ACDP-pFSD: A Personalized Federated Self-Decoupled Distillation Framework with User-level Differential Privacy based Adaptive Clipping

Abstract. Federated learning faces two major challenges: data heterogeneity and privacy leakage risks. In response to these problems, this paper introduces a personalized federated distillation method with user-level differential privacy, named ACDP-pFSD. This method provides personalized models for each client while considering the performance of the global model and ensuring user-level differential privacy. Specifically, the method saves the personalized models locally from previous training. Through self-decoupled knowledge distillation, it learns personalized knowledge into the new local models to train models that match the client's data distribution. Moreover, to mitigate the negative impact of user-level differential privacy on model performance, adaptive model update clipping is used to balance privacy and utility. Finally, extensive simulation experiments validate the effectiveness of ACDP-pFSD. The algorithm not only shows improved accuracy in both personalized and global models but also achieves a good balance between model performance and privacy protection.

Keywords: Federated Learning, Knowledge Distillation, User-level Differential Privacy, Adaptive Clipping.

1 Introduction

With the continued development of mobile computing technology, intelligent devices generate massive amounts of data. Nevertheless, due to increasing privacy concerns and strict data protection regulations, centralized machine learning faces challenges in aggregating raw data for training models. Driven by these real-world challenges, Federated learning (FL) has emerged as a novel distributed machine learning paradigm to improve data availability and protect data privacy[1]. The server coordinates multiple clients to train the global model, in which the participating clients only upload the models to the server and always keep their data locally. FL has been widely used in several domains, such as finance, healthcare, transportation, and education [2,3].

However, there are still some challenges in FL research. In most application scenarios, client data is non-independent and identically distributed (non-IID), which leads to data heterogeneity. In this situation, the global model of the classical FL suffers from performance degradation and slow convergence, and it is difficult to satisfy all participating clients in the training process, which undermines the initial purpose of participating clients in FL. For instance, in cases where client data is non-IID, models trained locally may outperform the global model through FL[4]. Personalized federated learning (pFL) has been proposed to train a well-performing personalized model for each client to address the challenge of data heterogeneity. However, the pFL approaches often fail to consider the performance of the global model simultaneously. In addition,

it has been shown that the “curious” server may still infer private information about clients from the models they upload [5,6,7].

The data heterogeneity and the privacy leakage risk have been widely studied as two significant challenges in FL. To solve the data heterogeneity, existing research in pFL includes methods such as parameter decoupled [8], multi-task learning [9,10], regularization [11], and knowledge distillation (KD) [12]. However, most of the previous works only consider the performance of personalized models while ignoring the global model. Nevertheless, pFL requires personalized knowledge to train a personalized model that fits the local client distribution and public knowledge of the global model to enhance the generalization of both personalized models and the global model. Hence, it is necessary to improve the performance of the global model while focusing on the performance of personalized models in pFL. On the one hand, clients can benefit from the global model and get a local model with both personalization and generalization capabilities [12]; on the other hand, the performance of an excellent global model improves the performance of the initial model for new clients joining in each round, thereby impacting the motivation of clients to participate in FL [13]. Additionally, to mitigate the privacy leakage risk of FL, differential privacy (DP) mechanisms [14] have been widely employed, where user-level differential privacy (UDP) protects the privacy of the dataset of each client by Gaussian noise perturbing the model parameters or model updates [15,16,17,18,19,20,21,22]. However, compared to FL approaches that do not consider UDP, there is still a significant reduction in model accuracy. This occurs because the norms of the model or the model updates gradually decrease during the training process. Maintaining a fixed clipping threshold under these conditions introduces a substantial amount of noise. Therefore, it is imperative to adopt adaptive clipping of the model updates.

Addressing the aforementioned issues, this paper proposes a personalized federated distillation framework based on user-level differential privacy. This framework not only mitigates the impact of data heterogeneity by providing personalized and global models for each client but also offers UDP guarantees to balance privacy protection with model utility. The contributions of this paper are as follows:

(1) We propose a UDP-based pFL framework that learns richer knowledge through self-decoupled distillation to obtain both personalized models adapted to the client's data distribution and enhance the generalization ability of both personalized and global models. Finally, the model updates after updating is adaptively clipped and Gaussian noise is added to achieve privacy protection.

(2) An adaptive clipping algorithm based on Proximal Policy Optimization (PPO) is proposed. The method transforms the process of finding a suitable clipping threshold into a Markov decision process, which can automatically select the optimal clipping threshold for each round of the training process of federated learning, thus improving the utility of the model with UDP guarantees.

(3) Experimentally simulated in multiple datasets and multiple heterogeneous scenarios, and compared with the current state-of-the-art FL methods, the framework in this paper performs well in both personalized models and the global model, while trading off privacy preservation and model accuracy.

The remainder of this paper is organized as follows: Section 2 reviews related work on FL and UDP. Section 3 introduces fundamental concepts related to federated learning, as well as definitions and theorems about DP. Section 4 provides a detailed description of the ACDP-pFSD process and its privacy analysis. Section 5 analyzes and summarizes the experimental results. Finally, Section 6 concludes the paper.

2 Related work

2.1 Personalized Federated Learning

The study of pFL has drawn a lot of attention recently. It can be broadly categorized into research directions such as parameter decoupled [8,13], multi-task learning [9,10], regularization [11], and knowledge distillation [12]. Arivazhagan et al. [8] proposed FedPer, in which the model is divided into base layers and personalized layers, and samples are processed through base layers before undergoing personalized layers. Moreover, the client's personalized model consists of base layers and its personalized layers. Despite the outstanding personalized performance, the performance of the global model is poor. Based on this, Mei et al. [13] proposed the FedVF method, which improves the performance of both the global model and personalized models using a phased learning strategy that adjusts the update frequency of parameters on different levels. In multi-task learning, Sattler et al.[9] proposed the federated multi-task learning framework FMTL, which first computes the cosine similarity of the gradient update of the clients to infer the data distribution of the clients, and clients with similar data distributions are clustered together, each with its proprietary personalized model, but it is only suitable for the scenarios with high similarity. Li et al. [11] train a personalized model and the global model separately locally, and each client trains its personalized model and adjusts the difference between personalized models and the global model using a penalty factor, equivalent to local training only when the penalty factor is zero. Although these methods allow each client to train a personalized model that performs well on its specific task, most methods have poor performance of the global model.

2.2 Knowledge Distillation in Federated Learning

In 2015, Hinton et al. [23] first proposed KD. Zhao et al.[24] provided a new perspective on KD by decoupling the weights of target and non-target classes of the classical KD formulation. In FL, KD can transfer knowledge from the server to each client to improve the performance of personalized models [25,26] or transfer knowledge locally to the server to get a more robust global model [27,28]. Li et al. [25] proposed a knowledge distillation approach, FedMD, where the client gets logits using the public dataset each round and then sends these logits to the server for aggregation. Finally, the client learns global knowledge through knowledge distillation. Lin et al. [28] proposed an ensemble distillation method for model aggregation in FL, where the local model acts as the teacher model, and the global model is the student model. At each round,

soft labels for the local models are obtained on the server using small batches of unlabeled public datasets, and knowledge is transferred to the global model, resulting in an improved performance of the global model. In the studies, some methods require public datasets, while others rely on proxy data, which can be challenging to obtain in practice.

In recent years, variants of knowledge distillation, such as mutual distillation [29], cyclic distillation [30], and self-knowledge distillation [12] have emerged as research directions in pFL. To address multiple heterogeneous problems, Shen et al. [29] proposed FML that clients have both a personalized model and the global model locally and mutual learning between them through knowledge distillation. Moreover, during each communication round, only the global model is uploaded. Nevertheless, the performance of personalized models generated via mutual distillation may also suffer if the generalization of the global model is not sufficiently robust. Shen et al. [30] proposed a cyclic distillation method, named CD^2-pFed, which decouples each layer of the model into personalized and global parameters, and cyclically distills between the personalized and global parameters to obtain a better personalized model. Jin et al. [12] proposed pFedSD, which obtains personalized models through self-knowledge distillation and tradeoffs for personalization and generalization. However, classical KD is highly coupled, meaning that as the teacher model gains confidence in predicting the target class, the student model captures less "dark knowledge". This can prevent clients from obtaining personalized knowledge, as well as a lack of rich external knowledge for self-knowledge distillation in FL, affecting the model's generalizability. As the aforementioned researches show, it is vital to pay attention to the performance of both personalized models and the global model in pFL, because the global model's overall performance has an impact on the majority of knowledge distillation methods.

2.3 User-level differential privacy in FL

McMahan et al. [16] initially proposed the concept of user-level differential privacy in federated learning, aimed at protecting all privacy information of an entire client. They trained a model through federated learning to predict the next word, utilizing the Gaussian mechanism and moment accounting to ensure user-level differential privacy. Geyer et al. [17] utilized the median norm of all clients participating in a round as the clipping threshold for that round, and then the server clipped and noised the model updates (i.e., the differences between local and global models) to provide privacy protection. To balance privacy and performance, Andrew et al. [18] tracked quantiles on the server and adaptively learned the clipping thresholds to minimize the adverse impact of noise. However, as federated learning is a form of distributed machine learning, it is prudent to assume that the server is untrustworthy. Based on this premise, related work [15,19] added noise to models or model updates locally before uploading them to the server. Wei et al. [19] proposed the NbADL algorithm, which first clips the local model to be uploaded and then adds noise to the model before sending the perturbed model to the server. Subsequently, Zhang et al. [15] theoretically demonstrated that clipping model updates is superior to clipping models themselves, concluding that greater differences

in data distribution lead to greater declines in model performance due to clipping. Additionally, Bagdasaryan et al. [20] proved that user-level differential privacy can reduce the effectiveness of backdoor attacks in federated learning.

3 Preliminaries

In this section, background knowledge and relevant definitions of FL and DP will be presented.

3.1 Federated learning

Federated learning typically consists of a server and P clients that collaboratively train a global model. Let P denotes the set of all clients with $|D_i|$ samples for each client $i \in P$, and $D = \bigcup_{i \in P} D_i$ denotes the entire dataset. On model w and sample z, $\mathcal{F}_i(\theta, z)$ denotes the loss function of the i-th client and $\mathcal{F}_i(\theta, D_i) = \frac{1}{|D_i|} \sum_{i \in D_i} \mathcal{F}_i(\theta, z)$ denotes the loss function on the dataset D_i and model θ. The goal of classical FL is to find a global model θ that minimizes the total loss on the total dataset D_i:

$$\min_{\theta} \mathcal{F}(\theta) = \sum_{i \in P} \frac{|D_i|}{|D|} \mathcal{F}_i(\theta, D_i) \tag{1}$$

To solve this optimization task, FL performs multiple communication rounds between the local training phase $\theta_i = \text{avgmin}_{\theta} \mathcal{F}_i(\theta, D_i)$ and the server aggregation phase $\theta = \sum_{i \in P} \frac{|D_i|}{|D|} \theta_i$, where the following two phases are performed in each communication round:

Local training phase: in each communication round, the participating clients receive the global model and initialize it as their local model then the clients perform several local updates, and finally send the local model to the server.

Server aggregation phase: the server averages the local models from participating clients and aggregates a new global model, which is then sent to the participating clients.

3.2 Differential privacy

Differential privacy is a rigorous privacy concept for measuring privacy risk, and its related concepts are described next.

Definition 1 (ε, δ)-DP [14]

When the privacy budget $\varepsilon > 0$ and $0 \leq \delta < 1$, if for any two neighboring datasets D, D' and every possible subset of outputs O, a randomized mechanism $\mathcal{M}$ satisfies the following inequality:

$$Pr[\mathcal{M}(D) \in O] \leq e^{\varepsilon} Pr[\mathcal{M}(D') \in O] + \delta \tag{2}$$

then $\mathcal{M}$ satisfies (ε, δ)-DP.

In FL, DP can be categorized into sample-level differential privacy and user-level differential privacy, which are defined concerning the definition of the neighboring dataset D, D' [15].

Definition 2 User-level differential privacy [15]

Assume that $D = \{D_i\}_{i=1}^{N}$, if dataset D' can be obtained by adding or removing a dataset D_i to D, then D, D' is defined as the two neighboring datasets in the user-level differential privacy mechanism.

Definition 3 l_2 Sensitivity [14]

For any two neighboring datasets D, D', f is a query function on a dataset, l_2 sensitivity defined as:

$$\Delta_2 f = \max_{D,D'} \|f(D) - f(D')\|_2 \tag{3}$$

where $\|\cdot\|_2$ is the Euclidean norm.

Definition 4 (α, ρ)-RDP [31]

When a real number $\alpha > 1$ and privacy parameter $\rho \geq 0$, for any two neighboring datasets D, D', the $Rényi$ α-divergence between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ if satisfied:

$$D_\alpha[\mathcal{M}(D)||\mathcal{M}(D')] := \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{\mathcal{M}(D)}{\mathcal{M}(D')}\right)^{\alpha}\right] \leq \rho(\alpha) \tag{4}$$

then $\mathcal{M}$ satisfies (α, ρ)-RDP, which is a valid tool for measuring privacy and has tighter bounds.

Lemma 1 (α, ρ)-RDP translates into (ε, δ)-DP [31]. If a random mechanism $\mathcal{M}$ satisfies the $\big(\alpha, \rho(\alpha)\big)$-RDP, it satisfies $\left(\rho(\alpha) + \frac{log(1/\delta)}{\alpha - 1}, \delta\right)$-DP.

Lemma 2 The Gaussian mechanism is translatable to $(\alpha, \rho(\alpha))$-RDP [31]. Given f as a query function on a dataset with ℓ_2 sensitivity $\Delta_2 f$, the Gaussian mechanism $\mathcal{M} = f(D) + \mathcal{N}(0, \sigma^2 \Delta_2 f^2)$ satisfies $\left(\alpha, \frac{\alpha}{2\sigma^2}\right)$-RDP.

Lemma 3 The composition theorem of RDP [31]. For a dataset D and identical α, if multiple random mechanisms $\mathcal{M}_i$ satisfies (α, ρ_i)-RDP, their collective composition theorem substantiates $(\alpha, \sum_i \rho_i)$-RDP.

4 Methodology

In this section, we propose a user-level differential privacy-based personalized federated learning framework(see Fig. 1). And we introduce the personalized federated self-decoupled distillation method, followed by modeling the adaptive trimming threshold problem as a Markov decision process.

4.1 ACDP-pFSD Framework

In traditional federated learning, a singular global model fails to adapt to the heterogeneous distributions of different clients and also has privacy leakage risks. Moreover, the application of UDP in FL necessitates a balance between privacy protection and model utility. To address these issues, this paper introduces a personalized federated distillation algorithm based on user-level differential privacy. On one hand, it uses a

self-decoupled distillation method to learn personalized knowledge from each client to generate personalized models, making full use of local knowledge to obtain a global model with enhanced generalization capabilities. On the other hand, it reduces noise addition by adaptively adjusting the clipping threshold, thus maintaining the high utility of the model while achieving a balance between model utility and privacy protection.

Algorithm 1 describes the basic procedure of the personalized federated privacy protection algorithm ACDP-pFSD based on adaptive clipping, jointly executed by the server and multiple clients. During the federated learning training process, the server randomly initializes the global model and initializes the parameters required for the PPO. At the start of each communication round, the server adaptively selects the clipping threshold for that round according to Algorithm 2, performs local training, and then uploads the differentially private model update $\tilde{\Delta}_i^t$ to the server. Finally, the server aggregates the model updates to generate a new global model. Steps 3 to 6 are repeated until the model converges.

Algorithm 1： ACDP-pFSD

Input: Clients set P, Communication rounds T
Output: Personalized model $\{\theta_1, \theta_2, \ldots, \theta_P\}$, Global model θ^T

1. Initialize global model;
2. Initialize PPO;
3. **for** $t = 1,2, \ldots, T$ communication rounds **do**
4. $C^t, \tilde{\Delta}_i^t \leftarrow$Get clipping threshold and update clients (Algorithm 2);
5. $\theta^{t+1} \leftarrow \theta^t + \frac{1}{|P|}\sum_{i \in P^t} \tilde{\Delta}_i^t$;
6. **end for**
7. **return** θ^T

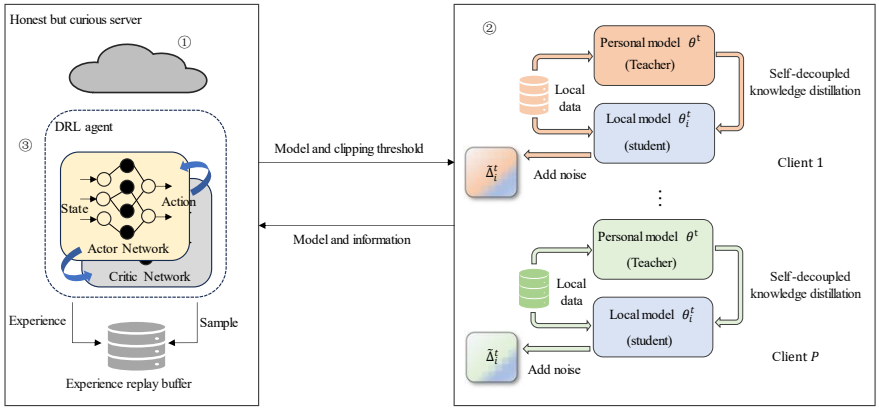


Fig. 1 ACDP-pFSD framework

4.2 Self-decoupled Knowledge Distillation in pFL

We propose a self-decoupled knowledge distillation method in pFL(see Fig. 2). Specifically, the previously personalized model θ_i is distilled through self-decoupled

knowledge distillation to obtain the local model θ_i^t of the current round. And the local model θ_i^t trained by this round is stored locally as the latest personalized model so that the local model can subsequently learn personalized knowledge. At this point, model θ_i is the teacher model, and the model θ_i^t is the student model. To address the problem of high coupling of classical KD formulas, we decouple the weights of target and non-target classes for self-knowledge distillation and transfer their logits separately. This not only aims to improve the performance of personalized models in learning "dark knowledge" from the teacher model but also contributes to the aggregation of the robust global model with better generalization.

Specifically, classical knowledge distillation can be expressed as:

$$\mathcal{KL}(\theta_{tm}||\theta_{sm}, x) = \mathcal{KL}\left(b^{\theta_{tm}}\|b^{\theta_{sm}}\right) + \left(1 - p_x^{\theta_{tm}}\right)\mathcal{KL}\left(\hat{p}^{\theta_{tm}}\|\hat{p}^{\theta_{sm}}\right) \quad (5)$$

where the class probability is denoted as $p = [p_1, p_2, \dots, p_x, \dots, p_C] \in \mathbb{R}^{1\times C}$, C is the number of classes, and p_i is the probability of the i-th class. The binary probability of all classes $b = \left[p_x, p_{\setminus x}\right] \in \mathbb{R}^{1\times 2}$, where p_x represents the probability of the target class and $p_{\setminus x}$ represents the probability of all other non-target classes. Then, excluding the target class (i.e., the x-th class), the probability of the non-target class is denoted as $\hat{p} = [p_1, \dots, p_{x-1}, p_{x+1}, \dots, p_C] \in \mathbb{R}^{1\times(C-1)}$ and $\hat{p}_i = \frac{\exp(z_i/KT)}{\sum_{j=1, j\neq t}^{C}\exp(z_j/KT)}$, z_i is the prediction of the i-th class, and KT is the temperature at distillation. Observing Equation(5), it can be inferred that there is a high coupling between the percentage of non-target classes and the teacher model's target class probability $p_x^{\theta_{tm}}$[24] . In a word, as the teacher model becomes more confident in the target class, i.e., the higher the target class probability, the less "dark knowledge" of the non-target class will be learned, which is likely to lead to the model performing confidently only on the target class, resulting in a degradation of the model performance.

Therefore, we redesign the formula for self-decoupled knowledge distillation in the FL, aiming for the student model to learn more knowledge from all classes. The goal is to enhance the generalization capability of personalized models, subsequently aggregating to obtain a robust global model, which in turn personalized models can also better benefit from the global model. The objective is to enable the student model to learn more knowledge from all classes, thereby augmenting the generalization capacity of personalized models. Then, the server aggregates local model updates to get a global model with strong generalization, after which the personalized model can also better benefit from the global model. The formula is shown below:

$$l_{TCKD} = \mathcal{KL}\left(b^{\theta_i}\|b^{\theta_i^t}\right) \quad (6)$$

$$l_{NCKD} = \mathcal{KL}\left(\hat{p}^{\theta_i}\|\hat{p}^{\theta_i^t}\right) \quad (7)$$

$$\mathcal{KL}_i(\theta_i\|\theta_i^t, x) = l_{TCKD}(\theta_i\|\theta_i^t, x) + \mathcal{N} l_{NCKD}(\theta_i\|\theta_i^t, x) \quad (8)$$

where $\mathcal{KL}_i(\theta_i||\theta_i^t, x)$, $l_{TCKD}(\theta_i||\theta_i^t, x)$ and $l_{TCKD}(\theta_i||\theta_i^t, x)$ respectively denotes the Kullback-Leibler(KL) divergence between the teacher model θ_i and the student model θ_i^t of the i-th client, the KL divergence of the target class, and the KL divergence of the non-target class, x denotes the sample, $\mathcal{N}$ is the weights of the non-target class. In summary, the formula for computing the local loss function for each client is presented as follows:

$$\mathcal{L}_i(\theta_i^t, x, y) = \alpha\, \mathcal{F}_i(\theta_i^t, x, y) + (1-\alpha)\mathcal{KL}_i(\theta_i || \theta_i^t) \quad (9)$$

where $\mathcal{F}_i(\cdot)$ and $\mathcal{KL}_i(\cdot)$ represents the cross-entropy loss and the KL loss of the i-th client, $1-\alpha$ denotes the weight of KL loss, x denotes the sample, and y denotes the label.

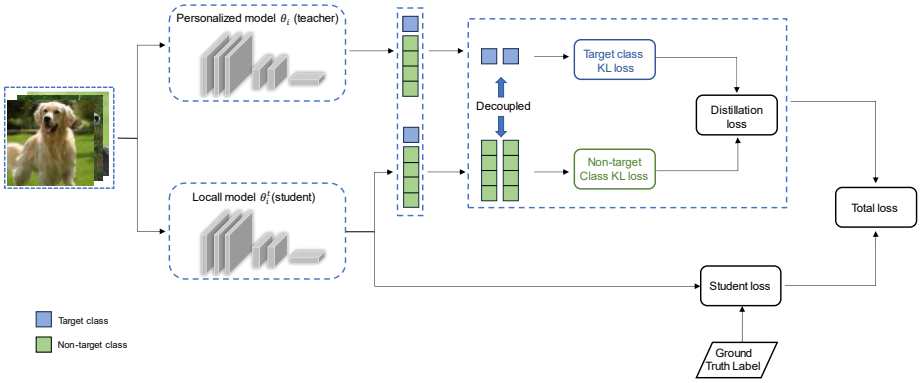


Fig. 2 Self-Decoupled Knowledge Distillation

4.3 Adaptive Clipping Threshold Method

Currently, in federated learning, the primary goal of optimizing user-level differential privacy is to balance privacy and utility. Clipping is an indispensable step in user-level differential privacy, where the risk of privacy leakage is reduced by limiting the norm of model updates within a specific range, the upper bound of which is known as the clipping threshold. Moreover, the clipping method involves scaling all parameters based on the ratio of the norm of model updates to the clipping threshold. Notably, the clipping threshold also impacts the noise addition process. Specifically, as the model tends towards convergence in the later stages of training, the norm of the model updates gradually decreases. With a fixed clipping threshold, if the threshold consistently exceeds the norm of the model updates, the ratio of injected noise to the actual model update parameters becomes increasingly significant, leading to severe model distortion or even rendering the model unusable; conversely, if the clipping threshold is consistently less than the norm of the model updates, a substantial amount of model parameter information is lost during training, thus slowing down the training process. Consequently, both excessively high and low clipping thresholds can affect model performance.

Furthermore, each training round considers the dynamism of user-level differential privacy in personalized federated learning, where the clipping threshold of different communication rounds affects model performance. Thus, the clipping threshold for each round should be different. The model design incorporates this issue, dynamically updating global information each round to ultimately achieve the global optimal strategy, which aims to enhance model performance as much as possible while maintaining the same level of privacy protection.

This chapter proposes an adaptive clipping threshold method designed to find the appropriate clipping threshold C^t for each client in every round, allowing the algorithm

to maximize its contribution to model performance while protecting privacy. Hence, the adaptive clipping threshold issue in the ACDP-pFSD algorithm can be transformed into an optimization problem P, defined as follows:

$$P := \sum_{t=1}^{T} \max_{C^t} \{\phi\} \tag{10}$$

where C^t represents the clipping threshold for round t, while ϕ signifies the contribution of the clipping threshold to model performance.

Evaluation Metrics

Due to the impracticality of numerically assessing clipping thresholds directly, this section considers the impact of the clipping threshold on model performance, treating its contribution as an indicator of the threshold's merit. The loss function calculates the discrepancy between model predictions and actual values, serving as a measure of model performance. l_i^t is the loss value for client i's personalized model over the local test set in round t Extremely low or high loss values can skew the average loss of the overall personalized models, thereby not reflecting true model performance accurately. Consequently, this section defines the loss for the personalized models as the average value of all client losses within the 10th to 80th percentile, expressed as: $loss^t = \frac{1}{|M|}\sum_{l_i^t \in M} l_i^t$. Here, M is the set of loss values that fall between the 10th and 80th percentiles after sorting all client losses in round t, and $|M|$ is the number of elements in set M.

Definition 5 Loss Difference

During any given rounds $t-1$ and t, ϕ_l denotes the average loss disparity across all personalized models:

$$\phi_l = loss^{t-1} - loss^t \tag{11}$$

Here, $loss^{t-1}$ and $loss^t$ respectively represent the average loss values of the personalized models for rounds $t-1$ and t. A positive ϕ_l indicates a reduction in loss and an improvement in model performance.

Regarding model accuracy, which directly reflects predictive and generalization capabilities, ac_i^t is the accuracy of client i's personalized model traversing the local test set in round t. To mitigate the influence of outlier accuracies on the average accuracy of personalized models, this section defines the accuracy as the average of all client accuracies within the 10th to 80th percentiles: $acc^t = \frac{1}{|K|}\sum_{ac_i^t \in K} ac_i^t$. Here, K is the set of accuracy values ranked between the 10th and 80th percentiles in round t, and $|K|$ is the number of elements in set K.

Definition 6 Accuracy Difference

During any given rounds $t-1$ and t, ϕ_a represents the change in the average accuracy across all personalized models:

$$\phi_a = acc^t - acc^{t-1} \tag{12}$$

where acc^t and acc^{t-1} respectively represent the average accuracies of the personalized models for rounds t and $t-1$. A positive ϕ_a indicates an increase in accuracy, signifying improved model performance on the test set.

Problem Modeling

Because the clipping threshold C^t impacts model performance, which in turn affects the choice of C^{t+1}, the static optimization problem P cannot be directly solved. Therefore, this section models the adaptive clipping issue within personalized federated learning as a reinforcement learning problem and transforms the decision-making process of the clipping threshold into a Markov decision process, aiming for the model to converge optimally under a constant level of privacy protection. This transformation involves defining a quintuple $\langle S, A, P, R, \gamma \rangle$, which describes the process of the server selecting the clipping thresholds.

S represents the state space. In the training process of federated learning, the global model is updated at the end of each communication round, and the model's performance can be reflected through the accuracy and loss values on the test set. Therefore, the state space for round t is defined as $s^t = \{acc^t, loss^t\}$.

A denotes the action space. In the adaptive clipping threshold issue, the agent is responsible for selecting actions and deciding the clipping thresholds for each communication round. Actions are defined as a list of continuous variable clipping thresholds C^t and the action for round t is defined as $a^t = C^t$.

P represents the state transition probability matrix. In the adaptive clipping threshold issue, the unknown environment leads to an indeterminate state transition probability matrix. Therefore, this article estimates the state using the state value function and action value function.

R represents the reward function, which calculates the reward r^t that the agent will receive under state s^t if action a^t is executed. The reward is defined as the total contribution of the clipping threshold to the model performance ϕ, which is computed through a linear combination of equations (11) and (12) as $\phi = \phi_l + \phi_a$. The reward function is defined as follows:

$$r^t = R(s^t, a^t) = \phi \tag{13}$$

Thus, the static problem P is transformed into a dynamic Markov decision process P_o, defined as:

$$P_o := \max_{\pi_\omega} \sum_{t=1}^{T} \{\gamma^{t-1} \cdot r^t\} \tag{14}$$

where $\gamma \in (0,1)$ is the discount factor for the reward, with values closer to 1 indicating a greater emphasis on long-term rewards. Additionally, π_ω can be understood as a function that generates actions from states.

Specifically, given a continuous action space, the ACDP-pFSD algorithm can identify the optimal solution for the Markov decision process. This process is divided into four steps: (1) If it is the first round, actions are randomly initialized and corresponding clipping thresholds are obtained. Otherwise, the server first updates the policy then receives the current state from the environment and selects the optimal action through the policy to obtain the clipping threshold. The global model and the clipping threshold are then transmitted to clients. (2) Clients begin local training. Each client first downloads the global model and uses it as a local model to train on their local dataset through the self-decoupled distillation method. Subsequently, each client saves the updated model as a personalized model and calculates the model update amount for this round. The

clients then clip the model update amount using dynamically selected clipping thresholds and dynamically inject Gaussian noise. Finally, they test the performance of the personalized model on the local dataset and send the perturbed model update amount and relevant information back to the server. (3) After the second round of communication, the server calculates rewards, and obtains information of the next round's state, storing the trajectory tuples in the experience replay buffer. (4) When the experience replay buffer is full, the agent updates the parameters of the actor and critic networks. In addition, the framework ACDP-pFSD of this paper satisfies the UDP, and the privacy analysis is the similar as that in the previous work [22].

Algorithm 2： Adaptive clipping threshold selection

Input: State space S, Action space A, Experience replay buffer D_P, Actor Net work π_ω with weights ω, Critic Network V_μ with weights μ, Noise scale σ, Communication rounds T, clipping threshold C^t, Global model θ^t, Learning rate η, Local batch size B, Local epochs Q, Dataset D_i

Output: Clipping threshold C^t, Client updates $\{\tilde{\Delta}_i^t\}$

if $t < 3$ **then**
    Initialize randomly action;
else
    Update $\pi_{\omega_{old}} \leftarrow \pi_\omega$;
    $s^t \leftarrow \{loss^t, acc^t\}$;
    Feed s^t into policy network π_ω;
    Obtain the action $a^t$and clipping threshold C^t;
end if
    for each client $i \in P$ in parallel **do**
        $\theta_i^t \leftarrow$ Download θ^t;
        for $q = 1,2, \dots, Q$ **do**
            Sample batch $B \subseteq D_i$;
            $\theta_i^{t,q} \leftarrow \theta_i^{t,q-1} - \eta \frac{1}{|B|} \sum_{(x,y)\in B} \nabla \mathcal{L}_i(\theta_i^{t,q-1}, x, y)$;
        end for
        $\theta_i \leftarrow \theta_i^{t,Q}$;
        $\Delta_i^t \leftarrow \theta_i^{t,Q} - \theta_i^{t,1}$;
        $\bar{\Delta}_i^t = \Delta_i^t \cdot \min\left(1, \frac{C^t}{\|\Delta_i^t\|_2}\right)$;
        $\tilde{\Delta}_i^t = \bar{\Delta}_i^t + \mathcal{N}\left(0, \frac{(\sigma C^t)^2}{|P^t|}\right)$;
        $\theta_i^{t+1} \leftarrow \theta_i^t + \tilde{\Delta}_i^t$;
        $l_i^{t+1}, ac_i^{t+1} \leftarrow$ Test local model θ_i^{t+1};
    end for
if $t \geq 3$ **then**
    Calculate the reward r^t;
    Get the state $s^{t+1} \leftarrow \{loss^{t+1}, acc^{t+1}\}$;
    Store the experience $\langle s^t, a^t, r^t, s^{t+1} \rangle$ into Trajectory memory τ;

end if
if $(t-2)\%|D_P| == 0$ **then**
  for $m = 1,2,\dots,M$ **do**
    Sample experiences U from D_P;
    /* Use experiences update Actor network and Critic network */
    Update π_ω using PPO;
    Update the Critic network by minimizing the loss function: $\frac{1}{U}\sum_{j=1}^{U}\left[r_j + \gamma V_\mu(s_{j+1}) - V_\mu(s_j)\right]^2$;
  end for
end if
return $C^t, \tilde{\Delta}_i^t$

5 Experiments

In this subsection, we conduct an extensive experimental evaluation of the effectiveness of our proposed framework, ACDP-pFSD. Additionally, we perform ablation studies to validate the efficacy of our proposed personalized federated self-decoupled algorithm, hereafter referred to as pFSD.

5.1 Experimental settings

Experimental Setup. This study assumes two different FL scenarios: 1) Fifty clients with a sampling rate $q = 100\%$; 2) One hundred clients with a sampling rate $q = 50\%$. Selected clients perform $Q = 5$ local training iterations per round and $T = 100$ rounds of communication, with a batch size of 64.

Datasets and Models. We utilize three datasets in our study: EMNIST [32], SVHN [33], and CIFAR10 [34], where SVHN is a real-world dataset of house numbers, and all three are image datasets. The specifics are detailed in Table 1. For each dataset, we employ a pathologically non-i.i.d. distribution: each client holds up to s shards, representing the maximum number of classes a client can possess. As s decreases, the degree of data heterogeneity increases. We consider two levels of heterogeneity, s_1 and s_2, as detailed in Table 2. For the EMNIST dataset, a simple 8-layer CNN model is used, while for CIFAR10 and SVHN, following prior work, we utilize the ResNet18 model.

Table 1. Statistics of the Datasets

Dataset	#Samples	#class	Task
EMNIST	131600	42	Image classification
SVHN	99289	10	Image classification
CIFAR10	60000	10	Image classification

Hyperparameter Settings. Regarding FedPer, this paper preserves the last two layers of the CNN model or the final block of the ResNet18 model as the foundational layers. For pFedSD, the distillation temperature KT is set at 3. ACDP-pFSD employs a non-target class weight of $\mathcal{N} = 0.25$. In terms of privacy protection, consistent with the

approach in [16], the privacy parameter δ for all datasets is defined as $\delta = \frac{1}{P^{1.1}}$, ensuring $\delta < \frac{1}{P}$. Additionally, the fix clipping threshold C for the EMNIST dataset is 0.5, whereas for CIFAR10 and SVHN, C is set at 1.5. This paper also explores the impact of different noise factors σ on accuracy. For the EMNIST dataset, the noise factors of σ are set at 0.8, 1, and 1.3, which correspond to privacy budgets of 112.56, 77, and 50.01 after 100 rounds, respectively. For the CIFAR10 and SVHN datasets, the noise factors of σ are set 1, 1.5, and 2.1, with privacy budgets of 77, 39.78, and 23.55 after 100 rounds, respectively.

Table 2. Shards Setings of the Datasets

Dataset	s_1	s_2	#class
EMNIST	15	5	42
SVHN	4	2	10
CIFAR10	4	2	10

Implementation. This study implements all aforementioned baselines in PyTorch. All experiments were conducted on a deep learning server equipped with an 80G A100 GPU.

5.2 Experimental Evaluation

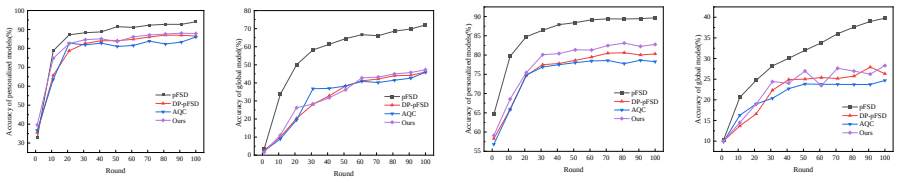
Accuracy Comparison. This paper initially examines the accuracy of models under various noise factors. The noise factor determines the level of privacy protection; a higher noise factor indicates a smaller privacy budget and stronger privacy protection, whereas a lower noise factor indicates a larger privacy budget and weaker privacy protection capabilities. Among these, pFSD is a federated learning method without differential privacy, DP-pFSD employs a fixed clipping threshold, and AQC utilizes adaptive clipping, applied within the same personalized settings as ACDP-pFSD. As observed in Table 3, the proposed ACDP-pFSD algorithm consistently outperforms other algorithms across all three datasets. Particularly at higher levels of privacy protection, the personalized and global model accuracies of the ACDP-pFSD algorithm are significantly superior to those of DP-pFSD and AQC. This phenomenon indicates that model performance is more sensitive to the clipping threshold when more noise is added; an appropriate clipping threshold can enhance model performance, while an inappropriate one can lead to convergence to poorer performance. Furthermore, the performance of all algorithms decreases as the level of privacy protection increases, due to the greater amount of noise added, which distorts the model and directly impacts performance.

Accuracy Analysis at the Same Level of Privacy Protection. Under the heterogeneous setting s_2 and noise factor $\sigma = 1$, this section compares the performance of personalized and global models across different datasets, as illustrated in Fig. 3. Observing Fig. 3(a), it is apparent that on the EMNIST dataset, the personalized and global model performances utilizing AQC method are the lowest. Moreover, the accuracy initially increases rapidly but soon converges to a lower performance level. This could be attributed to the initially reasonable adaptive clipping thresholds, which become progressively less appropriate as the training advances, leading to insufficient information within the model and thus hindering its training. In contrast, the DP-pFSD method with

a fixed clipping threshold converges slower but ensures normal model training. However, its personalized and global model accuracies are suboptimal due to accumulating biases and errors introduced by the fixed threshold as training progresses. The proposed ACDP-pFSD algorithm achieves optimal accuracy in both personalized and global models by dynamically finding appropriate clipping thresholds in the action space during training via the Proximal Policy Optimization (PPO) method. This process prevents selecting excessively large thresholds that reduce model accuracy, as well as overly small ones that could impair normal training, thereby balancing privacy protection and model utility. As indicated by Fig. 3(b), on the CIFAR10 dataset, the ACDP-pFSD algorithm also exhibits the best performance, achieving the highest accuracies in both model types. Notably, at the onset of training on the more complex CIFAR10 dataset, the accuracy of ACDP-pFSD occasionally falls below that of other methods. This is primarily due to the PPO method sometimes selecting inappropriate thresholds during the exploration phase. However, in the later stages of training, this method consistently identifies suitable thresholds, thus balancing the model's privacy and utility aspects.

Table 3. Comparison of accuracy on different noise fators

Dataset	Model	Noise factor	NoDP	DP-pFSD	AQC	ACDP-pFSD(Ours)
EMNIST	Personalized	0.8	94.07	87.18	87.02	**88.80**
		1	94.07	86.51	85.98	**87.85**
		1.3	94.07	85.63	84.53	**86.75**
	Global	0.8	72.05	48.91	48.03	**49.55**
		1	72.05	46.01	45.87	**47.24**
		1.3	72.05	35.11	35.96	**38.49**
CIFAR10	Personalized	1	89.68	80.30	78.28	**82.75**
		1.5	89.68	76.28	76.14	**77.77**
		2.1	89.68	74.67	73.28	**75.62**
	Global	1	39.76	26.30	24.69	**28.30**
		1.5	39.76	20.29	22.81	**23.29**
		2.1	39.76	18.81	18.99	**19.21**
SVHN	Personalized	1	97.41	93.01	91.50	**94.40**
		1.5	97.41	91.14	89.54	**91.98**
		2.1	97.41	80.73	80.21	**82.92**
	Global	1	88.00	53.95	52.24	**55.64**
		1.5	88.00	41.37	47.88	**49.65**
		2.1	88.00	20.07	25.69	**26.33**

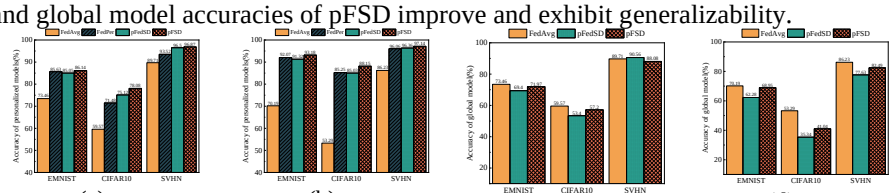


(a)EMNIST (b)CIFAR10

Fig. 3 Comparison of model accuracy on different datasets

Compared to the heterogeneous setting s_1, setting s_2 exhibits an enhanced degree of heterogeneity. In this chapter, with $|P| = 100$ and $p = 0.5$, the performance of personalized and global models across all algorithms within the heterogeneous settings s_1 and

s_2 is evaluated, as illustrated in Fig. 4. Observations from Fig. 4(a) and Fig. 4(b) reveal that the personalized model accuracy of pFSD consistently surpasses that of other algorithms. As heterogeneity shifts from s_1 to s_2, the accuracy of personalized models in the EMNIST and CIFAR10 datasets for FedPer, pFedSD, and pFedSD methods increases, attributing to the resilience of these personalized approaches under extreme data heterogeneity. From Fig. 4(c) and Fig. 4(d), it is evident that the global model accuracy of pFSD closely approaches that of FedAvg and maintains a distinct advantage under the setting s_1, due to the self-decoupled distillation method having the ability to learn large amounts of dark knowledge. However, as heterogeneity progresses from s_1 to s_2, there is a decline in global model accuracy, reflecting the inevitable performance deterioration in more heterogeneous data distributions. The experimental outcomes demonstrate that under varying degrees of heterogeneity, both the personalized and global model accuracies of pFSD improve and exhibit generalizability.



(a) s_1 (b) s_2 (c) s_1 (d) s_2

Fig. 4 Accuracy of models on different datasets

6 Conclusion

In this paper, a simple and effective personalized federated distillation learning scheme with user-level differential privacy, called ACDP-pFSD, is proposed to address the challenges of data heterogeneity and privacy leakage. The method in this paper fully learns historical personalized knowledge through self-decoupled distillation, which can improve personalized model accuracy, enhance global model generalization, and reduce noise through adaptive model updates clipping, achieving a better trade-off between model performance and privacy protection. Experimental results on multiple datasets show that ACDP-pFSD outperforms current state-of-the-art methods for both the same level of heterogeneity and privacy preservation.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
2. Li, Y., Wen, G.: Research and practice of financial credit risk management based on federated learning. Engineering Letters 31(1) (2023)
3. Wang, W., Li, X., Qiu, X., Zhang, X., Brusic, V., Zhao, J.: A privacy preserving framework for federated learning in smart healthcare systems. Information Processing & Management 60(1), 103167 (2023)
4. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: International conference on machine learning. pp. 2089–2099. PMLR (2021)
5. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning 14(1–2), 1–210 (2021)
6. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Advances in neural information processing systems 32 (2019)
7. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 739–753. IEEE (2019)
8. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)
9. Sattler, F., Müller, K.R., Samek, W.: Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE transactions on neural networks and learning systems 32(8), 3710–3722 (2020)
10. Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.S.: Federated multi-task learning. Advances in neural information processing systems 30 (2017)
11. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: Fair and robust federated learning through personalization. In: International Conference on Machine Learning. pp. 6357–6368. PMLR (2021)
12. Jin, H., Bai, D., Yao, D., Dai, Y., Gu, L., Yu, C., Sun, L.: Personalized edge intelligence via federated self-knowledge distillation. IEEE Transactions on Parallel and Distributed Systems 34(2), 567–580 (2022)
13. Mei, Y., Guo, B., Xiao, D., Wu, W.: Fedvf: Personalized federated learning based on layer-wise parameter updates with variable frequency. In: 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). pp. 1–9. IEEE (2021)
14. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 9(3–4), 211–407(2014)
15. Zhang, X., Chen, X., Hong, M., Wu, Z.S., Yi, J.: Understanding clipping for federated learning: Convergence and client-level differential privacy. In: International Conference on Machine Learning, ICML 2022 (2022)
16. McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963 (2017)
17. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557 (2017)
18. Andrew, G., Thakkar, O., McMahan, B., Ramaswamy, S.: Differentially private learning with adaptive clipping. Advances in Neural Information Processing Systems 34, 17455–17466 (2021)

19. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security 15, 3454–3469 (2020)
20. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International conference on artificial intelligence and statistics. pp. 2938–2948. PMLR (2020)
21. Cheng, A., Wang, P., Zhang, X.S., Cheng, J.: Differentially private federated learning with local regularization and sparsification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10122–10131 (2022)
22. Shi, Y., Liu, Y., Wei, K., Shen, L., Wang, X., Tao, D.: Make landscape flatter in differentially private federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24552–24562 (2023)
23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
24. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 11953–11962 (2022)
25. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581 (2019)
26. Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: International conference on machine learning. pp. 12878–12889. PMLR (2021)
27. Sui, D., Chen, Y., Zhao, J., Jia, Y., Xie, Y., Sun, W.: Feded: Federated learning via ensemble distillation for medical relation extraction. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). pp. 2118–2128 (2020)
28. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. Advances in Neural Information Processing Systems 33, 2351–2363 (2020)
29. Shen, T., Zhang, J., Jia, X., Zhang, F., Huang, G., Zhou, P., Kuang, K., Wu, F.,Wu, C.: Federated mutual learning. arXiv preprint arXiv:2006.16765 (2020)
30. Shen, Y., Zhou, Y., Yu, L.: Cd2-pfed: Cyclic distillation-guided channel decoupled for model personalization in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10041–10050 (2022)
31. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
32. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: Emnist: Extending mnist to handwritten letters. In: 2017 international joint conference on neural networks (IJCNN). pp. 2921–2926. IEEE (2017)
33. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
34. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)