FedCPD: Personalized Federated Learning with Prototype-Enhanced Representation and Memory Distillation

Kaili Jin[1] , **Li Xu**[1] , **Xiaoding Wang**[1*] , **Sun-Yuan Hsieh**[2] , **Jie Wu**[3,4] and **Limei Lin**[1*]
[1]College of Computer and Cyber Security, Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350117, China
[2]Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan
[3]China Telecom Cloud Computing Research Institute, Beijing 100088, China
[4]Department of Computer and Information Sciences, Temple University, PA 19122, USA
kailijin452@gmail.com, {xuli, wangdin1982}@fjnu.edu.cn, hsiehsy@mail.ncku.edu.tw,
jiewu@temple.edu, linlimei@fjnu.edu.cn

Abstract

Federated learning, as a distributed learning framework, aims to develop a global model while preserving client privacy. However, heterogeneity of client data leads to fairness issues and reduced performance. Techniques like parameter decoupling and prototype learning appear promising, yet challenges such as forgetting historical data and limited generalization persist. These methods also lack local insights, with locally trained features prone to overfitting, which affects generalization in global parameter aggregation. To address these challenges, we propose FedCPD, a personalized federated learning framework. FedCPD maintains historical information, reduces information loss, and increases personalization through hierarchical feature distillation and cross-layer feature fusion. Moreover, we utilize representation techniques like prototype contrastive learning and prototype alignment to capture diverse client data features, thus improving model generalization and fairness. Experiments show FedCPD outperforms state-of-the-art models, enhancing generalization by up to 10.40% and personalization by up to 4.90%, highlighting its effectiveness and superiority.

1 Introduction

Federated Learning (FL) [McMahan *et al.*, 2017] is a decentralized machine learning strategy that improves data privacy by enabling local model training on various devices with updates communicated to a central server. However, training a single global model becomes inefficient when the data is heterogeneous or the devices handle distinct tasks, as traditional FL approaches such as FedAvg [McMahan *et al.*, 2017] and FedProx [Li *et al.*, 2020] inadequately address personalized demands. Client data heterogeneity poses a fairness issue, a major challenge for federated learning in diverse settings. Personalized Federated Learning (PFL), [Tan *et al.*, 2022a] addresses these issues by allowing client-specific models that leverage global insights to enhance local outcomes. The main challenge in PFL lies in balancing global knowledge sharing with preserving client-specific information, making the tradeoff an important research area.

In personalized federated learning, historical information forgetting proves challenging as it causes local test accuracy to drop post-global model update. This happens because global model optimization prioritizes overall performance, often overlooking the requirements of individual clients. Consequently, personalized local knowledge diminishes over time. Commonly, personalized federated learning (PFL) employs parameter decoupling to tackle this issue. Techniques like FedPer [Arivazhagan *et al.*, 2019], FedRep [Collins *et al.*, 2021], and FedGH [Yi *et al.*, 2023] separate local private parameters from global ones, allowing clients to tailor task-specific representations using local data. While a shared backbone network (feature extractor) exists, the classifier (head) is customized for local tasks. Despite partially mitigating historical information forgetting, this strategy's efficacy in extracting local features declines when client data exhibits substantial heterogeneity. The core challenge is that task-specific knowledge preservation mechanisms are lacking, making local knowledge loss during global optimization almost unavoidable. Additionally, global optimization objectives might gradually neglect some client-specific tasks, curtailing local model test accuracy improvements.

Personalized federated learning contends with limited generalization performance alongside historical information forgetting. To boost generalization, earlier approaches used prototype methods in representation learning. Prototype learning involves shared class prototypes to curb overfitting and enhance generalization. FedProto [Tan *et al.*, 2022b] enhances generalization by aligning prototypes through measuring distances between similar label representations. FedProc [Mu *et al.*, 2023] and FedCRL [Huang *et al.*, 2024] utilize contrastive loss to connect local features with global prototypes, minimizing representation variations. Yet, as most methods

[*]Corresponding author

tweak FedAvg's basic averaging for data aggregation, they struggle with diverse data and fall short in representation learning. Thus, current methods inadequately merge prototype alignment with contrastive techniques, limiting their generalization efficacy on varied data.

Both challenges pertain to optimizing personalized federated learning, yet their solutions don't cross-apply. Parameter decoupling protects local knowledge to prevent forgetting but falls short on sharing global insights, thus struggling with generalization. On the other hand, prototype learning curbs overfitting and boosts generalization by sharing class prototypes, yet it misses retaining clients' historical task knowledge.

In this paper, we introduce a Federated Class Prototype and Feature Distillation (FedCPD) framework that tackles the challenges of historical information loss and restricted generalization in personalized federated learning. Key contributions are as follows.

Advanced Cross-Layer Fusion and Prototype Alignment for Optimal Historical Conservation and Improved Generalization. In preliminary experiments, we identified historical information forgetting and limitations in parameter decoupling. To address these, we developed techniques like cross-layer feature fusion, hierarchical feature distillation, and attention mechanisms to reduce information loss from global model aggregation and improve personalized model accuracy. We also explored global prototypes for better model generalization and proposed a prototype contrast and alignment strategy to optimize global-local prototype relationships, mitigate overfitting from local data scarcity, enhance cross-label knowledge sharing, and boost generalization in data-limited environments.

Comprehensive Theoretical Analysis Uncovers Key Mechanisms Enhancing Model Performance. We rigorously prove the upper bound of FedCPD algorithm convergence, highlighting two findings. (1) Feature distillation mitigates historical information forgetting from global model aggregation bias by retaining key local features across training, enhancing stability and speed. (2) Aligning global and local prototypes enhances generalization in data-scarce environments, reducing overfitting, aiding cross-label knowledge sharing, accelerating convergence, and improving adaptability to new data.

Empirical Validation Shows Outstanding Performance Across Various Real-World Datasets. We conducted extensive experiments on multiple real-world datasets with varying data heterogeneity. The results showed that, compared to traditional methods, FedCPD significantly improved generalization ability by up to 10.40% and maintained personalization accuracy with an improvement of up to 4.90%, validating the effectiveness and superiority of our approach.

2 Related Work

2.1 Personalized Federated Learning

In personalized federated learning, various approaches have been developed to address data heterogeneity between clients. Parameter decoupling techniques improve model personalization by isolating specific parameters from local modules. For example, FedRep [Collins *et al.*, 2021] alternates between training the global extractor and the local classifier following FedPer's [Arivazhagan *et al.*, 2019] separation strategy. Prototype learning methods mitigate data heterogeneity by computing the average feature representation for each category, efficiently utilizing sparse sample information for classification tasks. These strategies hold significant promise in scenarios with limited learning samples. For example, FedTGP [Zhang *et al.*, 2024a] uses an adaptive margin contrastive learning technique on the server to greatly enhance the representational capabilities of feature vectors. Meanwhile, FedKTL [Zhang *et al.*, 2024b] creates prototype image vector pairs aligned with client tasks through a server-based pretrained generator, thus supporting the learning of the client model and improving performance. These approaches effectively handle heterogeneous data with shared representations, but still offer room for advances through better utilization of these representations. We merge parameter decoupling with prototype learning in the personalized federated learning framework, efficiently tackling the challenges of data heterogeneity between clients and enhancing the model's generalization.

2.2 Feature Distillation

Knowledge Distillation (KD) is a model compression technique that transfers knowledge from well-trained large models to simpler, smaller models, making them suitable for deployment on various devices. Proposed by Hinton et al. [Hinton, 2015], KD reduces model complexity while maintaining performance. FitNets [Adriana *et al.*, 2015] uses intermediate teacher features to guide the student model, ensuring similar predictions. FEED [Park and Kwak, 2019] facilitates knowledge transfer through distillation of feature map-level features through non-linear transformations. Heo et al. [Heo *et al.*, 2019] improve the synergy between teacher and student models with methods such as feature localization and distance functions. DKKR [Chen *et al.*, 2021] uses multilevel feature distillation to guide the student layer by layer. Our method introduces feature distillation into federated learning, combined with attention mechanisms and feature fusion, to facilitate efficient knowledge transfer between teacher and student models.

3 Methodology

In this section, we dive into the details of the implementation of the proposed FedCPD, as illustrated in Figure 1 (global structure diagram) and Algorithm 1.

3.1 Problem Statement

In PFL, an architecture consists of N clients and one server, where each client i has non-IID private data D_i. Clients collaboratively train their personalized models $W_1, ..., W_N$. Similarly to FedPer and FedRep, the backbone network is split into a feature extractor f and a classifier g, where $f : \mathbb{R}^D \to \mathbb{R}^K$ maps the input samples into a feature space and $g : \mathbb{R}^K \to \mathbb{R}^C$ transforms the feature vectors into the label

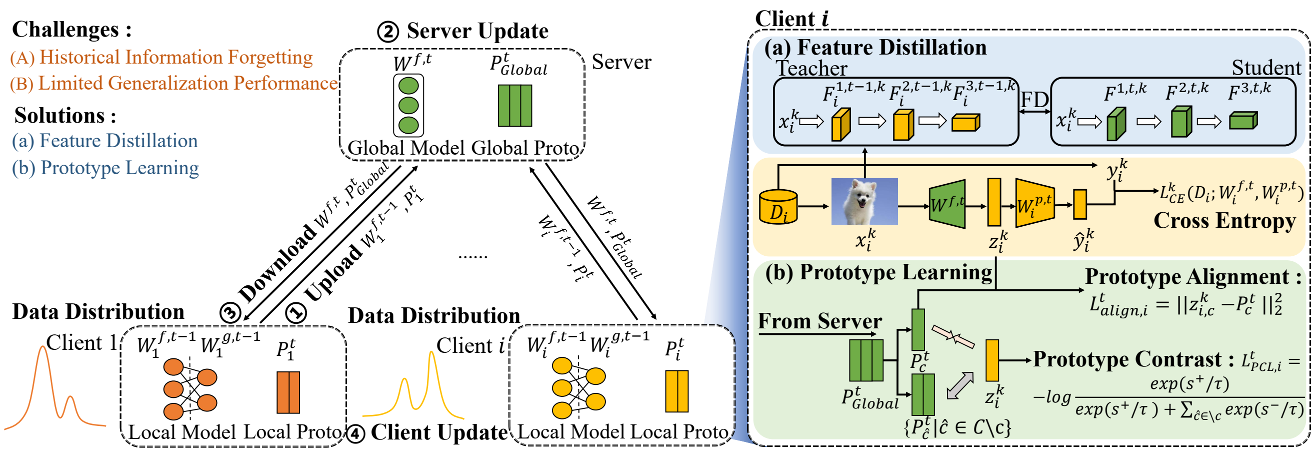
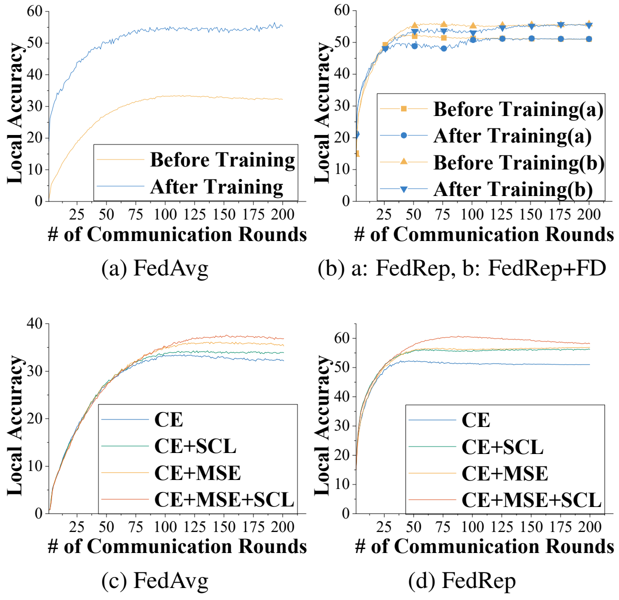

Challenges :
(A) Historical Information Forgetting
(B) Limited Generalization Performance
Solutions :
(a) Feature Distillation
(b) Prototype Learning
② Server Update
Server
$W^{f,t}$
P^t_{Global}
Global Model
Global Proto
③ Download $W^{f,t}$, P^t_{Global}
① Upload $W_1^{f,t-1}$, P_1^t
$W_i^{f,t-1}$, P_i^t
$W^{f,t}$, P^t_{Global}
......
Data Distribution
Client 1
$W_1^{f,t-1}$ $W_1^{g,t-1}$ P_1^t
Local Model
Local Proto
④ Client Update
Data Distribution
Client i
$W_i^{f,t-1}$ $W_i^{g,t-1}$ P_i^t
Local Model
Local Proto
Client i
(a) Feature Distillation
Teacher
Student
$F_i^{1,t-1,k}$ $F_i^{2,t-1,k}$ $F_i^{3,t-1,k}$
FD
$F^{1,t,k}$ $F^{2,t,k}$ $F^{3,t,k}$
x_i^k
D_i
$W^{f,t}$
$W_i^{p,t}$
z_i^k
$\hat{y}_i^k$
y_i^k
$L^k_{CE}(D_i; W_i^{f,t}, W_i^{p,t})$
Cross Entropy
(b) Prototype Learning
Prototype Alignment :
$L^t_{align,i} = ||z^k_{i,c} - P^t_c||^2_2$
From Server
P^t_{Global}
P^t_c
$\{P^t_{\hat{c}} | \hat{c} \in C \backslash c\}$
z_i^k
Prototype Contrast : $L^t_{PCL,i} = -log \frac{exp(s^+/\tau)}{exp(s^+/\tau) + \sum_{\hat{c}\in\backslash c} exp(s^-/\tau)}$

Figure 1: Architecture Illustration of the FedCPD. ① Clients upload their local feature extractors and prototype representations. ② The server aggregates these to create global feature extractors and prototypes. ③ The server sends the global extractors and prototypes back to the clients. ④ Clients initialize their local extractors with the global extractors. FedCPD tackles the challenge of historical information forgetting and limited generalization performance by using **(a) feature distillation** to preserve past information and reduce forgetting, and **(b) prototype learning** to identify varied features and strengthen class distinctions, which in turn boosts the model's ability to generalize.



(a) FedAvg

(b) a: FedRep, b: FedRep+FD

(c) FedAvg

(d) FedRep

Figure 2: The performance gap between the received global model (yellow) and the previous round's locally trained model (blue) is shown in the top two images above. The comparison of accuracy between FedAvg and FedRep under different loss function combinations is illustrated in the bottom two images below. The above algorithms were applied to the CIFAR-100 dataset with 20 non-IID clients ($\beta = 0.1$) and evaluated for local test accuracy.

space. For each client i, the local model $W_i = [W_i^f; W_i^g]$ is trained on the data set D_i. For each sample-label pair $(x_i, y_i) \in D_i$, the model predicts $\hat{y}_i = g(f(x_i; W_i^f); W_i^g)$ and the objective is to minimize the empirical risk across all clients as follows.

$$\arg\min_{W_1,\ldots,W_N} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(W),$$

where $\mathcal{L}_i(W) = \mathbb{E}_{(x_i,y_i)\sim D_i} \mathcal{L}(x_i, y_i; W_i^f, W_i^g)$ represents the empirical loss for client i, given by the expected loss across all samples in D_i. We adopt FedRep as our baseline algorithm. In the t-th round of communication, the central server sends the global extractor $W^{f,t}$ to the active client set $C_t \subseteq C$. Each client $C_i \in C_t$ initializes its parameters $W_{i,0}^{f,t}$ and performs K optimization iterations using its local data, alternating updates between the local extractor and the local classifier in each iteration. The server collects the local extractors $W_i^{f,t}$ and computes the global extractor for the next round $W^{f,t}$ by averaging the parameters, repeating the process until convergence.

3.2 Motivation

Global Model Performance Gap and Historical Information Forgetting. In federated learning, clients receive and train on a global model each round. However, this global model, being an aggregate of multiple client models, may not perform well on clients with significantly different data distributions. This often results in a gap in personalized testing accuracy between the global model and previous local models of clients, as shown in Figure 2a, indicating a potential decrease in personalized performance after updating the global model, especially in cases of high data heterogeneity.

Methods such as FedRep [Collins *et al.*, 2021] tackle the challenge of forgetting historical information [Jin *et al.*, 2022] through techniques such as parameter decoupling. However, they still face limitations in fully leveraging historical local knowledge, which can introduce new challenges. For example, while FedRep preserves the local personalized head with each update, the shared feature extractor may still be susceptible to influence from the global model. This can result in an inadequate representation of local data nuances,

Algorithm 1 FedCPD

```
Require: Total communication rounds $T$, $\{D_i\}_{i=1}^N$, global
    feature extractor parameters $W^{f,0}$, learning rate $\eta$.
Ensure: $\{W_i^{f,*}\}_{i=1}^N$.
for iteration $t = 1, \ldots, T$ do
    Server aggregates $\{W_i^{f,t}\}_{i=1}^N$ and $\{P_i^t\}_{i=1}^N$ to obtain
    $W^{f,t}$ and $P^t$;
    Server sends $W^{f,t}$ and $P^t$ to all clients;
    for the $i$-$th$ client in parallel do
        Fix $W_i^{f,t}$, update $W_i^{g,t}$;          ▷ Classifier Update
        for layer $l = 1, \ldots, L$ do
            Calculate $\mathcal{L}_{fd,i}^{t+1} = \|A_i^{l,*,t+1} - A_i^{l,t}\|_2^2 +$
                $\|F_i^{l,*,t+1} - F_i^{l,t}\|_2^2$;
        end for
        Calculate $\mathcal{L}_{pcl,i}^{t+1} = \mathbb{E}_{(x_{i,j},y_{i,j})\sim D_i} - log$
            $\frac{exp(s(p_{i,j}^{t+1,+})/\tau)}{exp(s(p_{i,j}^{t+1,+})/\tau)+\sum_{\hat{c}\in C\setminus c} exp(s(p_{i,j}^{t+1,-})/\tau)}$;
        Calculate $\mathcal{L}_{align,i}^{t+1} = \|f(x_i) - P_c^t\|_2^2$;
        Calculate $\mathcal{L}_i^{t+1} = \mathcal{L}_{ce,i}^{t+1} + \alpha\mathcal{L}_{align,i}^{t+1} + \beta\mathcal{L}_{pcl,i}^{t+1} +$
        $\gamma\mathcal{L}_{fd,i}^{t+1}$;
        Train $W_i^{g,t+1} \leftarrow W_i^{g,t} - \eta\nabla_{W_i^{g,t}}\mathcal{L}_i^{t+1}$;
        Fix $W_i^{g,t}$,update $W_i^{f,t}$;
        Same as Classifier Update;
        Train $W_i^{f,t+1} \leftarrow W^{f,t+1} - \eta\nabla_{W^{f,t+1}}\mathcal{L}_i^{t+1}$;
        Collect and upload $W_i^{f,t+1}$ and $P_i^{t+1}$;
    end for
end for
return $\{W_i^{f,*}\}_{i=1}^N$.
```

ultimately restricting further enhancements in personalized model performance. As illustrated in Figure 2b, the performance of the model after training is inferior to that of FedAvg. By introducing the hierarchical feature distillation module in FedRep, we verify that the upper limit of local model accuracy can be improved to be comparable with that of local model in FedAvg after making full use of historical knowledge.

Prototype Alignment and Contrastive Learning for Improved Generalization. In the context of federated learning, the data between clients is typically non-IID. However, the success of centralized deep learning indicates that data share global feature representations, with statistical heterogeneity reflected primarily in labels. Due to the limited amount of data available to each client, locally trained features are prone to overfitting, leading to insufficient generalization. Therefore, sharing common prototypes to leverage data from other clients is an effective strategy. Inspired by this, we introduced prototype alignment in our FedAvg and FedRep experiments to ensure global prototype consistency across classes, improving generalization. We also applied prototype contrastive learning to enable cross-label collaboration. In FedAvg, the prototype alignment significantly increases accuracy, outperforming both the combination of alignment and contrastive losses, and contrastive loss alone (see Figure 2c). This suggests that prototype alignment enhances representation learning and reduces overfitting due to limited data.

In FedRep, combining prototype contrast and alignment is even more effective (see Figure 2d), improving class boundary understanding and classification accuracy. The difference arises from the methods used: FedAvg relies on prototype consistency and a simple averaging aggregation, where alignment reduces parameter diversity and stabilizes the global model. In contrast, FedRep focuses on representation learning and uses contrastive learning to capture subtle sample differences, allowing the combination of both methods to better leverage class structure, significantly boosting performance.

3.3 Attention-Guided Hierarchical Feature Distillation

To address the limitations of existing methods in overcoming historical information forgetting and its impact on personalized performance (as discussed in Section 3.2), we propose a feature distillation method based on FedRep to reduce the loss of personalized knowledge caused by updates to the feature extractor, thus improving the performance of personalized models. Specifically, we retain the local feature extractor from the previous round to fully leverage historical personalized knowledge, where $W_i^{f,t}$ serves as the teacher model and $W^{f,t+1}$ serves as the student model. Through hierarchical feature distillation, we transfer knowledge from previous models to the current model and guide the transfer of knowledge by integrating an attention mechanism and a feature fusion mechanism, as shown in Figure 3.

Attention Module. To enhance the effect of feature distillation, we introduce the Convolutional Block Attention Module (CBAM) [Woo *et al.*, 2018], aimed at implementing attention guidance. The goal of feature distillation is to improve the student model's performance through knowledge transfer from the teacher model. With attention mechanisms, we can precisely guide features, enhancing the effectiveness of the distillation process.

The attention map after the CBAM can be represented as:

$$A^l = CBAM(F^l),$$

where l denotes the index of layer, F^l is the feature map of the l-th layer, A^l is the attention map of the l-th layer.

Hierarchical Feature Distillation: In feature distillation, we distill feature maps and attention maps as the primary targets. We use the global backbone $W^{f,t+1}$ of the current round as the student model and the final local backbone $W_i^{f,t}$ of the previous round as the teacher model. Feature distillation typically relies on same-layer feature alignment techniques, which directly compare and align features from the same layers of the student and teacher models.

However, traditional intra-layer feature alignment methods fail to effectively leverage the advantages of shallow features in small object detection and fully integrate the semantic information captured by high-level features due to the lack of interlayer feature reuse. This limitation results in poor distillation performance, especially when dealing with visual tasks that involve complex semantic structures.

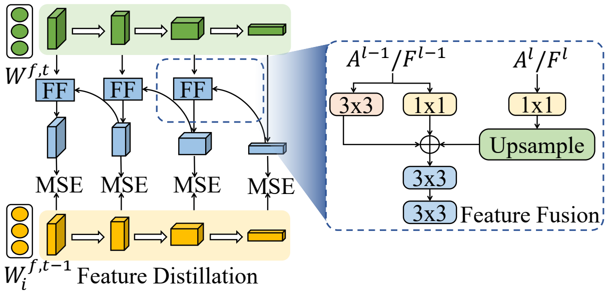


Figure 3: **Framework of Feature Distillation.** The global backbone is used as the student model, and the local backbone from the previous round acts as the teacher model. Cross-layer connections enable feature fusion across layers. We introduce three convolution layers: one for unifying channels (1x1), one for smoothing fusion (3x3), and one for enhancing low-level features (3x3). MSE loss is used for alignment after multi-layer fusion.

Thus, motivated by the cross-layer feature alignment methods [Chen *et al.*, 2021], we constructed cross-layer connection pathways amongst student networks, depicted in Figure 3. Employing a cross-layer feature fusion approach allows us to effectively merge student features from various layers. This strategy not only boosts the target detection abilities of shallow features but also amplifies the semantic expression of higher-level features, thereby facilitating a more thorough transfer of the teacher model's knowledge.

Firstly, we design three types of convolutional layers Cv_1, Cv_2, Cv_3 for each input feature layer. Cv_1 is a transverse convolution that unifies the channel counts of input feature maps of different scales to a fixed output channel count, Cv_2 is used for secondary smoothing after feature fusion, and Cv_3 is used to enhance low-level features. The feature fusion steps can be represented as follows.

$$A^{l,*} = Cv_2(upsamp(Cv_1(A^{l+1})) + Cv_1(A^l) + Cv_3(A^l)),$$

where $upsamp$ represents upscaling using bilinear interpolation, and $A^{l,*}$ represents the final fusion result. We used the highest-layer feature map for the first transverse convolution, performed feature fusion from the bottom up, and before fusion, further processed each layer's original input feature map through an enhanced low-level convolution layer to enhance low-level features. Combining the high-level upsampled features, transverse convolution features, and enhanced low-level features, we then apply Cv_2 to smooth. The final feature alignment process can be represented as follows.

$$\mathcal{L}_{fd} = \|A_s^{l,*} - A_t^l\|_2^2 + \|F_s^{l,*} - F_t^l\|_2^2,$$

where s denotes the student, t denotes the teacher, and we use MSE loss for feature alignment.

3.4 Prototype Contrast and Alignment

To effectively improve the generalizability of the model, as mentioned in Section 3.2, we adopted the method of prototype contrast and alignment. The core objective of this approach is to enhance the model's generalization ability while reducing the decline in personalized performance caused by data heterogeneity. Specifically, we dynamically adjust the relationship between global prototypes and local prototypes, ensuring the retention of personalized features while improving the model's generalization ability.

Prototype Alignment. Initially, we generate feature embeddings on the local datasets $D_i = (x, y)^{N_i}$ of client i using the backbone network $W^{f,t+1}$ distributed by the server as follows.

$$z_i = \{f(W_i^{f,t+1}, D_i)\}^{N_i} \in \mathbb{R}^{K \times N_i},$$

where N_i indicates the number of images at client i. Next, based on the global prototypes P^t received from the server, we align the output embeddings z_i according to

$$\mathcal{L}_{align,i}^{t+1}(z_{i,j}, P_c^t) = \mathbb{E}_{(x_{i,j}, y_{i,j}) \sim D_i^{t+1}} \|z_{i,j} - P_c^t\|_2^2$$
$$i \in \{1, ..., N\}, j \in \{1, ..., N_i\}.$$

Here, $z_{i,j}$ is labeled c, j serves as an index for the local data, and P_c^t represents the global prototype for class c. The objective $\mathcal{L}_{align}$ aims to minimize the squared Euclidean distance between features and class prototypes. The loss of prototype alignment acts as a harmonizing mechanism, reducing feature representation discrepancies between different models or domains, making feature representations more consistent and stable when learning class boundaries.

Prototype Contrast. With the help of prototype alignment loss, contrastive loss optimizes the relative relations in the feature space without conflicting with supervised loss. It enhances the model's learning of robust feature representations, intra-class compactness, and inter-class separability. Specifically, we learn representations through contrasting positive and negative pairs [Huang *et al.*, 2024], which helps in personalization by leveraging positives and negatives between local and global representations.

For each feature representation $z_{i,j}$ with class label c, we consider the global prototype P^t of class c as the positive sample, while prototypes from other classes are considered negative samples. Therefore, for each feature representation, one positive sample pair $p_i^{t+1,+}$ and $|C| - 1$ negative sample pairs $p_i^{t+1,-}$ can be constructed.

$$p_{i,j}^{t+1,+} = (z_{i,j}, P_c^t), \{p_{i,j}^{t+1,-}\} = \{(z_{i,j}, P_{\hat{c}}^t) | \hat{c} \sim C \setminus c\},$$

where j serves as an index for the local data, and $P_{\hat{c}}^t$ represents the remaining prototypes that differ from the category of the output embedding $z_{i,j}$.

To rectify the knowledge from local training in each client using global prototypes, we introduce Global Prototype Contrastive Loss $\mathcal{L}_{pcl,i}$. This loss encourages each client's sample to approach its class's global prototype while distancing itself from the other class's prototypes. We define the Global Prototype Contrastive Loss as follows.

$$\mathcal{L}_{pcl,i}^{t+1} = \mathbb{E}_{(x_{i,j}, y_{i,j}) \sim D_i^{t+1}}$$
$$-\, log \frac{exp(s(p_{i,j}^{t+1,+})/\tau)}{exp(s(p_{i,j}^{t+1,+})/\tau) + \sum_{\hat{c} \in C \setminus c} exp(s(p_{i,j}^{t+1,-})/\tau)},$$
$$s(z_{i,j}, P_c^t) = \frac{z_{i,j} \cdot P_c^t}{\|z_{i,j}\|_2 \cdot \|P_c^t\|_2} \in [-1, 1],$$

where τ is a temperature hyperparameter that adjusts the focus on positive and negative samples, and $s(\cdot)$ is the cosine similarity. This multi-constraint learning mechanism combines supervised loss, alignment loss, and contrastive loss, enhancing the model's ability to distinguish different categories, ensuring consistency in feature representations, and optimizing the structure of the feature space, thereby improving generalization and performance.

4 Convergence Bounds Analysis

In this subsection, we present the convergence analysis for FedCPD and outline the assumptions necessary to prove its convergence, following a framework similar to FedProto [Tan *et al.*, 2022b], FedHKD [Chen *et al.*, 2023] and FedCRL [Huang *et al.*, 2024]. Detailed assumptions and proofs are provided in Appendix B.

Theorem 1. *(One-Round Deviation). Let Assumptions 1 to 3 hold (see Appendix B). For an arbitrary client, between the iteration t and the iteration $t+1$, we have*

$$\mathbb{E}[\mathcal{L}^{t+1,1/2}] \le \mathcal{L}^{t,1/2} - (\eta_e - \frac{L_1\eta_e^2}{2}) \sum_{e=1/2}^{E-1} \|\nabla \mathcal{L}_i^{t,e}\|_2^2 + \frac{L_1 E\eta_0^2}{2}\sigma^2 + E\eta_0 G^2 + \frac{L_1}{2}E^2\eta_0^2 G^2 + 2\alpha L_2 E^2\eta_0^2 G^2 + \beta\frac{2}{\tau}.$$

Theorem 1 shows that for any client, adjusting the appropriate hyperparameters can bound the deviation in expected loss for the i-th client from iteration t to $t+1$, thereby ensuring convergence.

Theorem 2. *(Non-Convex FedCPD Convergence). $\lambda\eta_0 < \eta_e < \eta_0$, $e \in \{\frac{1}{2}, 1, 2, \ldots, E\}$, where λ represents the decay factor for the learning rate. If the learning rate for each epoch satisfies the following condition, the loss function decreases monotonically, leading to convergence*

$$\lambda\eta_0 < \eta_e < (B + \sqrt{B^2 - 4AC})/2A,$$

where $S = \sum_{e=1/2}^{E-1} \|\nabla \mathcal{L}_i^{t,e}\|_2^2, B = S - EG^2, A = (L_1(E\sigma^2 + E^2G^2 + S)/2 + 2\alpha L_2 E^2 G^2), C = 2\beta/\tau.$

Theorem 2 indicates that, under the specified learning rate conditions, the loss function for any client decreases consistently between successive communication rounds, guaranteeing algorithm convergence.

5 Experiments and Discussion

5.1 Experiment Setup

We evaluated FedCPD using three popular image classification datasets (CIFAR-10, FMNIST, and CIFAR-100). We compare it with isolated local training and eight popular federated learning methods, including these state-of-the-art algorithms FedProto [Tan *et al.*, 2022b], FedGH [Yi *et al.*, 2023], FedALA [Zhang *et al.*, 2023], and FedPA [Jiang *et al.*, 2024]. The experiments run for 200 communication rounds, each round involving 1 epoch of local training and a batch size of 10. All models are trained using the SGD optimizer with a learning rate of 0.01. The model consists of two convolutional layers and two fully connected layers. To simulate data heterogeneity, we use two settings: the "practical setting" based on the Dirichlet distribution and the "pathological setting" with varied class samples from the datasets.

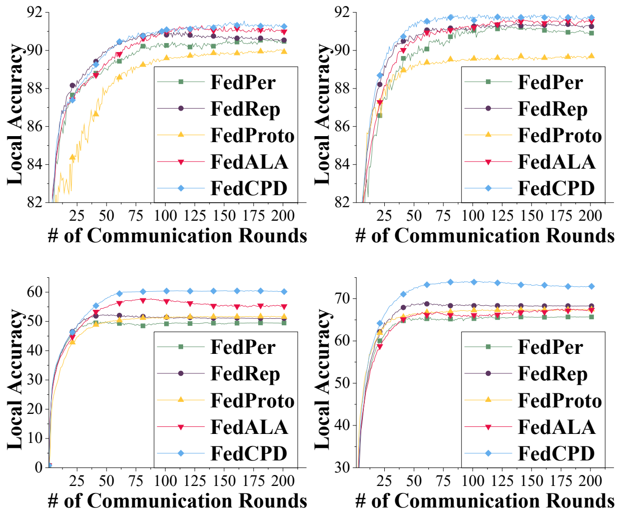


Figure 4: Test accuracy curves across CIFAR-10 and CIFAR-100 datasets under two levels of data heterogeneity ($\beta = 0.1$ and pat).

5.2 Performance Comparisons

By default, local clients are configured with $\beta = 0.1$ and $N = 20$.

Test Accuracy. To evaluate FedCPD, we benchmarked it against leading federated learning methods on three datasets, each subjected to two highly heterogeneous partitions (see Table 1). Accuracy is reported for every client's own test set, so the metric directly reflects the personalization performance. Figure 4 trace the evolution of the test accuracy during training. Across both realistic and pathological splits, FedCPD consistently delivers the highest accuracy, outperforming all baselines, and confirming its superior effectiveness.

Results Analysis. FedCPD converges significantly faster than other algorithms, especially in the early stages (e.g., the first 50 rounds), where the improvement in test accuracy is most notable.Moreover, as the number of label categories increases in both heterogeneous settings, FedCPD shows more significant improvements over the second-best method: CIFAR-100 (4.90%/4.58%), CIFAR-10 (0.76%/0.10%), and FMNIST (0.04%/0.15%). This shows that FedCPD can capture category differences more effectively, leading to a significant improvement in classification accuracy.

Generalization. To evaluate the performance of FedCPD under different levels of heterogeneity, we adjusted the value of β of the Dirichlet distribution in the CIFAR-10 data set to control actual heterogeneity and modified the number of label categories held by each client in the CIFAR-100 data set to control pathological heterogeneity (see Table 2). The results show that FedCPD performed best in both heterogeneity settings.

Method	Practical heterogeneous ($\beta = 0.1, N = 20$)						Pathological heterogeneous ($N = 20$)					
	FMNIST		CIFAR10		CIFAR100		FMNIST		CIFAR10		CIFAR100	
	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.
Local	97.21	5.57	88.91	11.39	47.77	4.93	93.78	3.02	88.74	5.52	64.13	5.96
FedAvg	85.86	11.87	57.24	12.48	32.35	3.72	87.19	3.25	57.34	11.39	27.10	4.72
FedProx	85.81	11.87	57.17	12.72	32.58	3.80	87.34	3.41	57.36	12.27	27.23	4.80
FedPer	97.64	5.11	90.48	9.45	49.45	4.45	95.27	2.30	90.93	4.82	65.68	4.24
FedRep	97.65	4.42	90.56	9.51	51.01	4.15	95.31	2.04	91.27	4.23	68.27	4.25
FedGH	96.32	11.46	83.06	20.89	49.08	5.06	93.74	3.10	88.80	5.45	65.49	5.71
FedPA	97.20	5.60	90.23	9.89	52.36	4.73	93.76	3.87	90.73	5.23	67.59	5.47
FedProto	97.43	5.77	89.99	10.29	51.63	4.80	93.94	3.34	89.65	5.78	67.41	5.51
FedALA	97.79	4.30	91.02	9.04	55.33	4.03	95.58	2.14	91.57	4.17	67.31	3.66
FedCPD	**97.83**	4.83	**91.78**	**7.87**	**60.23**	3.99	**95.73**	2.19	**91.67**	**3.89**	**72.85**	**2.78**

Table 1: The average test accuracy of the 3 datasets in the real-world environment and the 3 datasets in the pathological heterogeneous environment, as well as the average standard deviation of the accuracy across all clients.

Results Analysis. As shown in Table 2, in moderate and low heterogeneity environments, FedCPD still outperforms other personalized federated learning methods: CIFAR-10 (1.68%/2.11%), CIFAR-100 (7.55%/10.40%), which demonstrates its strong generalizability. Unlike other algorithms that rely on local data, FedCPD effectively integrates both global and local information, significantly improving model performance in scenarios with many label categories and uneven data distributions, thus improving its adaptability across different data distribution scenarios.

Fairness. We evaluated FedCPD's fairness using the standard deviations in Tables 1 and 2. In all experiments, FedCPD was ranked in the top three, showing minimal performance disparities between client groups and achieving high fairness.

Results Analysis. FedCPD achieves excellent fairness by effectively balancing global and local models, reducing performance disparities among clients. This improves the accuracy of classification and ensures consistency between different clients.

5.3 Discussion

Experimental findings and theoretical insights confirm that feature distillation efficiently reduces the forgetting of historical data by conveying knowledge throughout training iterations, leading to smoother optimization and faster early-stage convergence (see Figure 4). Retaining historical data significantly improves the global model's learning performance, raising classification accuracy by 4.66% under typical conditions (refer to Appendix A) and outperforming leading parameter decoupling techniques by approximately 4.90% across different label distributions (refer to Table 1). Moreover, feature distillation enhances client equity, promoting balanced model training and consistently securing a position among the top three for global fairness.

The global prototype enhances the connection between global and local models via prototype alignment and contrastive learning, leading to improved class separation, faster convergence and a 7.23% boost in classification accuracy in default settings (see Appendix A), especially with numerous labels and imbalanced data (refer to Table 1). These strategies help the model capture global insights and encourage cross-label knowledge sharing. For generalization, the global prototype combines global knowledge, reducing overfitting due to limited data, and delivers up to a 10.40% boost over the best SOTA in varied environments (refer to Table 2). Furthermore, the global prototype improves fairness by diminishing performance variance among different clients, promoting a balanced training process, particularly with uneven data distributions.

Method	CIFAR10		CIFAR100	
	$\beta = 0.5$	$\beta = 1$	cls./clt.=20	cls./clt.=50
	Acc.	Acc.	Acc.	Acc.
Local	88.91	61.23	48.04	31.06
FedAvg	57.24	70.61	30.17	32.05
FedProx	57.17	70.60	30.13	31.62
FedPer	90.48	69.13	51.62	35.59
FedRep	90.56	70.62	53.65	36.77
FedGH	83.06	62.46	48.04	32.16
FedPA	84.90	72.16	52.74	36.19
FedProto	90.40	63.36	50.78	32.88
FedALA	81.15	77.03	54.99	38.27
FedCPD	**92.24**	**79.14**	**62.54**	**48.67**

Table 2: The test accuracy with changes to the β of CIFAR-10 for the real-world heterogeneity evaluation and the label classes for each client (cls./clt.) in CIFAR-100 for the pathological heterogeneity evaluation.

6 Conclusion and Future Work

To address the issue of historical forgetting and enhance generalization, we introduce FedCPD, a personalized federated learning framework. This framework integrates hierarchical feature distillation, cross-layer feature fusion, and attention mechanisms to effectively balance local and global models while maximizing the utility of historical knowledge. Additionally, shared prototypes are employed to address label skew and accelerate convergence, facilitating privacy-aware and equitable adaptation. Experimental results indicate that FedCPD surpasses leading models, achieving an improvement in generalization by up to 10.40% and personalization by up to 4.90%, thereby demonstrating its efficacy and superiority. Future research will investigate its capability to manage other forms of statistical heterogeneity and evaluate the algorithm's impact on representation learning.

A Ablation Study

To validate the effectiveness of the various components of FedCPD, we performed ablation experiments on the CIFAR-100 data set with $\beta = 0.1$. The experiments aimed to verify the contributions of feature distillation, prototype alignment, and prototype contrast. Accuracy comparisons can be observed in Table 3, which demonstrates the effectiveness of the different components of FedCPD.

Results Analysis. In the absence of these three components, the algorithm is equivalent to FedRep, exhibiting the poorest performance. With the introduction of prototype contrast or prototype alignment, performance improved by approximately 5%, indicating that these strategies effectively improve the class separability of the model and improve classification accuracy. Furthermore, by incorporating local feature distillation, the model can learn more refined local feature representations, further improving classification accuracy. Ultimately, with the combination of all three components, FedCPD achieves optimal performance, conclusively confirming the effectiveness of these components.

B Convergence Theory Derivation

Additional Notation. The introduction of additional variables here is to better represent the local model update process. The embedding function corresponding to the i-th client is denoted by $f_i(W_i^f) : \mathbb{R}^D \to \mathbb{R}^K$, $g_i(W_i^g) : \mathbb{R}^K \to \mathbb{R}^C$ is the decision function for all clients. Here, D, K, and C denote the sizes of the input, feature, and label spaces, respectively. Thus, the labeling function can be written as $F_i(W_i^f, W_i^g) = g_i(W_i^g) \circ f_i(W_i^f)$, and W_i can be used as (W_i^f, W_i^g). Therefore, the local loss function for client i is given by

$$\begin{aligned}
&\mathcal{L}(W_i^f, W_i^g; x, y) = \frac{1}{B_i}\sum_{k=1}^{B_i}\mathcal{L}_{CE}(F_i(W_i^f, W_i^g; x_k), y_k) \\
&+ \alpha\frac{1}{B_i}\sum_{k=1}^{B_i}\|f_i(W_i^{f,t+1}, x_k) - P^{t+1}\|_2^2 \\
&+ \gamma\frac{1}{B_i}\sum_{k=1}^{B_i}[\|A_{i,k}^{l,*,t+1} - A_{i,k}^{l,t}\|_2^2 + \|F_{i,k}^{l,*,t+1} - F_{i,k}^{l,t}\|_2^2] \\
&- \beta\frac{1}{B_i}\sum_{k=1}^{B_i} log\frac{exp(S_1/\tau)}{exp(S_1/\tau) + \sum_{\hat{c}\in C\setminus c} exp(\hat{S}_1)/\tau)},
\end{aligned}$$

where $S_1 = s(f_i(W_i^{f,t+1}, x_k), P_c^{t+1}), \hat{S_1} = s(f_i(W_i^{f,t+1}, x_k), P_{\hat{c}}^{t+1})$.

Let $(x_k, y_k) \sim D_i$ represent the local data points, where D_i denotes the private dataset of client i, and B_i is the training batch size. The loss function $\mathcal{L}$ remains fixed within each communication round, but varies between rounds, adding complexity to the convergence analysis.

	$\mathcal{L}_{fd}$	$\mathcal{L}_{align}$	$\mathcal{L}_{pcl}$	Accuracy
Module	×	×	×	51.01
	✓	×	×	55.67
	×	×	✓	56.20
	×	✓	×	56.83
	×	✓	✓	58.24
	✓	✓	✓	60.23

Table 3: The impact of various components of FedCPD.

For notation, t indicates the communication round and $e \in 1/2, 1, 2, \ldots, E$ refers to the local iterations, where E is the total number of local updates. Thus, $tE + e$ represents the e-th local iteration in the $(t+1)$-th round. Furthermore, tE is the time step just before aggregation on the server, while $tE + 1/2$ marks the step between aggregation and the start of the first local update.

Assumption 1. *We assume that each loss function $\mathcal{L}(W)$ satisfies L_1-Lipschitz smoothness, indicating that its gradient is L_1-Lipschitz continuous, and that the embedding function of the feature extractor $f(\cdot)$ is L_2-Lipschitz continuous.*

$$\begin{aligned}
&\|\nabla\mathcal{L}(W^{t_1}) - \nabla\mathcal{L}(W^{t_2})\|_2 \le L_1\|W^{t_1} - W^{t_2}\|_2, \\
&\|f_i(W_i^{f,t_1}) - f_i(W_i^{f,t_2})\|_2 \le L_2\|W_i^{f,t_1} - W_i^{f,t_2}\|_2, \\
&\forall t_1, t_2 > 0, i \in \{1, 2, ..., N\},
\end{aligned}$$

which implies the following quadratic bound,

$$\begin{aligned}
&\mathcal{L}(W^{t_1}) - \mathcal{L}(W^{t_2}) \le \langle\nabla\mathcal{L}_{t_2}, (W^{t_1} - W^{t_2})\rangle + \\
&\frac{L_1}{2}\|W^{t_1} - W^{t_2}\|_2, \forall t_1, t_2 > 0, i \in \{1, 2, ..., N\},
\end{aligned}$$

Assumption 2. *The stochastic gradient $g_i^t = \nabla\mathcal{L}(W_i^t, \xi_i^t)$ is an unbiased estimator of the local gradient for each client. We assume that its expectation satisfies*

$$\mathbb{E}_{x_i\sim D_i}[g_i^t] = \nabla\mathcal{L}(W_i^t), \forall i \in \{1, 2, ..., N\},$$

the variance is bounded by σ^2 as follows.

$$\mathbb{E}[\|g_i^t - \nabla\mathcal{L}(W_i^t)\|_2^2] \le \sigma^2, \forall i \in \{1, 2, ..., N\}.$$

Assumption 3. *The expected value of Euclidean norm of the stochastic gradient is bounded by G,*

$$\mathbb{E}[\|g_i^t\|_2] \le G, \forall i \in \{1, 2, ..., N\}.$$

Lemma 1. *In the t-th communication round, the loss function for any client i, after conducting E local training rounds, is bounded as*

$$\mathbb{E}[\mathcal{L}_i^{t,E}] - \mathcal{L}_i^{t,1/2} \le -(\eta_e - \frac{L_1\eta_e^2}{2})\sum_{e=1/2}^{E-1}\|\nabla\mathcal{L}_i^{t,e}\|_2^2 + \frac{L_1E\eta_0^2}{2}\sigma^2.$$

Proof.

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_i^{t,1}] \le&\mathbb{E}[\mathcal{L}_i^{t,1/2} + \langle\nabla\mathcal{L}_i^{t,1/2}, (W^{t,1} - W^{t,1/2})\rangle \\
&+ \frac{L_1}{2}\|W^{t,1} - W^{t,1/2}\|_2^2] \\
=&\mathbb{E}[\mathcal{L}_i^{t,1/2} - \eta_e\langle\nabla\mathcal{L}_i^{t,1/2}, g_i^{t,1/2}\rangle + \frac{L_1}{2}\|\eta_e g_i^{t,1/2}\|_2^2] \\
=&\mathcal{L}_i^{t,1/2} - \eta_e\mathbb{E}[\langle\nabla\mathcal{L}_i^{t,1/2}, g_i^{t,1/2}\rangle] + \frac{L_1\eta_e^2}{2}\mathbb{E}[\|g_i^{t,1/2}\|_2^2] \\
=&\mathcal{L}_i^{t,1/2} - \eta_e\|\nabla\mathcal{L}_i^{t,1/2}\|_2^2 + \frac{L_1\eta_e^2}{2}\mathbb{E}[\|g_i^{t,1/2}\|_2^2] \\
\le&\mathcal{L}_i^{t,1/2} - (\eta_e - \frac{L_1\eta_e^2}{2})\|\nabla\mathcal{L}_i^{t,1/2}\|_2^2 + \frac{L_1\eta_e^2}{2}\sigma^2.
\end{aligned}$$

Then, by telescoping of E steps and setting the learning step at the beginning of local training to $\eta_{1/2} = \eta_0$, we have,

$$\mathbb{E}[\mathcal{L}_i^{t,E}] \le \mathcal{L}_i^{t,1/2} - (\eta_e - \frac{L_1\eta_e^2}{2})\sum_{e=1/2}^{E-1}\|\nabla\mathcal{L}_i^{t,e}\|_2^2 + \frac{L_1E\eta_0^2}{2}\sigma^2.$$

Lemma 2. *After completing E local training rounds in the t-th communication round and before starting local training in the $(t+1)$-th communication round, the loss function for any client i remains bounded.*

$$\mathbb{E}[\mathcal{L}_i^{t+1,1/2}] \leq \mathcal{L}_i^{t+1} + E\eta_0 G^2(1 + \frac{L_1}{2}E\eta_0^2 + 2\alpha L_2 E\eta_0) + \beta\frac{2}{\tau}.$$

Proof.

$$\begin{aligned}\mathcal{L}_i^{t+1,1/2} - \mathcal{L}_i^{t,E} =&\mathcal{L}(W_i^{f,t+1,1/2}, P^{t+2}) - \mathcal{L}(W_i^{f,t,E}, P^{t+2})\\ &+ \mathcal{L}(W_i^{f,t,E}, P^{t+2}) - \mathcal{L}(W_i^{f,t,E}, P^{t+1})\\ \leq&\langle \nabla\mathcal{L}_i^{t,E}, W_i^{f,t+1,1/2} - W_i^{f,t,E}\rangle\\ &+ \frac{L_1}{2}\|W_i^{f,t+1,1/2} - W_i^{f,t,E}\|_2^2\\ &+ \mathcal{L}(W_i^{f,t,E}, P^{t+2}) - \mathcal{L}(W_i^{f,t,E}, P^{t+1}).\end{aligned}$$

Applying expectations to both sides of the inequalities

$$\begin{aligned}&\mathbb{E}[\mathcal{L}_i^{t+1,1/2}] - \mathcal{L}_i^{t,E}\\ \leq&E\eta_0 G^2 + \frac{L_1}{2}\mathbb{E}\|\sum_{j=1}^{S_t} p_j W_j^{f,t,E} - (W_i^{f,t,E} - W_i^{f,t,1/2})\|_2^2\\ &+ \frac{L_1}{2}E^2\eta_0^2 G^2 + \mathbb{E}\mathcal{L}(W_i^{f,t,E}, P^{t+2}) - \mathbb{E}\mathcal{L}(W_i^{f,t,E}, P^{t+1})\\ \leq&E\eta_0 G^2 + L_1 E^2\eta_0^2 G^2 + \mathbb{E}\mathcal{L}(W_i^{f,t,E}, P^{t+2}) - \mathbb{E}\mathcal{L}(W_i^{f,t,E}, P^{t+1}).\end{aligned}$$

It should be noted that when the model parameters remain unchanged, the first two terms of the loss function are the same. Therefore, we are left only with the terms related to the prototype representations.

$$\begin{aligned}&\|f_i(W_i^{f,t+1,E}, x_k) - P^{t+2}\|_2^2 - \|f_i(W_i^{f,t+1,E}, x_k) - P^{t+1}\|_2^2\\ \leq&\|P^{t+2} - P^{t+1}\|_2^2\\ \leq&\sum_{i=1}^{N} p_i \frac{1}{N_i^j}\sum_{k=1}^{N_i^j} L_2\|W_i^{t+1,E} - W_i^{t,E}\|_2^2\\ =&L_2\sum_{i=1}^{N} p_i\|W_i^{t+1,E} - W_i^{t,E}\|_2^2.\end{aligned}$$

Taking expectation of both sides,

$$\begin{aligned}&\mathbb{E}[\|f_i(W_i^{f,t+1,E}, x_k) - P^{t+2}\|_2^2 - \|f_i(W_i^{f,t+1,E}, x_k) - P^{t+1}\|_2^2]\\ \leq&L_2\sum_{i=1}^{N} p_i(\mathbb{E}\|W_i^{t+1,E} - W_i^{t+1,1/2}\|_2^2 + \mathbb{E}\|W_i^{t+1,1/2} - W_i^{t,E}\|_2^2)\\ \leq&L_2\sum_{i=1}^{N} p_i(E^2\eta_0^2 G^2 + \mathbb{E}\|\sum_{j=1}^{S_t} W_j^{t,E} - W_i^{t,1/2} + W_i^{t,1/2} - W_i^{t,E}\|_2^2)\\ \leq&L_2\sum_{i=1}^{N} p_i(E^2\eta_0^2 G^2 + \mathbb{E}\|W_i^{t,E} - W_i^{t,1/2}\|_2^2)\\ =&L_2\sum_{i=1}^{N} p_i(E^2\eta_0^2 G^2 + \mathbb{E}\|\sum_{e=1/2}^{E-1}\eta_e g_i\|_2^2) \leq 2L_2 E^2\eta_0^2 G^2.\end{aligned}$$

The proof of the other term is as follows.

$$\begin{aligned}&- log\frac{exp((S_2)/\tau)}{exp((S_2)/\tau) + \sum_{\hat{c}\in C\backslash c} exp((\hat{S_2})/\tau)}\\ &+ log\frac{exp((S_1)/\tau)}{exp((S_1)/\tau) + \sum_{\hat{c}\in C\backslash c} exp((\hat{S_1})/\tau)}\\ \leq&- log\frac{exp(-1/\tau)}{exp(-1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(1/\tau)}\\ &+ log\frac{exp(1/\tau)}{exp(1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(-1/\tau)}\\ =&-(\frac{-1}{\tau} - logD^{t+1,1/2}) + (\frac{1}{\tau} - logD^{t+1})\\ =&\frac{2}{\tau} + (logD^{t+1,1/2} - logD^{t+1}) \leq \frac{2}{\tau},\end{aligned}$$

where

$$\begin{aligned}&logD^{t+1,1/2} - logD^{t+1}\\ =&log(exp(-1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(1/\tau))\\ &- log(exp(1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(-1/\tau))\\ \approx&log((C-1)exp(1/\tau)) - logexp(1/\tau)\\ =&log(C-1),\end{aligned}$$

where $D^{t+1,1/2} = exp(-1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(1/\tau)$, $D^{t+1} = exp(1/\tau) + \sum_{\hat{c}\in C\backslash c} exp(-1/\tau)$. Since $log(C-1)$ is a constant, and when τ is very small, $\frac{2}{\tau}$ is numch greater than $log(C-1)$,we can omit $log(C-1)$.

Summarizing the above derivations, we obtain an upper bound for the loss function.

$$\begin{aligned}\mathbb{E}[\mathcal{L}_i^{t+1,1/2}] \leq&\mathcal{L}_i^{t+1} + E\eta_0 G^2 + \frac{L_1}{2}E^2\eta_0^2 G^2\\ &+ 2\alpha L_2 E^2\eta_0^2 G^2 + \beta\frac{2}{\tau}.\end{aligned}$$

Theorem 1. *(One-round deviation). Assume that Assumptions 1 to 3 are satisfied. For any client, after each communication round, we obtain*

$$\mathbb{E}[\mathcal{L}^{t+1,1/2}] \leq \mathcal{L}^{t,1/2} - (\eta_e - \frac{L_1\eta_e^2}{2})\sum_{e=1/2}^{E-1}\|\nabla\mathcal{L}_i^{t,e}\|_2^2 +$$
$$\frac{L_1 E\eta_0^2}{2}\sigma^2 + E\eta_0 G^2 + \frac{L_1}{2}E^2\eta_0^2 G^2 + 2\alpha L_2 E^2\eta_0^2 G^2 + \beta\frac{2}{\tau}$$

Theorem 2. *(Non-convex FedCPD convergence). $\lambda\eta_0 < \eta_e < \eta_0$, $e \in \{\frac{1}{2}, 1, 2, \dots, E\}$, where λ represents the decay factor for the learning rate. If the learning rate for each epoch satisfies the following condition, the loss function decreases monotonically, leading to convergence:*

$$\lambda\eta_0 < \eta_e < \frac{B + \sqrt{B^2 - 4AC}}{2A}$$

where $S = \sum_{e=1/2}^{E-1}\|\nabla\mathcal{L}_i^{t,e}\|_2^2, B = S - EG^2, A = (\frac{L_1(E\sigma^2+E^2G^2+S)}{2} + 2\alpha L_2 E^2 G^2), C = \frac{2\beta}{\tau}$.

Proof.

$$\begin{aligned}&\frac{L_1 E\eta_e^2}{2}\sigma^2 + E\eta_e G^2 + \frac{L_1}{2}E^2\eta_e^2 G^2 + 2\alpha L_2 E^2\eta_e^2 G^2 + \beta\frac{2}{\tau}\\ &- (\eta_e - \frac{L_1\eta_e^2}{2})\sum_{e=1/2}^{E-1}\|\nabla L_i^{t,e}\|_2^2\\ =&\frac{L_1 E\eta_e^2}{2}\sigma^2 + E\eta_e G^2 + \frac{L_1}{2}E^2\eta_e^2 G^2 + 2\alpha L_2 E^2\eta_e^2 G^2 + \beta\frac{2}{\tau}\\ &- (\eta_e - \frac{L_1\eta_e^2}{2})S\\ =&(\frac{L_1 E\sigma^2}{2} + \frac{L_1 E^2 G^2}{2} + 2\alpha L_2 E^2 G^2 + \frac{L_1 S}{2})\eta_e^2\\ &+ (EG^2 - S)\eta_e + \frac{2\beta}{\tau} \leq 0.\end{aligned}$$

Since $A > 0$, the quadratic function opens upwards. The inequality $A\eta_e^2 + B\eta_e + C \leq 0$ holds between the roots of the quadratic equation. Therefore, the acceptable range for η_e is between the two roots. However, we are interested in the maximum allowable η_e that satisfies the inequality, so we focus on the larger root:

$$\eta_e < \frac{B + \sqrt{B^2 - 4AC}}{2A}, \beta < \frac{\tau B^2}{8A}.$$

Acknowledgments

This work is supported by National Natural Science Foundation of China under grants 62171132, 62471139, and U1905211, and Natural Science Foundation of Fujian Province under grant 2024J09032.

References

[Adriana *et al.*, 2015] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2(3):1, 2015.

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[Chen *et al.*, 2021] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.

[Chen *et al.*, 2023] Huancheng Chen, Haris Vikalo, and Johnny Wang. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. *arXiv preprint arXiv:2301.08968*, 2023.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

[Heo *et al.*, 2019] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1921–1930, 2019.

[Hinton, 2015] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Huang *et al.*, 2024] Chenghao Huang, Xiaolu Chen, Yanru Zhang, and Hao Wang. Fedcrl: Personalized federated learning with contrastive shared representations for label heterogeneity in non-iid data. *arXiv preprint arXiv:2404.17916*, 2024.

[Jiang *et al.*, 2024] Lei Jiang, Xiaoding Wang, Xu Yang, Jiwu Shu, Hui Lin, and Xun Yi. Fedpa: Generator-based heterogeneous federated prototype adversarial learning. *IEEE Transactions on Dependable and Secure Computing*, 22(2):939–949, 2024.

[Jin *et al.*, 2022] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):567–580, 2022.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Mu *et al.*, 2023] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems*, 143:93–104, 2023.

[Park and Kwak, 2019] SeongUk Park and Nojun Kwak. Feed: Feature-level ensemble for knowledge distillation. *arXiv preprint arXiv:1909.10754*, 2019.

[Tan *et al.*, 2022a] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, 2022.

[Tan *et al.*, 2022b] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[Yi *et al.*, 2023] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8686–8696, 2023.

[Zhang *et al.*, 2023] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023.

[Zhang *et al.*, 2024a] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16768–16776, 2024.

[Zhang *et al.*, 2024b] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. An upload-efficient scheme for transferring knowledge from a server-side pre-trained generator to clients in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12109–12119, 2024.