PDP-FD: Federated Knowledge Distillation Based on Personalized Differential Privacy

Abstract. Federated Learning (FL) is a distributed machine learning framework where each participant achieves model training by exchanging model parameters. However, when the server is dishonest, it may lead to the leakage of the private data of participants. Furthermore, FL is also affected by non-independent and identically distributed (Non-IID) data, resulting in a decrease in model accuracy. At the same time, Non-IID data may also lead to different privacy protection requirements among participants, making it difficult to handle them uniformly. In this paper, we propose a Federated Knowledge Distillation Based on Personalized Differential Privacy (PDP-FD) that considers both data heterogeneity and privacy protection problems. To solve the data heterogeneity problem, we adopt a network structure with a personalization layer and propose a strategy of dynamically adjusting the personalization layer. By adjusting the personalization layer, we can adequately preserve local features while better adapting to the data characteristics among different participants. To solve the problem of different privacy requirements among participants, we propose a personalized privacy budget allocation strategy, which adds appropriate noise based on the training state of the local model to achieve personalized privacy protection. Finally, the experimental results show the effectiveness of our mechanism and its superior performance over other differential privacy FL schemes.

Keywords: Personalized Differential Privacy, Federated Learning, Data Heterogeneity, Knowledge Distillation.

1 Introduction

Federated Learning [1] is a distributed machine learning framework in which data is locally stored and trained, and only updated parameters are transmitted to the server for aggregation to train a global model jointly. However, this framework unavoidably faces data heterogeneity and privacy problems.

The problem of data heterogeneity primarily is mainly caused by the different distribution of data [2]. Each client may have different limitations and requirements during the data collection and processing stages. These factors may lead to different data distributions, resulting in certain differences in the models trained on different clients. Additionally, although federated learning keeps the data locally to avoid cross-domain sharing, the model parameters information may still pose potential privacy risks [3]. Malicious attackers can exploit various attack techniques, such as inference attacks [4] and model inversion attacks [5], to acquire clients' private information [6–8].

As the two main challenges of FL, the above two issues are widely studied in existing work. On one hand, some studies propose personalized federated learning approaches, where the global model is personalized on the client to obtain a personalized model. For example, fine-tuning the local model after obtaining the global model [9]; passing

knowledge from the server to the client [10]; adding a personalization layer to the local model [11]. On the other hand, in order to enhance privacy, various technologies such as homomorphic encryption [12], secure multi-party computation [13], and differential privacy [14, 15] are commonly used for privacy protection. Homomorphic encryption and secure multiparty computation rely on cryptographic techniques to encrypt communication content, as the encryption process requires additional computing power, resulting in high interaction costs and computational overhead. Conversely, differential privacy accumulates lower computational costs as it doesn't necessitate encrypting communication content. Moreover, it boasts rigorous mathematical proofs and solid background knowledge in information theory, making it a focal point of research in the field of data security.

In the Non-IID environment, the gradients generated by different clients have differences [16]. When the same distribution of noise is added, it causes different perturbations to the gradient, consequently affecting the direction of gradient descent and convergence speed. The model requires more gradient update steps to achieve the expected performance level, which can lead to a cumulative increase in privacy costs during each gradient calculation and convergence process [17]. To solve the above problems, this paper proposes a new framework called PDP-FD. This framework uses base layer and personalization layer to implement personalized federated learning. On the server side, knowledge distillation techniques are used to improve the generalization ability of the model base layer. On the client side, the personalization layer is dynamically adjusted to retain local features while reducing the amount of added noise. We optimize the model base layer and personalization layer of the model separately to reduce the impact on model performance under Non-IID data. In addition, we propose a personalized privacy budget allocation strategy to better balance privacy and utility by allocating different privacy budgets, ensuring that the different privacy requirements of each client are adequately fulfilled. The main contributions of this paper are as follows:

- We propose a federated knowledge distillation based on personalized differential privacy (PDP-FD), which achieves knowledge distillation on the server through sharing layers, thereby improving the generalization ability of the base layer. Additionally, the algorithm dynamically adjusts the personalized layers on the client side, allowing the client to better retain local personalized information and thereby reduce the impact of data heterogeneity on the model's performance.
- We propose a personalized privacy budget allocation strategy. In this strategy, we consider the data heterogeneity among clients, evaluate the training state of each client based on the similarity between the global model and the local model, and allocate an appropriate privacy budget to each client to meet their different privacy requirements.
- We evaluate the performance of the PDP-FD algorithm on the CIFAR-10 and MNIST datasets. Experimental results demonstrate that this algorithm outperforms existing differentially private federated learning methods in terms of model accuracy, convergence speed, and privacy cost.

The rest of the paper is organized as follows. In Section 2, we present related work on data heterogeneity and privacy protection in federated learning; in Section 3, we present the required preparatory knowledge; in Section 4, we describe the PDP-FD

algorithm in detail; in Section 5, we analyze and summarize the experimental results; finally, we conclude the full paper in Section 6.

2 Related Works

2.1 Data Heterogeneity in FL

Due to the different data distributions of each client, it is difficult to train a global model that can fit all clients. To solve this problem, personalized federated learning has recently received a lot of attention. Firstly, Arivazhagan et al. [11] proposed the FedPer algorithm which for the first time divides the neural network architecture into a base layer and a personalization layer for personalized federated learning. This approach enables personalized federated learning by retaining personalization layers locally. Based on FedPer, Liang et al. [18] proposed the opposite algorithm LG-Fed, which uses the bottom layer as the personalization layer and the top layer as the base layer, and keeps the top layer shared by all clients. Although this approach can reduce the impact of data heterogeneity, but it requires prior knowledge for the determination of which layers to be personalized. To alleviate the problem of hierarchical selection and effectively solve the problem of data heterogeneity, Shen et al. [19] proposed a novel channel decoupling paradigm to decouple the global model at the channel dimension for personalization. This approach provides a unified solution to address a broad range of data heterogeneity. To prevent excessive personalization of local models on each client, leading to the inability of the global model to converge, Mei et al. [20] proposed the FedVF algorithm. This algorithm adopts a two-stage approach by periodically aggregating the personalization layer, achieving a better balance between the global model and personalized models. However, most methods have failed to adequately consider the selection of personalization layer when addressing the data heterogeneity problem, particularly in the context of big data and big models, where the impact of the personalization layer on model performance remains crucial. Additionally, some studies have not provided effective privacy protection mechanisms and have failed to consider the influence of different noise scales on the results.

2.2 Privacy Problem in FL

To solve the privacy leakage problem, researchers have applied differential privacy techniques to federated learning for enhanced privacy protection [21–23]. However, most current methods do not fully consider the impact of data heterogeneity on privacy. These methods typically employ a uniform noise addition strategy, which makes it difficult to satisfy the different privacy requirements of individual users. Huang et al. [24] proposed a novel differential privacy federated learning called DP-FL. This framework introduces adaptive noise addition on the client side. Unlike traditional methods for privacy budget allocation, DP-FL divides the privacy budget into two parts. One part is used to add noise to the gradient, while the other part is used to select the optimal step size. Dong et al. [25] proposed a personalized and adaptive differential privacy federated meta learning framework, PADP-FedMeta, which allocates different privacy

budgets according to the distribution of client sample labels. Yang et al. [26] proposed an algorithm PLU-FedOA, where PLU allows clients to update parameters locally and select a personalized privacy budget, and FedOA helps the server to aggregate the optimized local parameters. Shen et al. [27] proposed a federated learning scheme based on personalized local differential privacy by tuning the privacy budget parameters and introducing a different for each client security range parameters to achieve personalized privacy protection. Although differential privacy technology has been widely applied in federated learning, the trade-off between model performance and privacy protection level remains the main problem currently faced.

3 Preliminaries

3.1 Federated Learning

Federated learning can be defined as follows: suppose there are N users who want to participate in the training process of machine learning, and the datasets of these users are denoted as $\{D_1, D_2, \dots, D_N\}$. Each user is trained using their local dataset by minimizing the local loss function:

$$w_i^* = arg\ min\ L(D_i, w_i) \tag{1}$$

where $L(D_i, w_i)$ is the loss function of user i on the dataset D_i.

The user sends the updated model to the server to update the global model parameters through averaging or weighted averaging methods. The formula is as follows:

$$W_G = \sum_{i=1}^{N} p_i w_i \tag{2}$$

where w_i is the model parameter uploaded by the i-th participating user, W_G is the model parameter after aggregation, N is the number of participating users, p_i represents the proportion of data from the i-th participating user in relation to the total amount of data.

3.2 Knowledge Distillation

Knowledge distillation [28] is a technique for model compression, whereby the knowledge of a teacher model is transferred to a student model with the assistance of auxiliary datasets. By optimizing the distance between the outputs of the teacher and student, the objective can be formulated as:

$$L_{total} = \lambda \times L_{KD}\big(p(u,T), p(z,T)\big) + (1-\lambda) \times L_S\big(y, p(z,1)\big) \tag{3}$$

where λ is a hyperparameter, $L_{KD}\big(p(n,T), p(z,T)\big)$ is distillation loss, $L_S\big(y, p(z,1)\big)$ is student loss. u and z represent the logical units generated by the teacher and student models, respectively. T refers to the temperature.

3.3 Differential Privacy

The core idea of differential privacy is to protect the privacy of data by adding noise to the statistical results, ensuring a change in one of the records in the dataset does not significantly affect the results of the algorithm. It is defined as follows:

Definition 1 ($(\varepsilon, \delta) - DP$ [15]). Let $M: D \to \mathrm{R}$ be a random mechanism, where D and D' are neighboring datasets that differ by at most one record. If for any output $S_M \in R$ generated by the randomized mechanism M on D and D', the following inequality holds, then the mechanism M satisfies $(\varepsilon, \delta) - DP$.

$$\Pr[\mathrm{M}(D) \in S_M] \leq e^{\epsilon} \times \Pr[\mathrm{M}(D') \in S_M] + \delta \tag{4}$$

where the parameter ε denotes the privacy budget, which reflects the degree of privacy protection of the algorithm, the smaller ε is the higher the degree of privacy protection. δ is a relaxation factor indicating the probability of privacy disclosure.

To better track and compute the strength of privacy protection, Mironov [29] generalized the$(\varepsilon, \delta) - DP$ concept to(α, ε) −Rényi differential privacy based on the divergence concept and it can be converted to $(\varepsilon, \delta) - DP$.

Definition 2 ($(\alpha, \varepsilon) - RDP$ [29]). For any two neighboring datasets D and D', if there exists a randomized mechanism $M: \mathrm{D} \to R$ that satisfies the following equation, then the randomized mechanism M is said to satisfy $(\alpha, \varepsilon) - RDP$.

$$D_{\alpha}(M(D)||M(D')) = \frac{1}{\alpha - 1}\log\left(E_{\theta \sim M(D')}\left[\left(\frac{M(D)(\theta)}{M(D')(\theta)}\right)^{\alpha}\right]\right) \leq \varepsilon \tag{5}$$

where $D_{\alpha}(M(D)||M(D')) \leq \varepsilon$ indicates that the Rényi divergence between the neighboring datasets is restricted to the privacy parameter ε, when $\alpha \to \infty$, $(\alpha, \varepsilon) - RDP$ reduces to $(\varepsilon, 0) - DP$.

Proposition 1 (Gaussian Mechanism of RDP [29]): Let $S = max_{D,D'}\|M(D) - M(D')\|_2$ be the L_2 sensitivity of function M. Then, for the Gaussian mechanism $f(D) = M(D) + N(0, \sigma^2 I)$, it satisfies $(\alpha, \alpha S^2 \,/\, 2\sigma^2) - RDP$.

Proposition 2 (Conversion from RDP to $(\varepsilon, \delta) - DP$ [29]): if M is a $(\alpha, \varepsilon) - RDP$ mechanism, then for $\forall \delta \in (0,1)$, mechanism M also satisfies $(\varepsilon^*, \delta) - DP$, where ε^* is defined as follows.

$$\varepsilon^* = \varepsilon + \frac{\log(1/\delta)}{\alpha - 1} \tag{6}$$

Proposition 3 (Sequential composition theorem of RDP [29]): Let $M_1: D \to R_1$ satisfy $(\alpha, \varepsilon_1) - RDP$, $M_2: D \to R_2$ satisfy $(\alpha, \varepsilon_2) - RDP$. Then the sequential composition mechanism $M_{1,2}$ of M_1 and M_2 satisfies $(\alpha, \varepsilon_1 + \varepsilon_2) - RDP$.

Proposition 4 (Parallel Composition Theorem of RDP [30]): Assuming mechanism M consists of n random mechanisms $M_1, \ldots, M_n$, where each mechanism M_i satisfies $(\alpha, \varepsilon_i) - RDP$. Let there be n mutually independent datasets $D_1, \ldots, D_n$. The composite

mechanism $M(M_1(D_1), \ldots, M_n(D_n))$ formed by these mechanisms satisfies $\left(\alpha, \max_i \varepsilon_i\right) - RDP$.

Proposition 5 (Postprocessing [31]): Suppose there is a mechanism M satisfies $(\varepsilon, \delta) - DP$ and another randomized mechanism A. The mechanism M after the operation of mechanism A still satisfies $(\varepsilon, \delta) - DP$.

4 Federated Knowledge Distillation Based on Personalized Differential Privacy

In this section, we first give an overview of the PDP-FD framework. Next, we present the relevant algorithms involving training. Finally, we analyze the differential privacy guarantees provided by PDP-FD and provide corresponding proofs.

4.1 PDP-FD Framework

In this subsection, we design the PDP-FD, Fig. 1 shows our proposed PDP-FD framework, and Table 1 demonstrates the main symbols used in this paper.

Table 1. Main symbols.

Symbols	Description
D_i	The dataset held by the client i
D_{aux}	Auxiliary datasets
G	Global model
Ens	Ensemble model
L_p	Number of personalization layer
V_i^t	Vote of client i in round t
C_{in}	Number of clients with increasing accuracy
C_{de}	Number of clients with decreasing accuracy
W_{base}^t	The base layer parameters in round t
W_{per}^t	The personalization layer parameters in round t
s_i	Similarity
ε	Privacy budget
g	Gradient

In the PDP-FD framework, each local model is trained on its respective dataset (step 1) and privacy protection is achieved through different privacy (step 2). The local models in this framework are divided into the base layer and the personalization layer. The parameters of the base layer are uploaded to the server, while the personalization layer remains local, which helps the model overcome the adverse effects of heterogeneous data. Therefore, to better preserve local personalized information, we propose a strategy to dynamically adjust the personalization layer.

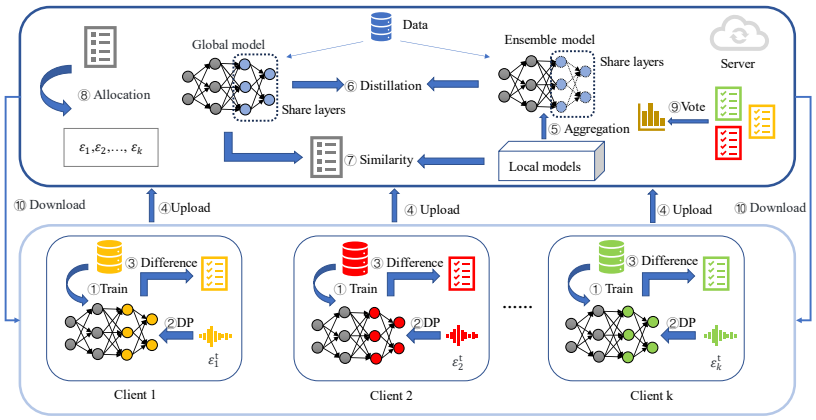


Fig. 1. PDP-FD framework

In this strategy, we compute the difference in model accuracy between neighboring rounds (step 3) and use V_i^t to count the number of clients whose accuracy increases and decreases (i.e., C_{in} and C_{de}). When C_{in} is greater than C_{de}, we improve the accuracy by adding local personalized information. Conversely, we allow more base layers to learn more knowledge on the server to improve the performance of the local model (step 9). If the model continues to dynamically adjust the layers when it is close to convergence, it may result in fluctuations and fail to reach the optimal solution. Therefore, we select the most frequently chosen layers in the early dynamic adjustments as fixed layers to complete the final training of the model.

Afterward, the updated parameters of the base layers are uploaded to the server (step 4), and they are aggregated to generate the ensemble model, Ens (step 5). To enhance the generalization ability of the global model, we utilize the auxiliary dataset, D_{aux}, to mimic the local knowledge and minimize the loss between the ensemble model and the global model (step 6):

$$\min_{w} E_{x\sim D_{aux}}\left[L\left(G(x;w), Ens\left(x, (Ens_{base}^{t}, G_{per}^{t-1})\right)\right)\right] \tag{7}$$

where Ens_{base}^{t} represents the base layer of the ensemble model, G_{per}^{t-1} denotes the shared layer, G is the global model, L represents the loss between the global model and the ensemble model.

The PDP-FD algorithm is shown in Algorithm 1. The personalized layer of the local model is combined with the base layer provided by the server to form a new local model (Line 7), which is then trained (Lines 8-15). Additionally, we compare the accuracy between two rounds of local models and return V_i^t for voting (Line 16). The server aggregates the uploaded base layer parameters $W_{i,base}^{t}$ and obtains the ensemble model Ens_{base}^{t}. Knowledge distillation is performed between the ensemble model Ens and

Algorithm 1. Federated Knowledge Distillation Based on Personalized Differential Privacy

Input: local datasets D_i, auxiliary datasets D_{aux}, number of clients N, rounds of communication T, number of local epochs E, learning rate η, privacy budget ε, personalization layer L_p, global model G

1: Initialize all model with random weights
2: Send W_{base}^0, ε_i, L_p to each client
3: **for** t =1,2…T **do**
4: // **Client Update**
5: **for** i =1,2…N **do**
6: Receive W_{base}^t, ε_i, L_p to each client
7: $W_i^t \leftarrow (W_{base}^t, W_{i,per}^{t-1})$
8: **for** e =1,2…,E **do**
9: **for** batch b ∈ B **do**
10: $g_t(x_j) \leftarrow \nabla \mathrm{L}(W_i, x_j)$
11: $g_t(x_j) \leftarrow g_t(x_j) / \max\left(1, \frac{||g_t(x_j)||_2}{C}\right)$
12: $\widetilde{g_t} \leftarrow \frac{1}{L}\left(\sum_i g_t(x_j) + N(0,\ \sigma^2 C^2 I)\right)$
13: $W_i \leftarrow W_i - \eta \widetilde{g_t}$
14: **end for**
15: **end for**
16: Contrast Accuracy and vote V_i^t
17: Send $W_{i,base}$, V_i^t, L_p to server
18: **end for**
19: // **Server Update**
20: Receive $W_i^{(t)}$, V_i^t, L_p from client
21: Aggregate $Ens_{base}^t \leftarrow \frac{1}{N}\sum_{i=1}^{N} W_{i,bass}^t$
22: $L_d \leftarrow L(G(x, W^{t-1}), Ens(x, (Ens_{base}^t, W_{per}^{t-1})))$
23: $W^t \leftarrow W^{t-1} - \eta \nabla L_d$
24: $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \leftarrow$ **Algorithm 3** (ε^t, W_{base}^t, $W_{i,base}^{t-1}$)
25: Count the number of C_{in} and C_{de} according to V_i^t
26: **if** $C_{in} > C_{de}$ **then**
27: $L_p = L_p + 1$
28: **else if** $C_{in} < C_{de}$ **then**
29: $L_p = L_p - 1$
30: **end if**
31: Send W_{base}^t, $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$, L_p to each client
32: end for
Return: Model parameters W_i for each client

the global model G to obtain the updated global model parameters W^t (Lines 20-23). Next, Algorithm 2 is used to allocate the privacy budget (Line 24). Finally, using the uploaded V_i^t for voting, we adjust the number of personalized layers based on the comparison between C_{in} and C_{de} (Lines 25-30). The value of L_p is within a certain range.

If the adjusted value exceeds this range, the boundary value is taken as the number of personalization layers.

4.2 Personalized Privacy Budget Allocation Strategy

To achieve privacy protection in Algorithm 1, we add an amount of noise in the gradient update process, and the parameters are updated as follows:

$$W_i^t = W_i^{t-1} - \eta\left(\frac{1}{B}\left(\sum_{x_j \in B} g_t(x_j) + N\right)\right) \tag{8}$$

where N is a random Gaussian noise vector and B denotes the training batch.

However, adding the same amount of noise to the gradient may affect the utility of the model, leading to a difficult balance between model effectiveness and privacy protection. Therefore, we propose a personalized privacy budget allocation strategy that achieves better model performance at cumulatively lower privacy costs.

Firstly, the similarity between the global model and the local model is calculated. At this stage, the global model is the model that has undergone knowledge distillation, gaining improved generalization ability with the assistance of an auxiliary dataset, which can better evaluate the state of local model training. The calculation formula is as follows:

$$s_i = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_i^n (A_i)^2} \times \sqrt{\sum_i^n (B_i)^2}} \tag{9}$$

where vectors A and B represent the parameters of the base layers for the global model and the local model, respectively.

Since the cosine similarity ranges between [-1,1], it needs to be normalized for easy calculation. The formula is as follows:

$$S_i = \frac{s_i - min}{max - min} \tag{10}$$

where s_i is the similarity between the local model and the global model, while S_i represents the normalized similarity. Here, min and max respectively indicate the minimum and maximum values in the interval, which are -1 and 1, respectively.

Calculate the privacy budget for each client from Equation (11).

$$\varepsilon_i^t = \varepsilon^{\mathrm{t}} \times \frac{1\text{-}S_i}{1\text{-}\min(S_i, \ldots, S_n)} \tag{11}$$

where $min\ (S_1, \ldots, S_n)$ is the minimum of all similarities, ε^t is the value of the privacy budget in round t, and ε_i^t is the privacy budget allocated to client i in round t.

The specific process is shown in Algorithm 2. The first step is to calculate the similarity s_i between the local model's base layer and the global model's base layer (lines 1-3). The similarity is then normalized, and the value of the privacy budget is computed (lines 4-7).

Algorithm 2. Personalized Privacy Budget Allocation Strategy.

Input: base layer of global model W_{base}^{t}, base layer of local model $W_{i,bass}^{t-1}$, privacy budget ε^{t}

Output: personalized privacy budget $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$

for i =1,2…n **do**
  Similarity $s_i \leftarrow similarity\ (W_{base}^{t}, W_{i,base}^{t-1})$
end for
for i =1,2…n **do**
  Normalized $S_i \leftarrow \frac{1}{2}(s_i + 1)$
  Compute privacy budget $\varepsilon_i \leftarrow \varepsilon^{t} \times \frac{1-S_i}{1-min(S_1,\ldots,S_n)}$
end for
return $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$

4.3 Privacy Analysis

In this section, we analyze the privacy guarantees of the proposed PDP-FD algorithm.

Theorem 1: Let S be the sensitivity of g, when $\sigma \geq S\sqrt{2\log(1/\delta)}\,/\varepsilon$, for any $\delta \in (0,1)$ and $\varepsilon \leq 2\log(1/\delta)$, the output of Algorithm 1 satisfies $(\varepsilon, \delta) - DP$.

Proof. Let g and g' be the private gradients of two neighboring datasets D and D'. For given sensitivity S

$$S = \max_{D,D'} |g - g'| \tag{12}$$

According to the Gaussian mechanism of RDP (Proposition 1) and sequence composition theorem (Proposition 3), for $\forall \delta \in (0,1)$ we have

$$D_{\alpha}\big(N(0,\sigma^2)||N(S,\sigma^2)\big) = \alpha S^2/(2\sigma^2) \tag{13}$$

According to the above equation, it can be observed that the client's upload process satisfies $(\alpha, \alpha S^2/2\sigma^2)$-RDP, i.e., $(\alpha, \varepsilon) - RDP$.

According to the post-processing mechanism (Proposition 5), when Algorithm 1 processes the output of the differential privacy mechanism, the output still satisfies differential privacy as long as there is no new interaction with the private dataset. Thus, multiple local models still satisfy differential privacy after server aggregation.

In Algorithm 1, the utilization of personalized differential privacy results in varying privacy budgets among different clients. According to parallel combinatoriality (Proposition 4), the global federated learning model satisfies $(\alpha, \varepsilon) - RDP$, where $\varepsilon = \max(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k)$.

Now we have proved that Algorithm 1 satisfies $(\alpha, \alpha S^2/2\sigma^2) - RDP$. In order to make it equivalent to $(\varepsilon, \delta) - DP$, according to the theorem on the conversion of RDP to $(\varepsilon, \delta) - DP$ (Proposition 2), we need the following equation:

$$\frac{\alpha S^2}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1} \leq \varepsilon \tag{14}$$

Choose $\alpha = 1 + \frac{2\log(1/\delta)}{\varepsilon}$ and rearrange Equation (14) we need the following equation:

$$\sigma^2 \geq \frac{S^2(\varepsilon + 2\log(1/\delta))}{\varepsilon^2} \tag{15}$$

Based on the constraints on ε, the proof is derived.

5 Experiments

In this section, we will evaluate the proposed PDP-FD algorithm. To compare it with other federated learning algorithms such as FedPer, FedAvg, and LG-Fed, we will assess their model accuracy on the CIFAR-10 and MNIST datasets. Additionally, we will compare the privacy costs of different privacy budget allocation strategies on the CIFAR-10 dataset.

5.1 Datasets and Models

Dataset. The CIFAR-10 dataset consists of 60,000 32×32 color images, with 6,000 images in each of the 10 different categories. Similarly, the CIFAR-100 dataset contains 600 images per category from 100 different classes, with 500 images used for training and 100 images used for testing. The MNIST dataset comprises 60,000 training images of handwritten digits and 10,000 test images, each with a size of 28×28 pixels and in grayscale. The USPS dataset contains handwritten digit images extracted from United States Postal Service envelopes. It consists of 10 classes, with each class containing 2,000 samples. The images in the USPS dataset are also grayscale and have a size of 28×28 pixels.

Our algorithm requires the use of auxiliary datasets. In our experiments with CIFAR-10 and MNIST, we utilized the CIFAR-100 and USPS datasets as auxiliary datasets, respectively.

Models. ResNet-18, ResNet-34, ResNet-50, CNN. The basic structure of ResNet is composed of multiple residual blocks with convolutional layers. ResNet has different network configurations with various numbers of layers, such as 18, 34, 50, and 101. In this paper, we utilized ResNet-18, ResNet-34, and ResNet-50 models, as well as the Convolutional Neural Network (CNN).

5.2 Implementation Details

Parameter Settings. In the experiment, the local epoch is set to 2, with a total of 100 communication rounds and 10 client participants. The local batch size is set to 128, and the learning rate is set to 0.01. The distillation epoch is set to 2, and the distillation batch size is set to 128. After constructing the local dataset for each user, we randomly selected 20% of the local dataset as the local training dataset. In the experiments involving ResNet, we considered the last fully connected layer and the basic blocks as the personalization layer. The range of the personalization layer was dynamically adjusted between layers 1 and 4. Testing the accuracy, the final performance is

represented by the average accuracy of the local models on their respective local test sets. The comparative federated learning methods include FedAvg [1], FedPer [11], FedProx [32], LD-Fed [18], and FedVF [20].

Dataset Partition. We refer to the method proposed in [11] and introduce an integer H to represent the degree of Non-IID. When H is set to 4, it implies that each client possesses images from a maximum of four different categories. In our experiments, we set the values of H to be 2, 4, and 8.

5.3 Experimental Results and Analysis

The algorithm presented in this paper is referred to as PDP-FD, where the section excluding DP is denoted as PFD. Tables 2, 3, and 4 respectively present the accuracy comparison between PFD, FedAvg, FedProx, FedPer, LG-Fed, and FedVF for H =8, 4, and 2. The bold numbers indicate the highest accuracy.

Table 2. Test accuracy based on CIFAR-10 and MNIST (H=8).

Dataset	Model	FedAvg	FedProx	FedPer	LG-Fed	FedVF	PFD
CIFAR-10	ResNet-18	51.14	51.14	64.97	60.85	69.29	**76.96**
	ResNet-34	53.29	54.65	67.37	58.87	70.57	**72.45**
	ResNet-50	53.64	54.48	64.24	57.51	71.80	**76.12**
MNIST	CNN	44.78	42.79	60.51	**62.74**	60.99	56.36
		94.83	96.50	**99.33**	99.16	99.00	96.55

Table 2 presents a detailed comparison of the accuracy among different algorithms in the scenario with a small degree of data heterogeneity (H=8). On the CIFAR-10 dataset, when using the ResNet-18 model, PFD achieved an accuracy approximately 7% higher than FedVF; On the ResNet-34 model, PFD is about 2% higher than FedVF; About 4% higher on the ResNet-50 model. However, on the CNN model, there is a gap of about 7% between the accuracy of the PFD and LG-Fed algorithms. On the MNIST dataset, the FedPer algorithm performs even better, with accuracy about 3% higher than our method.

Table 3. Test accuracy based on CIFAR-10 and MNIST (H=4).

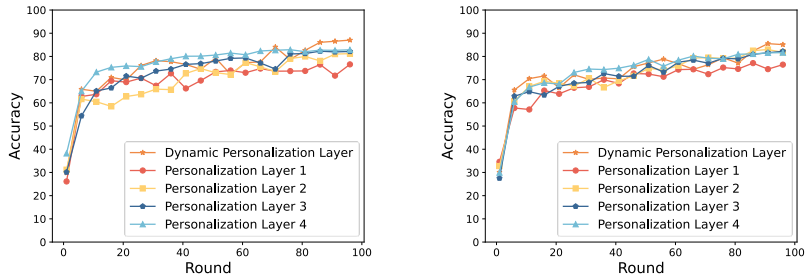
Dataset	Model	FedAvg	FedProx	FedPer	LG-Fed	FedVF	PFD
CIFAR-10	ResNet-18	49.40	48.37	74.08	70.70	76.34	**85.24**
	ResNet-34	50.51	46.97	77.42	75.69	76.31	**83.79**
	ResNet-50	48.58	46.84	73.94	73.67	74.04	**79.93**
MNIST	CNN	39.75	41.72	70.61	71.05	**71.86**	67.85
		95.50	97.33	99.16	99.50	**99.66**	99.16

Table 4. Test accuracy based on CIFAR-10 and MNIST (H=2).

Dataset	Model	FedAvg	FedProx	FedPer	LG-Fed	FedVF	PFD
CIFAR-10	ResNet-18	36.80	34.56	87.36	88.57	87.55	**91.02**
	ResNet-34	36.37	40.18	85.63	89.83	91.33	**92.39**
	ResNet-50	25.41	28.69	83.46	84.51	87.88	**90.20**
MNIST	CNN	33.54	30.83	86.80	**89.32**	88.83	88.44
		94.83	96.50	99.33	**99.66**	98.66	97.50

According to Table 3 and Table 4, it can be observed that as H decreases, the degree of data heterogeneity gradually increases, resulting in a gradual increase in the accuracy of the personalized model. In the residual network, PFD still achieves higher accuracy, while in the small model (CNN), the accuracy is slightly lower than other algorithms. This may be attributed to the fact that our proposed strategy of dynamically adjusting the personalized layer requires locally retaining the personalized layer, whereas the CNN model is relatively shallow with fewer layers, limiting its performance improvement comparatively. Although the performance of PFD on the CNN model has not yet reached its optimal level, our method performs excellently in more complex residual networks.

To further validate the effectiveness of this strategy, we compare the accuracy between the fixed personalized layers and dynamically adjusted personalized layers.

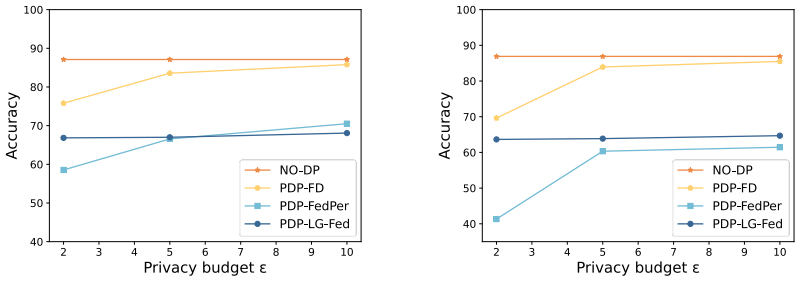


(a) ResNet-18 (b) ResNet-34

Fig. 2. Test accuracy for different personalization layers on the CIFAR-10

Fig. 2 shows the accuracy test results of our proposed algorithm on the CIFAR-10 dataset with a different number of personalization layers. In the ResNet-18 model, Fig. 2(a) shows that when the personalized layer is fixed at 4, the highest accuracy achieved is 84.02%. However, our proposed strategy of dynamically adjusting the personalized layer (indicated by the orange asterisk line) results in an accuracy improvement of approximately 3%, reaching 87.1%. In the ResNet-34 model, Fig. 2(b) shows that the highest accuracy of 82.83% is achieved when the number of fixed personalization layers is 2. However, our method reaches a maximum accuracy of 86.9%, which is about a 3% improvement over the fixed personalization layer approach. The experiment

further demonstrates the effectiveness of our proposed strategy, while eliminating the cumbersome steps of selecting personalized layers.



(a) CIFAR-10 (ResNet-18) (b) CIFAR-10 (ResNet-34)

Fig. 3. Test accuracy for different ε on the CIFAR-10

In Fig. 3, we compare the model accuracy under different privacy budgets (ε = 2, 5, 10). Fig. 3(a) and 3(b) depict the results obtained using the ResNet-18 and ResNet-34 models, respectively. It can be observed that as the privacy budget increases, the model accuracy also improves. Notably, our proposed algorithm significantly outperforms the other two methods. Additionally, we noticed that the accuracy of PDP-LG-Fed remains relatively stable across different privacy budgets, exhibiting only a minor fluctuation of approximately 1%-2%. This is because the algorithm only adds noise to the personalization layer, and in the experiments, it fixes the number of personalization layers to 3, which has a limited impact on the overall network. Therefore, its influence on the overall network is limited, resulting in insignificant accuracy variations across different privacy budgets.

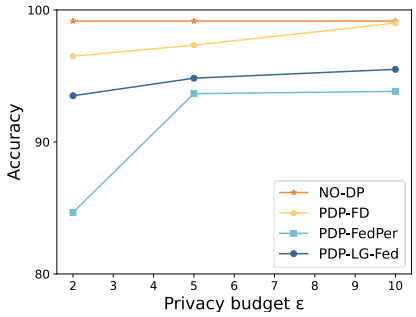
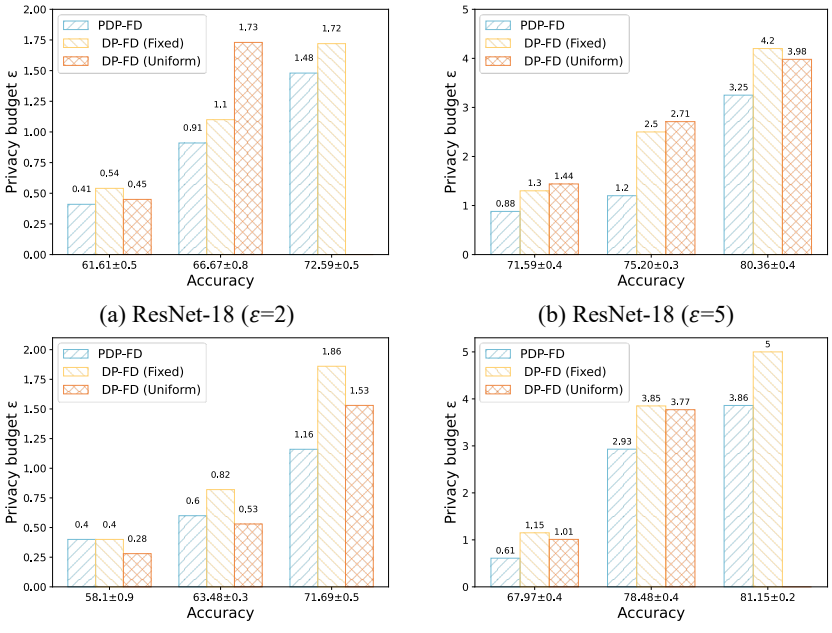


Fig. 4. Test accuracy for different ε in CNN models on the MNIST

In Fig. 4, we compare the accuracy of the CNN model on the MNIST dataset under different privacy budgets. When DP is not used, the accuracy of our algorithm is lower than the other two algorithms. However, after adding noise, the accuracy of our algorithm does not exhibit a significant decrease compared to the other two algorithms. As the privacy budget increases, we observe that the accuracy of PDP-FD gradually approaches that of NO-DP.

Fig. 5 illustrates a comparison of privacy costs between our proposed personalized privacy budget allocation strategy and the average allocation strategy (Fixed), as well as the monotonically decreasing allocation strategy (Uniform). To observe the privacy costs, we pre-specify the accuracy and compare their overall privacy costs at the end of training. In Fig. 5(a) and 5(b), the blue bars represent the privacy cost of our method, while the yellow and orange bars represent the privacy costs of Fixed and Uniform methods, respectively. We can observe that our method always has a lower privacy cost when achieving the same accuracy. However, in Fig. 5(c), initially Uniform has the lowest privacy cost as its privacy budget decreases gradually, resulting in faster convergence during the early stages of training. When the accuracy reached 71.69±0.5, our method demonstrated the minimum privacy cost. Conversely, the Uniform approach exhibits slower convergence in the later stages due to decreasing privacy budgets, resulting in the accumulation of higher privacy costs when reaching the target accuracy. In Fig. 5(d), our method also consistently has a lower privacy cost. It should be noted that the last set of bars in Fig. 5(a) and 5(d) does not include the results for the Uniform method, as it did not achieve the corresponding accuracy.



(a) ResNet-18 (ε=2) (b) ResNet-18 (ε=5)

(c) ResNet-34 (ε=2) (d) ResNet-34 (ε=5)

Fig. 5. Privacy budget consumption of ResNet-18 and ResNet-34 on CIFAR-10

6 Conclusion

In this paper, we propose a Federated Knowledge Distillation Based on Personalized Differential Privacy framework, PDP-FD. By dynamically adjusting the personalization

layer and the allocation strategy of the personalized privacy budget, this algorithm provides users with high-performance personalized models while simultaneously ensuring personalized privacy protection. Extensive experiments demonstrate that our algorithm is more effective than other algorithms and accumulates lower privacy costs. In future work, we will explore further adjustments to the personalization layer of the model to accommodate different client preferences.

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. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-iid data: A survey. Neurocomputing 465, 371–390 (2021)
3. Wu, J., Si, S., Wang, J., Xiao, J.: Threats and defenses of federated learning: a survey. Big Data Research 8(5), 12 (2022)
4. 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)
5. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1322–1333(2015)
6. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: User-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE conference on computer communications. pp. 2512–2520. IEEE (2019)
7. Yan, X., Cui, B., Xu, Y., Shi, P., Wang, Z.: A method of information protection for collaborative deep learning under gan model attack. IEEE/ACM Transactions on Computational Biology and Bioinformatics 18(3), 871–881 (2019)
8. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Advances in neural information processing systems 32 (2019)
9. T Dinh, C., Tran, N., Nguyen, J.: Personalized federated learning with moreau envelopes. Advances in Neural Information Processing Systems 33, 21394–21405(2020)
10. 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)
11. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)
12. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., Liu, Y.: {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In: 2020 USENIX annual technical conference (USENIX ATC 20). pp. 493–506 (2020)
13. Hao, M., Li, H., Luo, X., Xu, G., Yang, H., Liu, S.: Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Transactions on Industrial Informatics 16(10), 6532–6542 (2019)
14. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016
15. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)

16. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189 (2019)
17. Bagdasaryan, E., Poursaeed, O., Shmatikov, V.: Differential privacy has disparate impact on model accuracy. Advances in neural information processing systems 32 (2019)
18. Liang, P.P., Liu, T., Ziyin, L., Allen, N.B., Auerbach, R.P., Brent, D., Salakhutdinov, R., Morency, L.P.: Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523 (2020)
19. Shen, Y., Zhou, Y., Yu, L.: Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10041–10050 (2022)
20. 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)
21. 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)
22. Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M.: Personalized federated learning with first order model optimization, arXiv preprint arXiv:2012.08565 (2020)
23. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557 (2017)
24. Huang, X., Ding, Y., Jiang, Z.L., Qi, S., Wang, X., Liao, Q.: Dp-fl: a novel differentially private federated learning framework for the unbalanced data. World Wide Web 23, 2529–2545 (2020)
25. Dong, F., Ge, X., Li, Q., Zhang, J., Shen, D., Liu, S., Liu, X., Li, G., Wu, F., Luo, J.: Padp-fedmeta: A personalized and adaptive differentially private federated meta learning mechanism for aiot. Journal of Systems Architecture 134, 102754 (2023)
26. Yang, G., Wang, S., Wang, H.: Federated learning with personalized local differential privacy. In: 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS). pp. 484–489. IEEE (2021)
27. Shen, X., Jiang, H., Chen, Y., Wang, B., Gao, L.: Pldp-fl: Federated learning with personalized local differential privacy. Entropy 25(3), 485 (2023)
28. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
29. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
30. Zhang, P.: Research on Rényi Differential Privacy Protection Algorithm in Deep Learning. Master's thesis, Dalian Maritime University (2022) (in Chinese)
31. Zheng, M., Zhang, X., Ma, X., et al.: Unsupervised domain adaptation with differentially private gradient projection. International Journal of Intelligent Systems 2023 (2023)
32. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems 2, 429–450 (2020)