products/memory-storage/optane-dcpersistent-memory.html
[2] SHU J W, CHEN Y M, WANG Q, et al. TH-DPMS: design and implementation of an RDMA-enabled distributed persistent memory storage system [J]. ACM transactions on storage, 2020, 16(4): 1 - 31. DOI: 10.1145/3412852
[3] CHEN Y M, LU Y Y, SHU J W. Scalable RDMA RPC on reliable connection with efficient resource sharing [C]//Proceedings of the Fourteenth EuroSys Conference. ACM, 2019. DOI: 10.1145/3302424.3303968
[4] CHEN Y M, LU Y Y, YANG F, et al. FlatStore: an efficient log-structured key-value storage engine for persistent memory [C]//Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2020: 1077 - 1091. DOI: 10.1145/3373376.3378515
[5] LU Y Y, SHU J W, CHEN Y M, et al. Octopus: an RDMA-enabled Distribute Persistent Memory File System [C]//USENIX Annual Technical Conference. IEEE, 2017: 773 - 785
[6] ZHU B H, CHEN Y M, WANG Q, et al. Octopus+: an RDMA-Enabled Distributed Persistent Memory File System [J]. ACM Transactions on Storage, 2021, 17 (3): 1 - 25
[7] COBURN J, CAULFIELD A M, AKEL A, et al. NV-Heaps: making persistent objects fast and safe with next-generation, non-volatile memories [C]//Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2011: 105 - 118. DOI: 10.1145/1950365.1950380
[8] HONDA M, EGGERT L, SANTRY D. PASTE: network stacks must integrate with NVMM abstractions [C]//Proceedings of the 15th ACM Workshop on Hot Topics in Networks. ACM, 2016: 183 - 189. DOI: 10.1145/3005745.3005761
[9] NARAYANAN D, HODSON O. Whole-system persistence [C]//Proceedings of the 17th International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2012: 401 - 410. DOI: 10.1145/2150976.2151018
[10] VOLOS H, TACK A J, SWIFT M M. Mnemosyne: lightweight persistent memory [C]//Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2011: 91 - 104. DOI: 10.1145/1950365.1950379
[11] ZHANG Y Y, YANG J, MEMARIPOUR A, et al. Mojim: a reliable and highly-available non-volatile memory system [C]//Proceedings of the 20th International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2015: 3 - 18. DOI: 10.1145/2694344.2694370
[12] DULLOOR S R, KUMAR S, KESHAVAMURTHY A, et al. System software for persistent memory [C]//Proceedings of the 9th European Conference on Computer Systems. ACM, 2014: 15 - 30. DOI: 10.1145/2592798.2592814
[13] CHEN Y M, LU Y Y, ZHU B H, et al. 2021. Scalable Persistent Memory File System with Kernel-Userspace Collaboration [C]//USENIX Conference on File and Storage Technologies (FAST 21). FAST, 2021: 81 - 95
[14] DONG M K, BU H, YI J F, et al. Performance and protection in the ZoFS user-space NVM file system [C]//Proceedings of the 27th ACM Symposium on Operating Systems Principles. ACM, 2019: 478 - 493. DOI: 10.1145/3341301.3359637
[15] KADEKODI R, LEE S K, KASHYAP S, et al. SplitFS: reducing software overhead in file systems for persistent memory [C]//Proceedings of the 27th ACM Symposium on Operating Systems Principles. ACM, 2019: 494 - 508. DOI: 10.1145/3341301.3359631
[16] OU J X, SHU J W, LU Y Y. A high performance file system for non-volatile main memory [C]//Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016: 1 - 16 DOI: 10.1145/2901318.2901324
[17] XU J, SWANSON S. NOVA: a log-structured file system for hybrid volatile/non-volatile main memories [C]//The 14th USENIX Conference on File and Storage Technologies. ACM, 2016: 323 - 338. DOI: 10.5555/2930583.2930608
[18] ATIKOGLU B, XU Y H, FRACHTENBERG E, et al. Workload analysis of a large-scale key-value store [C]//Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems. ACM, 2012: 53 - 64. DOI: 10.1145/2254756.2254766
[19] LI J L, NELSON J, MICHAEL E, et al. Pegasus: tolerating skewed workloads in distributed storage with in-network coherence directories [C]//Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation. ACM, 2020: 387 - 406. DOI: 10.5555/3488766.3488788
[20] YANG J C, YUE Y, RASHMI K V. A large scale analysis of hundreds of in-memory cache clusters at Twitter [C]//The 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). ACM, 2014: 191 - 208
[21] STRUKOV D B, SNIDER G S, STEWART D R, et al. The missing memristor found [J]. Nature, 2008, 453(7191): 80 - 83. DOI: 10.1038/nature06932
[22] LEE B C, IPEK E, MUTLU O, et al. Architecting phase change memory as a scalable dram alternative [C]//Proceedings of the 36th Annual International Symposium on Computer Architecture. ACM, 2009: 2 - 13. DOI: 10.1145/1555754.1555758
[23] QURESHI M K, SRINIVASAN V, RIVERS J A. Scalable high performance main memory system using phase-change memory technology [C]//Proceedings of the 36th Annual International Symposium on Computer Architecture. ACM, 2009: 24 - 33. DOI: 10.1145/1555754.1555760
[24] ZHOU P, ZHAO B, YANG J, et al. A durable and energy efficient main memory using phase change memory technology [C]//Proceedings of the 36th Annual International Symposium on Computer Architecture. ACM, 2009: 14 - 23. DOI: 10.1145/1555754.1555759
[25] IZRAELEVITZ J, YANG J, ZHANG L, et al. Basic performance measurements of the intel optane DC persistent memory module [EB/OL]. [2022-03-14]. https://arxiv.org/abs/1903.05714
[26] EXT4. EXT4 (and EXT2/EXT3) Wiki [EB/OL]. (2016-09-20) [2022-03-14]. https://ext4.wiki.kernel.org/
[27] YANG J, KIM J, HOSEINZADEH M, et al. An empirical guide to the behavior and use of scalable persistent memory [C]//The 18th Conference on File and Storage Technologies. ACM, 2020: 169 - 182
[28] MIN C W, KASHYAP S, MAASS S, et al. Understanding manycore scalability of file systems [C]//USENIX Annual Technical Conference. ACM, 2016: 71 - 85

Biographies

TU Yaofeng received his PhD degree from Nanjing University of Aeronautics and Astronautics, China. He is a senior expert at ZTE Corporation. His research interests include big data, database and machine learning.

ZHU Bohong (zhubohong18@mails.tsinghua.edu.cn) received his master's degree from Tsinghua University, China in 2018. He is currently studying in the School of Informatics, Xiamen University, China for his PhD degree. His research interests include filesystems, memory storage and distributed systems.

YANG Hongzhang received his PhD degree from Peking University, China. He is an engineer at ZTE Corporation. His research interests include file systems, persistent memory and storage reliability.

HAN Yinjun received his master's degree from Nanjing University of Science and Technology, China. He is a senior engineer at ZTE Corporation. His research interests include distributed file systems, RDMA and persistent memory.

SHU Jiwu received his PhD degree in computer science from Nanjing University, China in 1998. He is currently the dean of the School of Informatics, Xiamen University, China and a professor in the Department of Computer Science and Technology, Tsinghua University, China. His research interests include network storage systems, non-volatile memory systems and technologies, reliability for storage systems, and parallel/distributed processing technologies. He is a Changjiang Professor, CCF Fellow, and IEEE Fellow.

The 1st Youth Expert Committee
for Promoting Industry-University-Institute Cooperation

Director **CHEN Wei,** Beijing Jiaotong University

Deputy Director **QIN Xiaoqi,** Beijing University of Posts and Telecommunications

LU Dan, Magazine House of ZTE Communications

Members (Surname in Alphabetical Order)

CAO Jin	Xidian University
CHEN Li	University of Science and Technology of China
CHEN Qimei	Wuhan University
CHEN Shuyi	Harbin Institute of Technology
CHEN Wei	Beijing Jiaotong University
GUAN Ke	Beijing Jiaotong University
HAN Kaifeng	China Academy of Information and Communications Technology
HE Zi	Nanjing University of Science and Technology
HU Jie	University of Electronic Science and Technology of China
HUANG Chen	Purple Mountain Laboratories
LI Ang	Xi'an Jiaotong University
LIU Chunsen	Fudan University
LIU Fan	Southern University of Science and Technology
LIU Junyu	Xidian University
LU Dan	Magazine House of ZTE Communications
LU Youyou	Tsinghua University
NING Zhaolong	Chongqing University of Posts and Telecommunications
QI Liang	Shanghai Jiao Tong University
QIN Xiaoqi	Beijing University of Posts and Telecommunications
QIN Zhijin	Tsinghua University
SHI Yinghuan	Nanjing University
WANG Jingjing	Beihang University
WANG Xinggang	Huazhong University of Science and Technology
WANG Yongqiang	Tianjin University
WEN Miaowen	South China University of Technology
WU Yongpeng	Shanghai Jiao Tong University
XIA Wenchao	Nanjing University of Posts and Telecommunications
XU Mengwei	Beijing University of Posts and Telecommunications
XU Tianheng	Shanghai Advanced Research Institute, Chinese Academy of Sciences
YANG Chuanchuan	Peking University
YIN Haifan	Huazhong University of Science and Technology
YU Jihong	Beijing Institute of Technology
ZHANG Jiao	Beijing University of Posts and Telecommunications
ZHANG Yuchao	Beijing University of Posts and Telecommunications
ZHANG Jiayi	Beijing Jiaotong University
ZHAO Yuda	Zhejiang University
ZHOU Yi	Southwest Jiaotong University
ZHU Bingcheng	Southeast University

ZTE COMMUNICATIONS
中兴通讯技术(英文版)

ZTE Communications has been indexed in the following databases:

- Abstract Journal
- Cambridge Scientific Abstracts (CSA)
- China Science and Technology Journal Database
- Chinese Journal Fulltext Databases
- Index of Copurnicus
- Ulrich's Periodicals Directory
- Wanfang Data
- WJCI 2022

ZTE COMMUNICATIONS
Vol. 21 No. 1 (Issue 82)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Publishing Group

Sponsored by:
Time Publishing and Media Co., Ltd.
Shenzhen Guangyu Aerospace Industry Co., Ltd.

Published by:
Anhui Science & Technology Publishing House

Edited and Circulated (Home and Abroad) by:
Magazine House of ZTE Communications

Editorial Correspondence:
Add: 12F Kaixuan Building, 329 Jinzhai Road,
Hefei 230061, P. R. China
Tel: +86-551-65533356
Email: magazine@zte.com.cn
Website: http://zte.magtechjournal.com

Annual Subscription: RMB 120
Printed by:
Hefei Tiancai Color Printing Company
Publication Date: March 25, 2023

China Standard Serial Number: ISSN 1673-5188
CN 34-1294/TN