Federated Learning Framework with Personalized Model Compression and Privacy Protection

Chenlin Ding[*†], Mingjun Xiao[*†], Yin Xu[*†§], Jie Wu[‡]
[*]School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
[†]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, China
[‡]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[§]Corresponding Author: yinxu@ustc.edu.cn
Email: {dingchenlin@mail., xiaomj@}ustc.edu.cn, jiewu@temple.edu

***Abstract*—Federated learning faces significant challenges in balancing communication efficiency, model accuracy, and privacy protection. While model compression effectively reduces communication overhead, existing approaches typically adopt a fixed compression rate, failing to dynamically balance compression efficiency and model performance while often overlooking privacy concerns. To address these issues, we propose FedCP—a Federated learning framework with personalized model Compression and Privacy protection. This novel framework integrates a Personalized Compression Mechanism (PCM) and an Optimized Piecewise noise Mechanism (OPM). PCM dynamically adjusts the model compression rate based on clients' privacy budgets and communication costs, achieving an optimal trade-off between communication overhead and model accuracy. Since model compression inherently provides a degree of privacy protection, OPM further refines the noise injection strategy through mathematical formulations, optimizing the noise addition process for larger privacy budgets and enhancing model performance. Experimental results on multiple real-world datasets demonstrate that FedCP outperforms existing baseline methods in both model accuracy and communication efficiency while ensuring rigorous privacy protection. This paper presents an effective solution to communication optimization in federated learning and introduces a more advanced privacy-preserving mechanism with significant theoretical and practical implications.**

***Index Terms*—Federated Learning, Personalized Compression Mechanism, Optimized Privacy Protection, Stackelberg Game.**

I. Introduction

Federated Learning (FL) is a distributed machine learning paradigm that enables multiple devices or data sources to collaboratively train models without centralizing data on a central server. Its core advantage lies in preserving user privacy through local data processing [1], while simultaneously enhancing model generalization by leveraging diverse distributed datasets. Although FL has been widely adopted in various fields such as retail, finance, and healthcare [2], its practical deployment still faces significant challenges, particularly in terms of communication efficiency and privacy protection.

In a typical FL framework, the server is responsible for coordinating clients, including collecting, aggregating, and distributing model parameters to iteratively refine the global model. However, this process often requires multiple rounds of communication, resulting in substantial communication overhead and severely impacting training efficiency. To address this, existing research primarily relies on quantization [3], [4], [5], [6] and sparsification [7], [8], [9], [10] techniques, which compress client-uploaded model parameters to reduce communication costs. However, most studies adopt fixed compression rates, overlooking key factors such as privacy budgets, data heterogeneity, and communication costs. This limitation prevents clients from optimizing compression rates based on their individual characteristics, leading to inefficient use of communication resources and potential degradation of model convergence speed. Furthermore, model compression may negatively impact performance, especially in scenarios with high data complexity or stringent task requirements. Therefore, achieving a fine-grained trade-off between compression rate and model accuracy remains a critical challenge.

Privacy protection is another crucial aspect of FL. Differential Privacy (DP) has emerged as a mainstream technique for mitigating privacy threats in FL systems [11], [12], [13], [14]. Integrating model compression can effectively reduce the amount of shared data in each iteration, thereby enhancing privacy protection at the transmission level. This reduction in potential information leakage enables clients to be allocated higher privacy budgets, ensuring stronger privacy guarantees while maintaining model utility. However, designing an efficient and personalized compression mechanism that balances privacy protection, model performance, and communication efficiency remains a significant challenge. Specifically, these challenges include optimizing the trade-off between compression rate and model accuracy, adapting to client heterogeneity, and ensuring robust privacy protection.

To address the above-mentioned challenges, this paper proposes a federated learning framework with personalized model compression and privacy protection (FedCP). First, we design a Personalized Compression Mechanism (PCM) that dynamically determines the compression rate for each client to balance communication efficiency and global model performance. Then, we formulate the problem of determining the optimal compression rate for each client as a Stackelberg game, where the server acts as the leader and the clients act

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62436010, 62172386, and 61872330, in part by the Natural Science Foundation of Jiangsu Province in China under Grants BK20231212 and BK20191194, and in part by the National Natural Science Foundation of USA under Grants CNS 2128378 and CNS 2107014.

as the followers, each independently optimizing their utility function. The server's utility depends on the improvement of the global model and the rewards allocated to the clients, while each client's utility is influenced by the received rewards, communication costs, privacy loss, and computational overhead. By deriving the unique equilibrium for this Stackelberg game, we can determine the optimal payment strategy for the server and the optimal compression strategies for clients to ensure system convergence. Additionally, we propose an Optimized Piecewise Noise Injection Mechanism (OPM) to enhance privacy protection in high-privacy-budget scenarios. Extensive experiments on real-world datasets confirm that FedCP effectively reduces communication costs, strengthens privacy protection, and maintains model stability in resource-constrained environments.

The key contributions of this paper are listed as follows:

1) We propose FedCP, a novel FL framework that jointly optimizes communication efficiency, model performance, and privacy protection while integrating game-theoretic compression optimization.

2) We propose PCM, which enables clients to dynamically adjust their compression rates based on their individual characteristics, thus achieving an optimal balance between communication overhead and global model performance. Furthermore, for higher privacy budgets, we introduce OPM to further enhance the model performance by combining with PCM.

3) We conduct extensive experiments on multiple real-world datasets, demonstrating that FedCP outperforms existing baseline methods in communication efficiency, privacy protection strength, and model performance, providing a promising direction for practical FL deployment.

II. Related Work

Communication efficiency and privacy protection are two critical challenges in FL that require urgent solutions. In recent years, numerous innovative approaches have been proposed to address these challenges.

Regarding communication efficiency optimization, researchers have focused on quantization and sparsification techniques to reduce communication overhead. Quantization reduces data precision to minimize transmission costs, with notable works including QSGD [3], which introduces a gradient quantization encoding scheme, and SignSGD [6], which employs a gradient sign transmission mechanism. Sparsification, on the other hand, optimizes communication by selectively transmitting key gradients. Representative methods include the Top-k gradient selection strategy [8] and fixed sparsity-based gradient transmission [10]. Notably, Gaia [16] introduces a dynamic communication content selection mechanism, while STC [17] achieves bidirectional data compression through a joint optimization of sparsification and ternarization.With the rapid advancement of deep learning models, compression algorithms tailored for specific models have also made significant progress. For instance, Laptop-diff [18] proposes a structured pruning method for diffusion models, DepGraph [19] develops a dependency-graph-based general model optimization framework, and LLM-Pruner [20] achieves efficient compression and performance retention for large-scale language models.

In terms of privacy protection [21], existing research primarily focuses on techniques such as Homomorphic Encryption (HE) [22], Secure Multi-party Computation (SMC) [23], and DP. Among these, DP has gained significant attention due to its lower computational and communication overhead. Wei et al. [12] proposed a privacy-preserving mechanism based on gradient clipping and Laplacian and Gaussian noise injection. Meanwhile, the LDP-FL framework [14] further enhances data perturbation effectiveness through adaptive range adjustment and parameter transformation. There has also been notable progress in leveraging incentive mechanisms to ensure privacy protection. For instance, one study integrates reputation and contract theory to design an effective incentive mechanism that encourages high-quality local data owners to participate in model training while protecting their privacy [30]. Zhang et al. [31] proposed an incentive mechanism based on reputation and reverse auction theory, which selects and rewards high-performing participants by considering their reputation and bidding behavior under limited privacy budgets. Xu et al. [32] introduced a personalized privacy-preserving mechanism for crowdsourced federated learning, modeling the personalized privacy budget optimization as a two-stage Stackelberg game to determine the optimal privacy budget for each participant.

However, existing research on communication optimization in federated learning often overlooks the heterogeneity of clients, focusing primarily on model compression techniques without effectively balancing privacy protection, model accuracy, and overall communication overhead. To address this limitation, we propose an innovative Personalized Compression Framework—FedCP. This framework enables clients to dynamically adjust their compression rates based on individualized factors such as privacy budget and communication cost, while ensuring privacy protection. As a result, FedCP achieves efficient global model convergence, significantly reduces communication overhead, and enhances privacy protection strength.

III. System Model

The FedCP framework is briefly described as shown in Fig. 1. Consider a federated learning system consisting of a parameter server, M edge nodes, and N clients. The server collects model parameters from the N clients for model aggregation, which results in the global model, with the client set denoted as $\mathcal{N} = \{1, 2, \cdots, N\}$. The set of edge nodes is denoted by $\mathcal{M} = \{1, \ldots, j, \ldots, M\}$, where the edge nodes establish a coordination mechanism between the server and the clients to enhance client anonymity and assist the server in quickly aggregating local models and propagating the global model. Each client $i \in \mathcal{N}$ collects data through mobile smart devices and stores it in its local dataset D_i. The system operates in time slices, with the entire process divided into T rounds. The joint training process in round $t \in \mathcal{T} = \{1, 2, \cdots, T\}$ in FedCP includes the following steps:

TABLE I: Symbols and their descriptions

Symbol	Description
i, N	Client i and the client set.
j, M	Edge node j and the edge node set.
N, M	The number of clients and edge nodes.
$c_i^{cm}, c_i^{pv}, c_i^{cp}$	Communication cost, privacy cost, and computation cost.
$\rho_i^{cm}, \rho_i^{pv}, \rho_i^{cp}$	Unit communication cost, unit privacy cost, and unit computation cost.
γ_i^t, R_i^t	The model compression rate γ_i^t and the payment reward R_i^t for client i.
γ_i^{t*}, R^{t*}	The personalized optimal model compression rate γ_i^{t*} and the server's optimal payment R^{t*} for client i.
$T, \mathbf{w}_k$	The interaction rounds and the positive adjustment factor.
U_i^t, U^t	The utility functions for client i and the server.
$\mathbf{w}_t^i, \tilde{\mathbf{w}}_t^i$	The model parameters uploaded by client i, before and after noise addition.

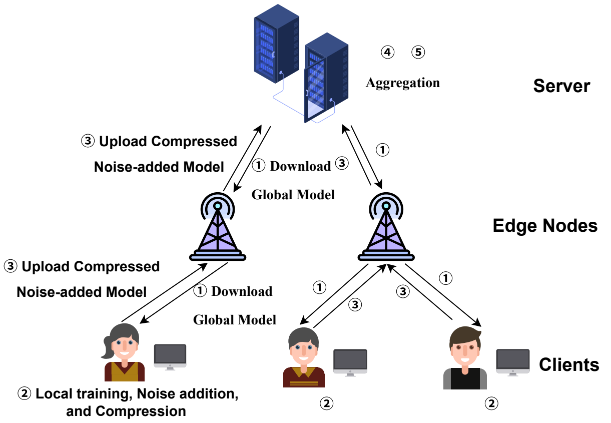


Fig. 1: Overview of the FedCP Framework.

1) Server broadcasts the global model: The server receives the global model from the previous round, $\mathbf{w}_{t-1}$, and broadcasts it to all clients along with the payment rules.

2) Local training: Each client i receives the global model $\mathbf{w}_{t-1}$ from the server and performs local training to obtain the model parameters $\mathbf{w}_{t0}^i$. To improve communication efficiency, client i first applies the PCM mechanism, which determines a personalized compression rate γ_i^t based on the server's payment reward R^t, and compresses the local model to obtain $\mathbf{w}_t^i$. To protect privacy and prevent sensitive data leakage, the client then applies the OPM mechanism to introduce noise to the compressed model, resulting in the final uploaded model $\tilde{\mathbf{w}}_i^t$.

3) Uploading local models: Clients upload their trained local models to the corresponding edge node $j \in \mathcal{M}$. The edge node performs partial aggregation to obtain an intermediate parameter matrix $\tilde{\mathbf{w}}_j^t = \sum_{i \in \mathcal{N}_j^t} \tilde{\mathbf{w}}_i^t$, where $\mathcal{N}_j^t \subseteq \mathcal{N}$ represents the set of clients whose disturbed model parameters are uploaded to edge node j in round t.

4) Incentive mechanism: The server collects the compressed models from all edge nodes and then pays clients according to their contributions (model accuracy). For fairness, the reward R_t^i for each worker is set to be proportional to its privacy budget ϵ_i and the ratio of retained parameters $(1 - \gamma_i^t)$. The total reward the server must pay to clients is $R^t = \sum_{i \in \mathcal{N}} R_t^i$.

5) Model aggregation: The server updates the global model by aggregating all the edge node model parameters $\tilde{\mathbf{w}}_j^t$.

The objective of this paper is to design a privacy-preserving personalized compression mechanism that comprehensively addresses the trade-offs between communication overhead, privacy protection, and global model performance. Specifically, we first aim to identify the optimal balance between communication cost and model performance. Subsequently, for larger privacy budgets, we propose an optimized privacy protection mechanism to further enhance the model's effectiveness.

IV. Game-Theoretic Personalized Compression Mechanism

This section introduces PCM, modeling the interaction between the server and clients as a two-stage Stackelberg game. The server, as the leader, aims to incentivize clients to upload more model parameters at minimal cost to accelerate global model convergence. Clients, as followers, balance monetary rewards against communication overhead when choosing their compression rates.

Clients must trade off between reducing communication costs and preserving model update quality, while the server must balance incentive costs with the quality of received updates.

We first define utility functions for both the server and clients, and then derive their optimal strategies based on the Stackelberg equilibrium, enabling joint optimization of communication efficiency and model performance.

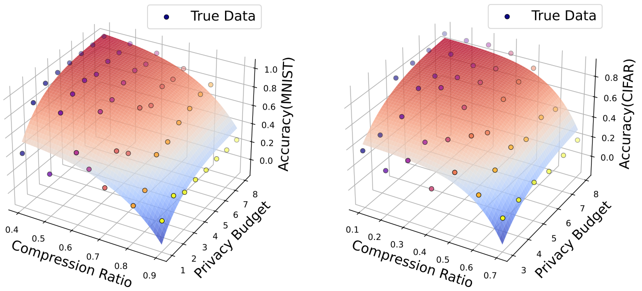
A. Utility Function Design

The server's objective is to minimize the total payment to all clients while maximizing the convergence rate of the global model. Thus, its utility function is influenced by both the degree of global model optimization and the total payment amount. In contrast, the client's objective is to maximize its individual utility by balancing reduced communication costs and higher rewards from the server. Consequently, the client's utility depends on both the received reward and the incurred communication cost. The server optimizes its utility by adjusting the payment amount R^t, while each client optimizes its utility by adjusting its compression rate γ_i^t.

In the following sections, we provide a detailed formulation of the utility functions for both the server and clients. For ease of reference, Table I summarizes the key notations used throughout this paper.

Server's Utility Function: The server's utility function increases with the degree of global model optimization and decreases with the increase in payment amounts. Therefore, the server's utility function is positively correlated with the accuracy of local models and is then reduced by the client rewards and the platform's operational costs.

Firstly, we define model performance as a function of the compression ratio γ_i^t and the privacy budget ϵ_i. Generally, a lower compression ratio and a higher privacy budget lead



a MNIST

b CIFAR

Fig. 2: The Model Accuracy varies with Privacy Budget and Compression Ratio.

to improved model performance, such as higher classification accuracy or prediction accuracy. We conduct experiments using machine learning models on different datasets to obtain model accuracy under various compression ratios and privacy budgets. The collected data is then used for nonlinear function fitting to derive the functional relationship between model accuracy, compression ratio, and privacy budget. As illustrated in Fig. 2, we present the functional graphs depicting how the test accuracy of a CNN model on the MNIST dataset and a ResNet12 model on the CIFAR-10 dataset vary with the compression ratio and privacy budget. Our observations indicate that the local model accuracy of client i, denoted as Ω_i^t, can be approximated as a convex function of the compression ratio and privacy budget, specifically, $\Omega_i^t(\gamma_i^t, \epsilon_i) = -\alpha e^{\lambda \gamma_i^t} - \mu e^{-\eta \epsilon_i} + \beta$, where $\alpha, \lambda, \mu, \eta$, and β are dataset-dependent parameters, all of which are positive.

Fig. 2a presents the test results on the MNIST dataset, where the fitted function is given by $\Omega_i^t(\gamma_i^t, \epsilon_i) = -0.0038 e^{5.8 \gamma_i^t} - 0.85 e^{-0.57 \epsilon_i} + 0.99$. Fig. 2b presents the test results on the CIFAR-10 dataset, where the fitted function is given by $\Omega_i^t(\gamma_i^t, \epsilon_i) = -0.0024 e^{7.8 \gamma_i^t} - 2.4 e^{-0.56 \epsilon_i} + 0.88$.

Next, for generality, we use a continuous invertible convex function $\Theta\left(\sum_i \Omega_i^t\right)$ to represent the evaluator's function, which assesses the overall model performance. Therefore, the server's utility function $\mathcal{U}^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^t\right)$ is defined as:

$$\mathcal{U}^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^{\mathbf{t}}\right) = \Theta\left(\sum_{i=1}^{N}\left(-\alpha e^{\lambda \gamma_i^t} - \mu e^{-\eta \epsilon_i} + \beta\right)\right) - R^t - R_p^t. \tag{1}$$

In the above equation, R^t represents the reward paid by the server in the t-th round, and R_p^t represents the rent paid to the platform in the t-th round.

Client's Utility Function: In the t-th round, the utility function $\mathcal{U}_i^t(R^t, \gamma_i^t, \gamma_{(-i)}^t)$ of each client i depends on the reward received from the server and the costs incurred in communication, privacy, and computation. It is defined as:

$$\mathcal{U}_i^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^{\mathbf{t}}\right) = R_i^t - c_i^{cm} - c_i^{pv} - c_i^{cp}$$

$$= \frac{\epsilon_i\left(1 - \gamma_i^t\right)}{\sum_{i=1}^{N} \epsilon_i\left(1 - \gamma_i^t\right)} R^t - \omega_1 \rho_i^{cm}\left(1 - \gamma_i^t\right) - \omega_2 \rho_i^{pv} \epsilon_i - \omega_3 \rho_i^{cp}\left|D_i\right| f_i^2. \tag{2}$$

In the above equation, ω_j (where $j \in \{1, 2, 3\}$) is a positive adjustment factor, and $\gamma_{(-i)}^t$ represents the vector $< \gamma_1^t, \gamma_2^t, \ldots, \gamma_N^t >$ excluding γ_i^t. The detailed explanation of each term is as follows:

- **Reward** R_i^t: The reward for client i comes from the total payment provided by the server, and it is proportional to ϵ_i and $(1 - \gamma_i^t)$, as higher privacy budgets and lower compression ratio enables the client to provide more effective local model information.
- **Communication Cost** c_i^{cm}: The communication cost for client i represents the communication resources consumed to upload the local model. It is a linear function of $(1 - \gamma_i^t)$. ρ_i^{cm} is a positive coefficient representing the unit communication cost. According to references [24], [25], our unit communication cost is defined as follows:

$$\rho_i^{cm} = \frac{|\Psi| \varrho_{ij}^t}{B_i^t \log_2\left(1 + \frac{g_{ij}^t \varrho_{ij}^t}{B_i^t M_0^t}\right)} + \frac{|\Psi| \varrho_{ji}^t}{B_i^t \log_2\left(1 + \frac{g_{ji}^t \varrho_{ji}^t}{B_i^t M_0^t}\right)}. \tag{3}$$

  Here, $|\Psi|$ represents the model size, and ϱ_{ij}^t, B_i^t, g_{ij}^t, and M_0^t are known parameters.
- **Privacy Cost** c_i^{pv}: The privacy cost represents the risk of privacy leakage and is proportional to the privacy budget. Here, ρ_i^{pv} is the unit privacy cost, where a higher unit privacy cost indicates that maintaining a higher level of privacy requires greater privacy costs.
- **Computational Cost** c_i^{cp}: According to reference [24], we define the computational cost of client i as $c_i^{cp} = \omega_3 \rho_i^{cp} |D_i| f_i^2$, representing the computational resource consumption for training the local model. Here, f_i is the computational capacity of client i, which depends on the clock frequency of the Graphics Processing Unit (GPU). ρ_i^{cp} is the unit computational cost.

The two-stage Stackelberg game we established is summarized as follows: By modeling the two-stage Stackelberg game, the optimal strategy set $\langle R^{t*}, \gamma^{t*} \rangle$ can be determined, where the server plays the role of the leader and the clients play the role of the followers. In the first stage, the server will choose the optimal payment R^{t*} to maximize its utility $\mathcal{U}^t$. Then, in the second stage, each client attempts to determine the optimal personalized compression rate γ^{t*} to maximize its own utility $\mathcal{U}_i^t$, given the payment R^t. For the two-stage Stackelberg game, our optimization objective is as follows:

$$\text{Server's side}: \text{Maximize } \mathcal{U}^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^{\mathbf{t}}\right), \tag{4}$$

$$\text{Client's side}: \text{Maximize } \mathcal{U}_i^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^{\mathbf{t}}\right), \tag{5}$$

$$\text{Subject to}: \sum_{i=1}^{N}\left(1 - \gamma_i^t\right) \leq \mathcal{C}, 0 \leq \gamma_i^t < 1, t = \{1, 2, \ldots\}. \tag{6}$$

Eq. (6) indicates that $\sum_{i=1}^{N}(1 - \gamma_i^t)$ has a strict upper bound. In other words, $\sum_{i=1}^{N} \gamma_i^t$ has a strict lower bound, which ensures that our communication cost does not exceed the maximum communication resource limit. As shown in Fig 2, as the compression rate decreases, the growth rate of model accuracy becomes very slow, meaning that retaining too many parameters does not contribute significantly to improving model accuracy.

B. The determination of the optimal strategy set

To obtain the optimal strategy set $\langle R^{t*}, \gamma^{t*}\rangle$, it is first necessary to determine the clients' optimal compression rate γ^{t*} based on the payment R^t provided by the server, as presented in Theorem 1.

Theorem 1: Given an arbitrary payment R^t, the optimal personalized compression rate for each client, γ_i^{t*}, is:

$$\gamma_i^{t*} = \frac{(N-1)R^t\left[(N-1)\rho_i^{cm} - \epsilon_i \sum_{i=1}^N \left(\frac{\rho_i^{cm}}{\epsilon_i}\right)\right]}{\omega_1 \epsilon_i^2 \left(\sum_{i=1}^N \left(\frac{\rho_i^{cm}}{\epsilon_i}\right)\right)} + 1. \tag{7}$$

Proof: First, we can derive the first-order and second-order partial derivatives of each client's utility function $\mathcal{U}_i^t(R^t, \gamma_i^t, \gamma_{-i}^t)$ with respect to γ_i^t. The results are shown as follows:

$$\frac{\partial \mathcal{U}_i^t}{\partial \gamma_i^t} = \omega_1 \rho_i^{cm} - \frac{\epsilon_i R^t \sum_{k \in N \setminus i} \epsilon_k \left(1-\gamma_k^t\right)}{\left[\sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right)\right]^2}, \tag{8}$$

$$\frac{\partial^2 \mathcal{U}_i^t}{\partial \left(\gamma_i^t\right)^2} = -\frac{2\epsilon_i^2 R^t \sum_{k \in N \setminus i} \epsilon_k \left(1-\gamma_k^t\right)}{\left[\sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right)\right]^3} < 0. \tag{9}$$

According to Eq. (9), the utility function of the client $\mathcal{U}_i^t(R^t, \gamma_i^t, \gamma_{-i}^t)$ is strictly convex within the domain of γ_i^t. Therefore, when $\partial \mathcal{U}_i^t / \partial \gamma_i^t = 0$, the utility function $\mathcal{U}_i^t(R^t, \gamma_i^t, \gamma_{-i}^t)$ reaches its maximum, leading to the desired optimal compression rate γ_i^{t*}. From the equation $\partial \mathcal{U}_i^t / \partial \gamma_i^t = 0$, we obtain:

$$\omega_1 \left(\frac{\rho_i^{cm}}{\epsilon_i}\right) \left[\sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right)\right]^2 = R^t \sum_{k \in N \setminus i} \epsilon_k \left(1-\gamma_k^t\right). \tag{10}$$

Simultaneously summing both sides of Eq. (10), we obtain:

$$\omega_1 \sum_{i=1}^N \left(\frac{\rho_i^{cm}}{\epsilon_i}\right) \left[\sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right)\right]^2 = (N-1) R^t \sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right). \tag{11}$$

According to Eq. (11), we can solve for the expression of $\sum_{i=1}^N \epsilon_i (1-\gamma_i^t)$ as follows:

$$\sum_{i=1}^N \epsilon_i \left(1-\gamma_i^t\right) = \frac{(N-1)R^t}{\omega_1 \sum_{i=1}^N \left(\rho_i^{cm}/\epsilon_i\right)}. \tag{12}$$

By substituting Eq. (12) into Eq. (10), we can obtain the optimal compression rate for the client γ_i^{t*} .

Based on Theorem 1, we find that the optimal compression rate γ_i^{t*} is related to the payment R^t, the privacy budget ϵ_i, and some known public parameters (i.e., $\omega_1, \ldots, N$, ρ_i^{cm}, and $\sum_{i=1}^N \frac{\rho_i^{\text{cm}}}{\epsilon_i}$). That is, as long as the leader (i.e., the server) confirms the payment, the follower (i.e., the client) can easily determine their optimal strategy. Therefore, the following Theorem 2 will provide the method for determining R^{t*}.

Specifically, we assume that the global communication resource is upper bounded by $\mathcal{C}$. We introduce a parameter ϕ to transform the communication constraint into the form $\sum_{i=1}^N \epsilon_i (1-\gamma_i^t) \leq \phi\mathcal{C}$. Based on this transformation, the Lagrangian function in the t-th round can be written as:

$$\mathfrak{L}\left(R^t, \gamma_i^t, \boldsymbol{\gamma}_{-i}^t, \varpi\right) \triangleq \mathcal{U}_i^t\left(R^t, \gamma_i^t, \boldsymbol{\gamma}_{-\mathbf{i}}^t\right) + \varpi \left(\phi\mathcal{C} - \sum_{i=1}^N \epsilon_i (1-\gamma_i^t)\right), \tag{13}$$

where ϖ is the Lagrange multiplier associated with the communication resource constraint. Since the two-stage Stackelberg game essentially involves a convex optimization problem, its optimal solution must satisfy the Karush-Kuhn-Tucker conditions:

$$\left(\frac{\partial \mathfrak{L}}{\partial \gamma_i^t}\right)\bigg|_{\gamma_i^t = \gamma_i^{t*}} = 0, \tag{14}$$

$$\varpi \geq 0, \quad \varpi \left(\phi\mathcal{C} - \sum_{i=1}^N \epsilon_i (1-\gamma_i^{t*})\right) = 0, \tag{15}$$

$$\sum_{i=1}^N \epsilon_i (1-\gamma_i^{t*}) \leq \phi\mathcal{C}. \tag{16}$$

Therefore, the optimal personalized compression rate for each client i, denoted by γ_i^{t*}, can be re-derived as:

$$\gamma_i^{t*} = -\frac{\phi\mathcal{C}}{\epsilon_i N} + \frac{\phi^2\mathcal{C}^2\omega_1}{\epsilon_i R^t}\left(\frac{\rho_i^{cm}}{\epsilon_i} - \frac{\mathcal{K}}{N}\right) + 1, \tag{17}$$

where $\mathcal{K} = \sum_{i=1}^N \frac{\rho_i^{cm}}{\epsilon_i}$.

Theorem 2: Given the optimal compression rate γ_i^{t*} determined in Theorem 1, the server's optimal payment R^{t*} satisfies the following equation:

$$\frac{\partial \Theta}{\partial \Omega_{sum}} \sum_{i=1}^N \left(\alpha\lambda e^{\lambda\gamma_i^{t*}} \cdot \frac{\partial \gamma_i^{t*}}{\partial R^t}\right) + 1 = 0. \tag{18}$$

Proof: Before proceeding with the detailed proof, we provide the definition of an equilibrium solution: In a game, if the strategies of all participants are determined, and each participant, knowing the strategies of the others, has no incentive to unilaterally change their own strategy to achieve a better outcome, then this strategy combination is called a Nash equilibrium [27]. The set of optimal personalized compression rate vectors $\gamma^{t*} = \{\gamma_1^{t*}, \gamma_2^{t*}, \ldots, \gamma_N^{t*}\}$ constitutes a Nash equilibrium. Thus, for any client i, we have:

$$\mathcal{U}_i^t\left(R^t, \gamma_i^{t*}, (\gamma_{-i}^t)^*\right) \geq \mathcal{U}_i^t\left(R^t, \gamma_i^t, (\gamma_{-\mathbf{i}}^t)^*\right). \tag{19}$$

The second stage can be considered a non-cooperative game because each client attempts to maximize its utility in a rational and self-interested manner. According to Eq. (19) and the Nash equilibrium theorem, the optimal personalized compression rate vector set $\gamma^{t*} = \{\gamma_1^{t*}, \gamma_2^{t*}, \ldots, \gamma_N^{t*}\}$ forms a Nash equilibrium in the non-cooperative game among the clients [28].

Next, we need to prove that the optimal payment obtained in the first stage also satisfies the Nash equilibrium condition. In this way, the entire game constitutes a Stackelberg equilibrium.

Substituting Eq. (7) into Eq. (1), we obtain $\mathcal{U}^t(R^t, \gamma_i^{t*}, \gamma_{-i}^{t*})$. Next, we derive the second-order partial derivatives of the server's utility function $\mathcal{U}^t(R^t, \gamma_i^{t*}, \gamma_{-i}^{t*})$ with respect to R^t, as shown below:

$$\frac{\partial^2 \mathcal{U}^t}{\partial \left(R^t\right)^2} = \frac{\partial^2 \Theta}{\partial \left(\Omega_{sum}\right)^2} \cdot \left(\sum_{i=1}^{N} \left(-\alpha \lambda e^{\lambda \gamma_i^{t*}} \frac{\partial \gamma_i^{t*}}{\partial R^t} \right) \right)^2 - \frac{\partial \Theta}{\partial \Omega_{sum}} \cdot \left(\left(\sum_{i=1}^{N} \alpha \lambda^2 e^{\lambda \gamma_i^{t*}} \left(\frac{\partial \gamma_i^{t*}}{\partial R^t} \right)^2 \right) + \left(\sum_{i=1}^{N} \alpha \lambda e^{\lambda \gamma_i^{t*}} \frac{\partial^2 \gamma_i^{t*}}{\partial \left(R^t\right)^2} \right) \right). \tag{20}$$

Since the function Θ we set is an increasing convex function (e.g., the widely used logarithmic function [29]), it always holds that $\partial\Theta/\partial\delta_{\text{sum}} > 0$ and $\partial^2\Theta/\partial(\delta_{\text{sum}})^2 \leq 0$. Meanwhile, according to Eq. (7), we have $\partial^2\gamma_i^{t*}/\partial(R^t)^2 \leq 0$.

Thus, it follows that $\partial^2\mathcal{U}^t/\partial(R^t)^2 < 0$, which indicates that $\mathcal{U}^t$ is a strictly convex function within the domain of R^t. When $\partial\mathcal{U}^t/\partial R^t = 0$, we obtain the unique optimal payment solution R^{t*}, which further leads to Eq. (18).

Since the exact solution for R^{t*} is difficult to obtain, but γ_i^{t*} is a linear function of R^t, we can approximate the solution to the equation using numerical methods. Specifically, common numerical methods, such as binary search or Newton's method, can be effectively used to find an approximation for R^{t*}.

Based on the above game-theoretic analysis, both the workers and the requester can find optimal strategies to maximize their utilities. Therefore, in such a complete-information game, there exists a unique Stackelberg equilibrium. In summary, the server can choose the optimal payment R^{t*} based on Theorem 2, while each client i can determine their optimal personalized compression rate γ_i^{t*} by submitting R^{t*}.

V. Optimized Piecewise Noise Mechanism Design

In this section, we design an efficient local differential privacy algorithm for clients to protect their privacy. We propose OPM to implement random noise addition. In this process, we not only consider the impact of model compression but also observe that when the privacy budget is large, the worst-case noise variance of OPM is significantly smaller than that of the traditional Piecewise Noise Mechanism (PM) [26]. Next, we will introduce the detailed content of this mechanism.

A. Optimized Denoising Mechanism

Given the compressed model $\mathbf{w}_t^i$ as input, after passing through the OPM, we obtain the noisy model $\tilde{\mathbf{w}}_i^t$. We assume that all the parameters of $\mathbf{w}_t^i$ have a value range of $[C - R, C + R]$, where C and R represent the center and radius of the range of $\mathbf{w}_t^i$, respectively. For a specific parameter $\mathbf{w}$ of $\mathbf{w}_t^i$, the true value is $x = \mathbf{w} - C$, and the perturbed value $\tilde{x}$ satisfies the following probability density function (for simplicity, we assume $\epsilon = \epsilon_i^t$):

$$F[\tilde{x} = y \mid x] = \begin{cases} \frac{e^{\frac{4\epsilon}{3}}(e^\epsilon - 1)}{2R\left(e^{\frac{\epsilon}{3}} + e^\epsilon\right)^2}, & y \in [f(\epsilon, x), g(\epsilon, x)] \\ \frac{e^{\frac{\epsilon}{3}}(e^\epsilon - 1)}{2R\left(e^{\frac{\epsilon}{3}} + e^\epsilon\right)^2}, & \\ \quad y \in [-RL, f(\epsilon, x)) \cup (g(\epsilon, x), RL] & \end{cases} \tag{21}$$

where

$$L = \frac{\left(e^\epsilon + e^{\epsilon/3}\right)\left(e^{\epsilon/3} + 1\right)}{e^{\epsilon/3}\left(e^\epsilon - 1\right)},$$
$$f(\epsilon, x) = \frac{\left(e^\epsilon + e^{\epsilon/3}\right)\left(xe^{\epsilon/3} - R\right)}{e^{\epsilon/3}\left(e^\epsilon - 1\right)}, \text{ and}$$
$$g(\epsilon, x) = \frac{\left(e^\epsilon + e^{\epsilon/3}\right)\left(xe^{\epsilon/3} + R\right)}{e^{\epsilon/3}\left(e^\epsilon - 1\right)}.$$

Let L be the carefully designed noise range adjustment parameter, used to limit the range of noise values. Therefore, based on the designed probability density function, the perturbed parameter $\tilde{\mathbf{w}}$ is given by $\tilde{\mathbf{w}} = C + \tilde{x}$. The perturbed model $\tilde{\mathbf{w}}$ is determined by the dynamic privacy budget ϵ_i^t, the disturbance range $[f(\epsilon, x), g(\epsilon, x)]$, and the intervals $[-RL, f(\epsilon, x)) \cup (g(\epsilon, x), RL]$.

B. Theoretical Analysis

Theorem 3: The OPM privacy protection mechanism satisfies ϵ-DP for each client $i \in N$.

Proof: For any pair of inputs x and x', we obtain:

$$\frac{F(x = \tilde{x} \mid x)}{F\left(x' = \tilde{x} \mid x'\right)} \leqslant \frac{\max_x F(x = \tilde{x} \mid x)}{\min_{x'} F\left(x' = \tilde{x} \mid x'\right)}$$
$$= \left[\frac{e^{\frac{4\epsilon}{3}}\left(e^\epsilon - 1\right)}{2R\left(e^{\frac{\epsilon}{3}} + e^\epsilon\right)^2}\right] \Big/ \left[\frac{e^{\frac{\epsilon}{3}}\left(e^\epsilon - 1\right)}{2R\left(e^{\frac{\epsilon}{3}} + e^\epsilon\right)^2}\right] = e^\epsilon. \tag{22}$$

According to the definition of differential privacy in [33], our OPM mechanism clearly satisfies ϵ-DP.

Theorem 4: The OPM perturbation mechanism does not introduce any bias, i.e., $E(\tilde{\mathbf{w}}) = \mathbf{w}$.

Proof: For any perturbation parameter $\tilde{\mathbf{w}}$, the expectation is as follows:

$$\begin{aligned} E(\tilde{\mathbf{w}}) &= E(C + \tilde{x}) = E(\tilde{x}) + C \\ &= \frac{xe^\epsilon}{e^\epsilon - 1} - \frac{x}{e^\epsilon - 1} + C = \mathbf{w}. \end{aligned} \tag{23}$$

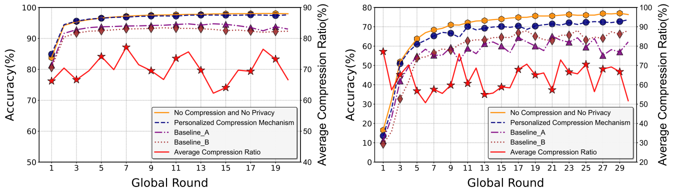
This shows that the OPM perturbation parameter does not introduce any bias. Thus, the proof of Theorem 4 is complete.

Theorem 5: Given any parameter $\mathbf{w}$, the variance of the perturbed model parameter $\tilde{\mathbf{w}}$ has explicit upper and lower bounds.

Proof: For a given parameter $\mathbf{w}$, the variance of the perturbed model parameter $\tilde{\mathbf{w}}$ is derived as follows:

$$\begin{aligned} \operatorname{Var}(\tilde{w}) &= \operatorname{Var}(\tilde{x} + c) = \operatorname{Var}(\tilde{x}) \\ &= E\left(\tilde{x}^2\right) - [E(\tilde{x})]^2 = E\left(\tilde{x}^2\right) - x^2 \\ &= \frac{e^{\epsilon/3} + 1}{e^\epsilon - 1}x^2 + \frac{\left(e^\epsilon + e^{\epsilon/3}\right)\left[\left(e^{\epsilon/3} + 1\right)^3 + e^\epsilon - 1\right]}{3e^{2\epsilon/3}\left(e^\epsilon - 1\right)^2}R^2. \end{aligned} \tag{24}$$

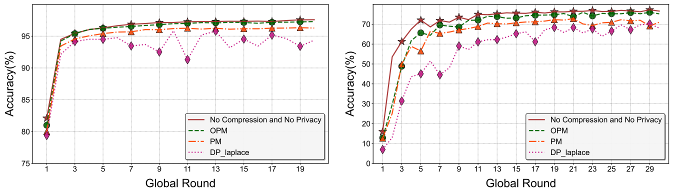
Since $x \in [-R, R]$, the variance $\operatorname{Var}(\tilde{\mathbf{w}})$ has explicit upper and lower bounds. Specifically, when $x = 0$, $\operatorname{Var}(\tilde{\mathbf{w}})$ attains its minimum value, whereas when $x = -R$ or $x = R$, $\operatorname{Var}(\tilde{\mathbf{w}})$



a MNIST

b CIFAR

Fig. 3: Accuracy comparison under different datasets.



a MNIST

b CIFAR

Fig. 4: Test performance of different differential privacy algorithms.

attains its maximum value. Thus, we obtain the upper and lower bounds for $\mathrm{Var}(\tilde{\mathbf{w}})$:

$$\mathrm{Var}(\tilde{\mathbf{w}}) \geq \frac{\left(e^{\epsilon}+e^{\frac{\epsilon}{3}}\right)\left[\left(e^{\frac{\epsilon}{3}}+1\right)^{3}+e^{\epsilon}-1\right]}{3e^{\frac{2\epsilon}{3}}\left(e^{\epsilon}-1\right)^{2}}R^{2}$$

$$\mathrm{Var}(\tilde{\mathbf{w}}) \leq \frac{e^{\frac{\epsilon}{3}}+1}{e^{\epsilon}-1}R^{2}+\frac{\left(e^{\epsilon}+e^{\frac{\epsilon}{3}}\right)\left[\left(e^{\frac{\epsilon}{3}}+1\right)^{3}+e^{\epsilon}-1\right]}{3e^{\frac{2\epsilon}{3}}\left(e^{\epsilon}-1\right)^{2}}R^{2}. \tag{25}$$

VI. Experimental Results and Analysis

To evaluate the performance of FedCP, we conducted simulations on real datasets. The experiment uses the MNIST and CIFAR-10 datasets, which are widely used in federated learning. The MNIST dataset contains 70,000 images, with 60,000 used for training and 10,000 for testing. Each image is a 28×28 grayscale handwritten digit. The CIFAR-10 dataset consists of 10 classes of 32×32 RGB images, containing 50,000 training samples and 10,000 test samples.

In the simulations, the total number of clients N is chosen from the range $[40, 120]$. The initial learning rate is set to 0.001, the local batch size is set to 16, and the number of local training rounds per client is set to 5. To ensure the reliability of the experiments, we randomize the unit privacy cost ρ_i^{pv} and unit computation cost ρ_i^{cp}, both of which are selected from the range $[5, 15]$, as well as the privacy budget ϵ of each client, which is chosen from the range $[1, 10]$. Finally, we set $M = N/2$, where clients are randomly assigned to edge nodes, and each edge node is guaranteed to have at least one client. We also set $\omega_j = 1$ $(j \in 1, 2, 3)$, and $R_p^t = 600$. Additionally, we use a linear function Θ to simplify the server's utility function, i.e., $\mathcal{U}^t\left(R^t, \gamma_i^t, \gamma_{-\mathbf{i}}^{\mathbf{t}}\right) = k\left(\sum_{i=1}^{N}\left(-\alpha e^{\lambda\gamma_i^t} - \mu e^{-\eta\epsilon_i} + \beta\right)\right) - R^t - R_p^t$, where k is a transformation parameter set to 100 in our experiments.

A. Simulation experiments of PCM

Given that existing research has not yet explored the dynamic adjustment of client model compression rates during the training process, this paper designs two rigorous and impartial controlled experiments. To ensure the scientific validity and verifiability of the experimental comparisons, we first conduct a systematic statistical analysis of the client model compression rates across different training rounds in the federated learning model incorporating the PCM mechanism. Based on this analysis, we construct the following two types of controlled experiments:

First controlled experiment (Baseline_A): This experiment adopts a round-based dynamic average compression rate strategy. Specifically, in each training round of the federated learning model with the PCM mechanism, we compute the arithmetic mean of the model compression rates for all clients in the current round and use this mean value as the uniform compression rate for all clients in this controlled experiment for that round.

Second controlled experiment (Baseline_B): This experiment employs a global static compression rate strategy. Specifically, across all training rounds of the federated learning model with the PCM mechanism, we first compute the global average of client model compression rates. This fixed compression rate value is then consistently applied to all clients in this controlled experiment throughout the entire training process.

This paper first conducted systematic experimental validation on the MNIST dataset, setting the total number of global rounds to 20 and the number of clients to 40. The experiment was performed in a distributed learning framework, where each client was equipped with a lightweight neural network consisting of two convolutional layers and one fully connected layer. The experimental results are shown in Fig. 3a. This figure illustrates the variation in the average compression ratio of clients across training rounds in federated learning. This metric directly reflects the efficiency of communication resource utilization, where a higher compression ratio corresponds to lower communication resource consumption. The experimental results indicate that the global average compression ratio is 70 %, implying that our method effectively reduces communication costs by approximately 70 %. Although our method achieves slightly lower classification accuracy on the test set compared to the baseline method (with an average difference of 0.21 %), it demonstrates significantly superior model performance and stability compared to the Baseline_A and Baseline_B approaches. This ensures high communication efficiency while substantially enhancing model performance.

We also conducted systematic experimental validation on the CIFAR-10 dataset. Due to the increased dataset size and learning complexity, the total number of global rounds was set to 30. The experiment was performed in a distributed learning framework, where each client was configured with a ResNet-12 deep learning neural network. The experimental results are shown in Fig. 3b. The results indicate that the global average compression ratio is 63 %, meaning that our method

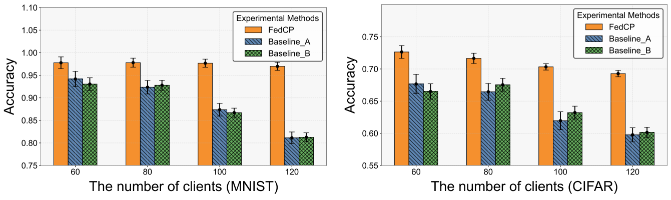


Fig. 5: Accuracy versus. The number of total clients under IID.

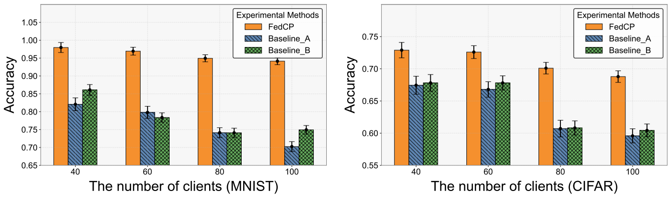
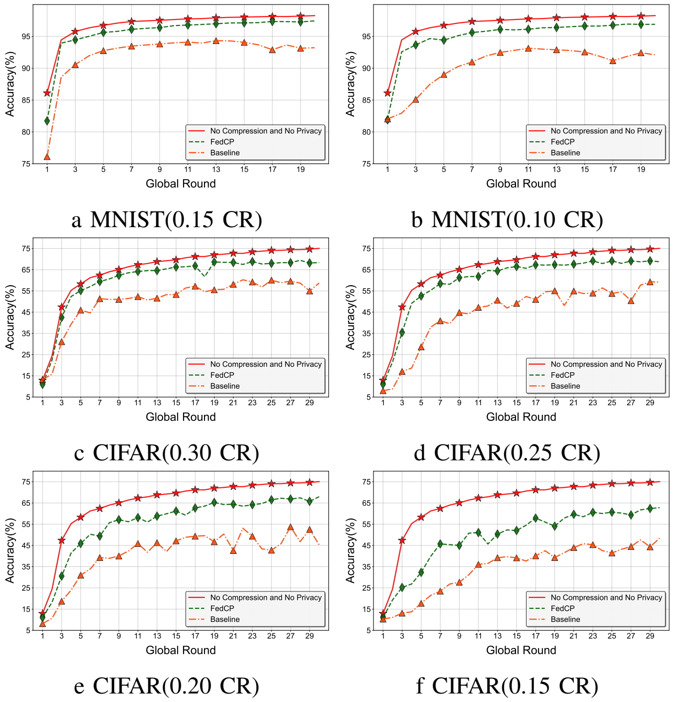


Fig. 6: Accuracy versus. The number of total clients under Non-IID.



a MNIST(0.15 CR)

b MNIST(0.10 CR)

c CIFAR(0.30 CR)

d CIFAR(0.25 CR)

e CIFAR(0.20 CR)

f CIFAR(0.15 CR)

Fig. 7: Model Accuracy versus. Rounds under communication-constrained scenarios.

effectively reduces communication costs by approximately 63 %. Furthermore, compared to other methods, our approach achieves significantly higher model accuracy and stability.

B. Simulation experiments of FedCP

We use the global model's test accuracy to evaluate the performance of the OPM. First, for the model compression rate of each client, we uniformly apply our proposed PCM. The comparative experiments use different privacy algorithms, which are: (1) the baseline method **No Compression and No Privacy**; (2) the PM algorithm; (3) the Laplace algorithm. We then test on the MNIST and CIFAR datasets. Based on the task difficulty, the total global training rounds are set to 20 and 30, respectively, with the number of clients set to 40. Our experimental results are shown in Fig. 4. Clearly, the accuracy and convergence speed of our proposed OPM are significantly better than the other algorithms. This is because OPM adds smaller noise variance, and the perturbed values are more likely to be close to the original values. Moreover, we observe that the performance gap on the CIFAR dataset is more pronounced, as the model used is more complex, leading to more significant improvements in performance.

Next, we gradually increase the total number of clients to 60, 80, 100, and 120 to investigate the impact on model accuracy. The experimental results are shown in Fig. 5. In the simulation process, we evenly distribute the total training samples among the clients, forming their respective local datasets. As a result, with the increase in the number of clients, the size of each local dataset decreases accordingly. The experimental results show that as the number of clients increases, the performance gap between the baseline method and the proposed method becomes more significant. We also conducted experiments on a Non-Independent and Identically Distributed (Non-IID) dataset to further validate the proposed method's robustness. As shown in Fig. 6, the performance gap between the baseline method and our approach is even more pronounced in this scenario. The primary reason for this phenomenon is that the Non-IID datasets further amplify the heterogeneity among clients. Our proposed method effectively adapts to this heterogeneity by dynamically adjusting the personalized compression rate, thereby enhancing the overall model performance.

C. Scenarios with Limited Communication Resources

We conducted a simulation to explore the impact of communication resource limitations on model performance. In the experiment, the number of clients was set to 100, and it was assumed that the global communication resources required without model compression were referred to as Communication Resources (CR). We tested two scenarios: one on the MNIST dataset with a communication resource limit of 0.15 CR, and the other on the CIFAR dataset with a communication resource limit of 0.30 CR. In the baseline experiment (Baseline), all clients used a uniform compression rate. The experimental results are shown in Fig. 7a and 7c. The accuracy of the FedCP model converges to the optimal result at a significantly faster rate. Compared to the case without communication resource limitations, FedCP shows a particularly significant improvement in global model accuracy. This phenomenon occurs because, when communication resources are limited, FedCP ensures that clients with higher contributions receive more communication resources, rather than dividing the limited resources equally among all clients, as in the baseline method.

Furthermore, we progressively increased the communication resource limits, and the experimental results are shown in Fig. 7b, 7d, 7e, and 7f. The results demonstrate that as the communication resource limits increase, the performance gap between the baseline method and FedCP becomes more pronounced. This phenomenon can be attributed to the fact that when communication resources are scarce, it is particularly important to allocate them efficiently to each client. FedCP achieves this by dynamically adjusting the compression rate of each client, thus ensuring the efficient use of communication resources.

VII. Conclusion

This paper proposes FedCP, a two-stage Stackelberg game-based federated learning framework that balances communication overhead, privacy protection, and model performance. In this framework, the server and clients act as the leader and followers, respectively, optimizing payment and compression strategies through game-theoretic decision-making. FedCP also integrates differential privacy with an optimized noise addition mechanism to enhance robustness and security under high privacy budgets. Experimental results demonstrate that FedCP significantly reduces communication costs and strengthens privacy protection while maintaining global model performance. Future work will explore its scalability and robustness in large-scale, asynchronous, and unreliable environments, with further optimization for real-world applications.

References

[1] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.

[2] Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning[J]. Foundations and trends® in machine learning, 2021, 14(1–2): 1-210.

[3] Alistarh D, Grubic D, Li J, et al. QSGD: Communication-efficient SGD via gradient quantization and encoding[J]. Advances in neural information processing systems, 2017, 30.

[4] Dettmers T. 8-bit approximations for parallelism in deep learning[J]. arXiv preprint arXiv:1511.04561, 2015.

[5] Hönig R, Zhao Y, Mullins R. DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning[C]//International Conference on Machine Learning. PMLR, 2022: 8852-8866.

[6] Bernstein J, Wang Y X, Azizzadenesheli K, et al. signSGD: Compressed optimisation for non-convex problems[C]//International Conference on Machine Learning. PMLR, 2018: 560-569.

[7] Wangni J, Wang J, Liu J, et al. Gradient sparsification for communication-efficient distributed optimization[J]. Advances in Neural Information Processing Systems, 2018, 31.

[8] Stich S U, Cordonnier J B, Jaggi M. Sparsified SGD with memory[J]. Advances in neural information processing systems, 2018, 31.

[9] Qian X, Richtárik P, Zhang T. Error compensated distributed SGD can be accelerated[J]. Advances in Neural Information Processing Systems, 2021, 34: 30401-30413.

[10] Aji A F, Heafield K. Sparse communication for distributed gradient descent[J]. arXiv preprint arXiv:1704.05021, 2017.

[11] Triastcyn A, Faltings B. Federated learning with bayesian differential privacy[C]//2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019: 2587-2596.

[12] 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.

[13] Truex S, Liu L, Chow K H, et al. LDP-Fed: Federated learning with local differential privacy[C]//Proceedings of the third ACM international workshop on edge systems, analytics and networking. 2020: 61-66.

[14] Sun L, Qian J, Chen X. LDP-FL: Practical private aggregation in federated learning with local differential privacy[J]. arXiv preprint arXiv:2007.15789, 2020.

[15] Asad M, Shaukat S, Hu D, et al. Limitations and future aspects of communication costs in federated learning: A survey[J]. Sensors, 2023, 23(17): 7358.

[16] Hsieh K, Harlap A, Vijaykumar N, et al. Gaia:Geo-Distributed machine learning approaching LAN speeds[C]//14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). 2017: 629-647.

[17] Sattler F, Wiedemann S, Müller K R, et al. Robust and communication-efficient federated learning from non-iid data[J]. IEEE transactions on neural networks and learning systems, 2019, 31(9): 3400-3413.

[18] Zhang D, Li S, Chen C, et al. Laptop-diff: Layer pruning and normalized distillation for compressing diffusion models[J]. arXiv preprint arXiv:2404.11098, 2024.

[19] Fang G, Ma X, Song M, et al. Depgraph: Towards any structural pruning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 16091-16101.

[20] Ma X, Fang G, Wang X. Llm-pruner: On the structural pruning of large language models[J]. Advances in neural information processing systems, 2023, 36: 21702-21720.

[21] Chen J, Yan H, Liu Z, et al. When federated learning meets privacy-preserving computation[J]. ACM Computing Surveys, 2024.

[22] Zhang C, Li S, Xia J, et al. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning[C]//2020 USENIX annual technical conference (USENIX ATC 20). 2020: 493-506.

[23] Byrd D, Polychroniadou A. Differentially private secure multi-party computation for federated learning in financial applications[C]//Proceedings of the First ACM International Conference on AI in Finance. 2020: 1-9.

[24] Ng J S, Lim W Y B, Dai H N, et al. Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled internet of vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(4): 2326-2344.

[25] Zhan Y, Zhang J. An incentive mechanism design for efficient edge learning by deep reinforcement learning approach[C]//IEEE INFOCOM 2020-IEEE conference on computer communications. IEEE, 2020: 2489-2498.

[26] Wang N, Xiao X, Yang Y, et al. Collecting and analyzing multi-dimensional data with local differential privacy[C]//2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, 2019: 638-649.

[27] Xu Y, Xiao M, Wu J, et al. Incentive mechanism for spatial crowdsourcing with unknown social-aware workers: A three-stage stackelberg game approach[J]. IEEE Transactions on Mobile Computing, 2022, 22(8): 4698-4713.

[28] Myerson R B. Game theory[M]. Harvard university press, 2013.

[29] Xu Q, Su Z, Lu R. Game theory and reinforcement learning based secure edge caching in mobile social networks[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3415-3429.

[30] Kang J, Xiong Z, Niyato D, et al. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory[J]. IEEE Internet of Things Journal, 2019, 6(6): 10700-10714.

[31] Zhang J, Wu Y, Pan R. Incentive mechanism for horizontal federated learning based on reputation and reverse auction[C]//Proceedings of the Web Conference 2021. 2021: 947-956.

[32] Xu Y, Xiao M, Wu J, et al. A personalized privacy preserving mechanism for crowdsourced federated learning[J]. IEEE Transactions on Mobile Computing, 2023, 23(2): 1568-1585.

[33] Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4): 211-407.