applied sciences

MDPI

Article

Personalized Federated Learning with Progressive Local Training Strategy and Lightweight Classifier

Jianhao Liu [1,2], Wenjuan Gong [1,2,*], Ziyi Fang [1], Jordi Gonzàlez [3] and Joel Rodrigues [4]

[1] Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; z22070026@s.upc.edu.cn (J.L.); 2217020111@s.upc.edu.cn (Z.F.)
[2] Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software, Qingdao 266580, China
[3] Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain; jordi.gonzalez@uab.cat
[4] Higher School of Technology, Amazonas State University, Manaus 69000, Brazil; joeljr@ieee.org
* Correspondence: wenjuangong@upc.edu.cn

Abstract: Data heterogeneity poses a significant challenge in federated learning (FL), which has become a central focus of contemporary research in artificial intelligence. Personalized federated learning (pFL), a specialized branch of FL, seeks to address this issue by tailoring models to the unique data distributions of individual clients. Despite its potential, current pFL frameworks face critical limitations, particularly in handling client training discontinuity. When clients are unable to engage in every training round, the resulting models tend to diverge from their local knowledge, leading to catastrophic forgetting. Moreover, existing frameworks often separate the model from the local classifier used for personalization, keeping the classifier local for extended periods. This inherent characteristic of classifiers frequently leads to overfitting on local training data, thereby impairing the generalization capability of the local models. To tackle these challenges, we propose a novel personalized federated learning framework, PFPS-LWC (Personalized Federated Learning with a Progressive Local Training Strategy and a Lightweight Classifier). Our approach introduces local knowledge recall and employs regularized classifiers to mitigate the effects of local knowledge forgetting and enhance the generalization of the models. We evaluated the performance of PFPS-LWC under varying degrees of data heterogeneity using the CIFAR10 and CIFAR100 datasets. Our method outperformed the state-of-the-art approach by up to 4.22% and consistently achieved the best performance across various heterogeneous environments, further demonstrating its effectiveness and robustness.

Keywords: federatedlearning; personalized federated learning; data heterogeneity; catastrophic forgetting

check for updates

Academic Editor: Douglas O'Shaughnessy

Received: 17 January 2025
Revised: 22 February 2025
Accepted: 24 February 2025
Published: 25 February 2025

Citation: Liu, J.; Gong, W.; Fang, Z.; Gonzàlez, J.; Rodrigues, J. Personalized Federated Learning with Progressive Local Training Strategy and Lightweight Classifier. *Appl. Sci.* **2025**, *15*, 2481. https://doi.org/10.3390/app15052481

1. Introduction

Federated learning (FL) is a distributed learning framework that enables collaborative training across multiple clients while preserving privacy, thereby possessing vast potential for application across numerous domains, such as healthcare [1–3], security [4,5], intelligent driving [6,7], and recommender systems [8–11].

Data heterogeneity is an inevitable problem in federated learning. Currently, there is a significant body of work that recognizes this issue, and considerable efforts have been directed towards addressing it. Model-based methods [12–14] correct the training direction of the model by introducing an auxiliary model or adding regular losses to constrain the training direction of the local model. Several works have addressed this issue from a data perspective by altering the distribution of local data to make it more similar to global distribution, thereby mitigating the problem. For instance, works have utilized Generative Adversarial Networks (GANs) [15] or other specialized and traditional data augmentation methods such as [16–19] to generate missing categories of data locally.

The traditional federated learning approach struggles with data heterogeneity, making it difficult to generate a globally effective model. Personalized federated learning (pFL) was proposed to address this by creating a personalized model for each client, improving adaptation to local data. Rather than relying on a single global model, pFL introduces personalized components locally. For example, FedPer [20] uses the last layer of a convolutional neural network, while FedTP [21] uses the self-attention layer of a transformer [22] network and FedBN [23] uses a BN (batch normalization) layer to personalize the model. Other pFL approaches, like FedALA [24] and HPFL [25], improve model training and aggregation by enhancing local model personalization. They achieve this by creating personalized feature extractors for each client instead of using a shared one. Another promising approach [26,27] leverages knowledge distillation [28], where intermediate or global models act as teacher models, guiding local student models trained on local data, thus improving performance.

Existing pFL frameworks typically decouple the model into a feature extractor and classifier. The local feature extractor is initialized with the globally aggregated one in each round, mitigating data heterogeneity. However, this approach overlooks the continuity of local model training. As shown in Figure 1, the reinitialization of the local feature extractor each round leads to inconsistency between the global model and local knowledge, especially for clients missing multiple rounds of training, referred to as "stragglers". These clients experience a growing deviation from their local knowledge, disrupting consistent learning. Additionally, while the global aggregation helps the feature extractor generalize, the local classifier tends to overfit after several rounds, reducing the generalization and performance on new data. We summarize the challenges of the current pFL framework as two main issues: training discontinuity and classifier overfitting.

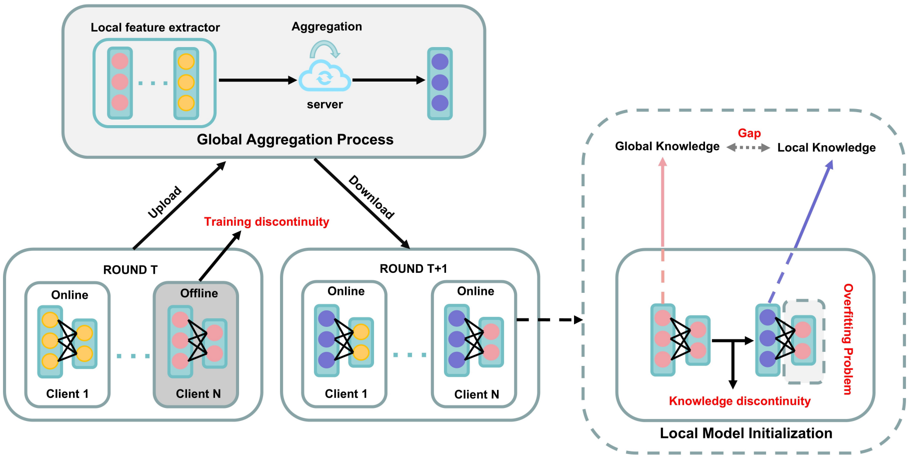


Figure 1. Problem illustration of discontinuity of pFL and classifier overfitting problem.

To address the training discontinuity problem, we introduce a local knowledge recall mechanism designed to retain the model from the previous training round on the client side. This mechanism ensures that when the client receives the updated global model in the subsequent iteration, it can enhance the similarity between the outputs of the global feature extractor and the retained local feature extractor. As shown in Figure 2, we add a stage for knowledge recall before the local model initialization phase. By doing so, the mechanism effectively achieves the goal of knowledge recall, allowing the model to retain and leverage previously acquired local knowledge.

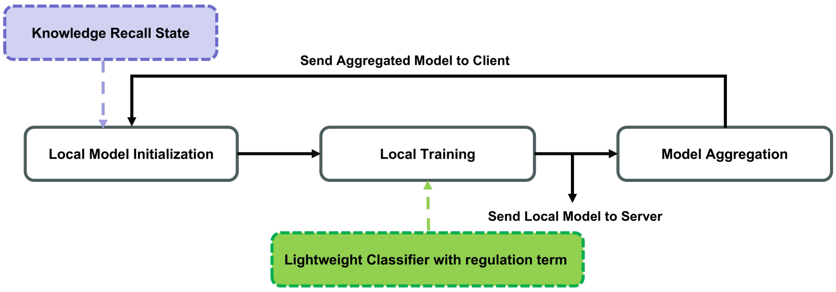

Knowledge Recall State
Send Aggregated Model to Client
Local Model Initialization
Local Training
Model Aggregation
Send Local Model to Server
Lightweight Classifier with regulation term

Figure 2. Simplified diagram of the personalized federated learning framework.

To address the classifier overfitting problem, we propose adding a regularization term to the loss function during the local model training phase. The regularization term aims to mitigate the classifier's tendency to overfit the local training data by making the classifier parameters sparser, making it a lightweight classifier. By incorporating this regularization, we aim to enhance the generalizability of the lightweight classifier, ensuring that it performs well not only on the local training data but also on unseen data. This approach helps to balance the benefits of local training with the need for a globally robust feature extractor, ultimately improving the overall performance of the pFL system.

In this paper, we propose Personalized Federated Learning with a Progressive Local Training Strategy and a Lightweight Classifier (PFPS-LWC). The key contributions of our work are as follows:

1. We propose a Progressive Local Training Strategy that enables a smooth transition from a global model to an initial model with local knowledge and mitigates discontinuity in customer training.
2. We propose a lightweight classifier strategy, which reduces the parameter density and overfitting tendency of the original classifier and alleviates the overfitting problem of the original pFL framework.
3. We conducted extensive evaluations under varying online rates and data distributions, demonstrating the effectiveness of PFPS-LWC through a series of rigorous experiments. These evaluations showed that our proposed method outperforms existing approaches, providing strong evidence of its practical applicability and benefits.

The remainder of this paper is structured as follows: Section 2 reviews related work on federated learning and personalized federated learning. Section 3 presents our proposed method, PFPS-LWC, in detail. Section 4 describes the experimental setup and results, highlighting the effectiveness of our approach. Finally, Section 5 provides a summary and discussion.

2. Related Works

2.1. Federated Learning Under Data Heterogeneity

Data heterogeneity is an inevitable problem due to the diverse sources of data from clients, leading to variations in federated learning. These differences can affect model updates, slowing convergence and reducing the performance of the global model. When there is significant heterogeneity, the aggregation process may incorporate these disparities into the global model, causing local knowledge to be forgotten, especially when initializing local models. This deviation from local data characteristics results in a loss of relevant information for each client.

As the importance of data privacy protection continues to grow, traditional methods of collecting and centrally training data are no longer feasible. For example, in healthcare, data from various hospitals may differ due to variations in equipment, imaging protocols, and patient demographics, resulting in Non-IID data distributions. Similarly, IoT devices, such as smartwatches, home security cameras, and fitness trackers, generate data with significant variations in format, scale, and relevance to users' activities. These discrepancies can affect local model training after aggregation, as well as the overall performance of the global model. This issue extends to other domains like smart homes, where devices like thermostats and security cameras process data locally and contribute to model improvements through aggregated updates. In autonomous vehicles, cars collect data for navigation and safety, training models on local data while ensuring privacy. In the finance sector, bank branches train fraud detection models on transaction data, with updates aggregated to enhance model accuracy while preserving customer privacy. The edge devices mentioned in the examples above, such as in hospitals, IoT devices, smart homes, autonomous vehicles, and bank branches, all correspond to the "clients" depicted in Figure 3.

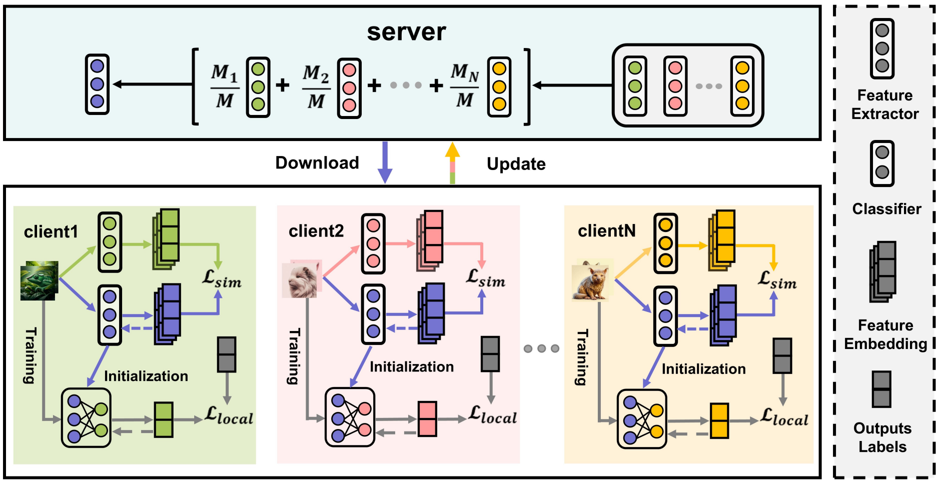


Figure 3. Overview of the proposed PFPS-LWC: the Personalized Federated Learning with a Progressive Local Training Strategy and a Lightweight Classifier method. After the client obtains the global feature extractor, it first executes a Progressive Local Training Strategy. Once the global feature extractor is localized, $\mathcal{L}_{local}$ is then used for local training.

This shift has led to the introduction of the federated learning paradigm, which enables collaborative training across multiple parties without the need to share raw data. Federated learning allows each client to train a model locally on their own data and then share only the model updates with a central server, thereby preserving data privacy. However, federated learning faces significant challenges due to the differences in data distribution among different clients. These differences, collectively known as data heterogeneity, can arise from variations in feature distribution, label distribution, or the data volume. This inconsistency can cause the models to converge at different rates, making it difficult to achieve a unified global model that performs well across all clients. As a result, the overall convergence speed and performance of the federated learning system are adversely affected. Addressing the challenges posed by data heterogeneity is crucial for the successful development and deployment of federated learning systems.

The earliest federated learning paradigm was FedAvg [29]. It achieves model aggregation by simply aggregating the model parameters trained locally on different clients. However, this paradigm has a significant drawback: it relies on the assumption that the data are IID (independently and identically distributed). When the data are Non-IID, its performance is suboptimal. FedProx [14] recognized the impact of data heterogeneity, specifically when data are Non-IID, on global model training. To mitigate this heterogeneity, it introduced a regularization term. This regularization term helps to mitigate the effects of data heterogeneity by stabilizing the training process and ensuring that the model updates from different clients are more aligned. By incorporating this regularization, FedProx aimed to improve the convergence speed and overall performance of the global model, making it more robust to the variations in data distributions across clients.

Various methods have been proposed to address this challenge. For instance, techniques introducing regularization terms such as FedDyn [30], FedDC [31], SCAFFOLD [12], FedDecorr [32], and [33] aim to reduce the gap between local and global models or correct the direction of local model training. These regularization terms help align the local models with the global model, thereby improving the overall consistency and performance of the federated learning system. In addition to regularization-based methods, several approaches have focused on improving federated model aggregation, such as pFedSim [34] and FedDisco [35]. They consider factors like the distribution similarity of local data and the model similarity to generate global models that are more suitable for each client. By taking into account these similarities, these approaches help create global models that better reflect the diverse data distributions across clients, thereby mitigating the impact of data heterogeneity to some extent. In addition, the work based on data enhancement methods, such as FedAug [16], FedM-UNE [19], and FedDA [36], has also alleviated the problem of Non-IID data to a certain extent. These methods aim to balance the classes for clients, reduce the degree of data heterogeneity, and enhance the diversity of the training data, which helps improve the generalizability of the models. Although the aforementioned approaches can learn useful features from heterogeneous data to some extent, they often fail to achieve satisfactory performance when the degree of heterogeneity is high. This is because the variations in data distributions can be too significant for these methods to handle effectively. Therefore, in this paper, we propose to address this issue using a personalized federated learning paradigm. Our approach focuses on tailoring the learning process to the specific characteristics of each client's data, thereby improving the overall performance and robustness of the federated learning system.

2.2. Personalized Federated Learning

Personalized federated learning has garnered significant attention for its outstanding performance on Non-IID data by incorporating personalized factors for each local model. This approach is particularly effective in scenarios where data distributions vary significantly across different clients. The mainstream strategy in pFL is model decoupling, which involves splitting the model into two distinct parts: the feature extractor (also known as the backbone network) and the classifier (or head). These two components are trained using different strategies to achieve a high degree of personalization.

In neural networks, the deeper layers of the network have a stronger correlation with the data distribution. Based on this conclusion, more recent works have taken an approach that uses the classifier as the personalized component. These works primarily use the classifier as the personalized component while treating the feature extractor as the shared component. For example, FedPer [20] treats the classifier as the personalized component, keeping it local, while the feature extractor is treated as the shared component. In this design, each client customizes its own classification standards, which allows for better feature extraction while avoiding discrepancies caused by the classifier. Other methods based on using classifiers as personalization components such as PFedMe [37], Ditto [38], and FedRep [39] promote the learning of global feature extractors by adjusting local training. These approaches ensure that the feature extractor can generalize well across different clients, while the classifier remains tailored to the specific data of each client. The more personalized approaches, like FedFomo [40] and FedALA [24], achieve personalization for each client by combining weighted models for each client and correcting class embedding gradients by injecting cross-client gradient terms. This helps in aligning the learning process across clients and improving the overall model performance. There is also an approach [11] that uses federated graph learning to capture user preferences based on distributed multi-domain data and improve the recommendation performance across all domains without compromising privacy. This method leverages the unique characteristics of graph data to enhance personalization. FedProto [41] employs a prototype learning approach, improving tolerance to heterogeneity by transmitting abstract class prototypes between clients and the server instead of gradients. This reduces the communication overhead and enhances the model's ability to handle diverse data distributions. FedRoD [42] balances general performance and personalized performance by training a general predictor and a personalized predictor through a dual prediction task framework. This approach ensures that the model can perform well on both local and global data.

As opposed to using a classifier as a personalized component, LG-FedAvg [43] proposed using the feature extractor as the personalized component while sharing the classifier among all clients. This strategy provides a unified standard for classification across all models, ensuring consistency in the classification process. At the same time, the feature extractor is tailored to better align with the local data distribution of each client. By doing so, the extracted features are more representative of the local data, thereby mitigating the impact of data heterogeneity to some extent. This decoupling approach allows the feature extractor to capture the unique characteristics of each client's data, enhancing the model's ability to generalize across different data distributions.

The aforementioned works have made efforts in different directions to mitigate the impact of data heterogeneity to varying degrees. However, they often lack consideration of the inherent flaws within the pFL frameworks themselves. These flaws can include issues such as the complexity of model training, the communication overhead, and the difficulty in balancing personalization with generalization. Addressing these inherent flaws is crucial for the further development and effectiveness of personalized federated learning systems.

2.3. Comparison with Existing Methods

In this section, we compare our proposed method with existing representative studies in the field, focusing on key aspects such as the model architecture, method categories, training strategies, and their respective advantages and disadvantages. To facilitate a clear and detailed analysis, we provide tables summarizing the relevant comparisons. As shown in Table 1, the comparison highlights the key differences across various methods.

Table 1. Comparison of federated learning and personalized federated learning methods.

Method	Architecture	Categories	Strategies	Advantages	Disadvantages
FedAvg [29]	CNN	FL	Simple aggregation	Simple and efficient	Sensitive to data heterogeneity
FedProx [14]	CNN	FL	Regularization	Robust to data heterogeneity	More computational costs
FedPer [20]	CNN	pFL	Model peronalization	High adaptability	Prone to overfitting
pFedMe [37]	CNN	pFL	Personalized loss	High personalization	More computional costs
FedBN [23]	CNN	pFL	Batch normalization	Fast convergence	More computional costs
FedProto [41]	CNN	pFL	Prototype learning	High communication efficiency	Slow training
PFPS-LWC	CNN	pFL	Feature alignment and regularization	Efficient and robust	More computional costs

The table provides a comprehensive comparison of various federated learning methods, highlighting the trade-offs between efficiency, personalization, robustness, and computational costs. While methods like FedAvg [29] offer simplicity and efficiency, they struggle in heterogeneous environments. FedProx [14] addresses data heterogeneity through regularization, but with added computational complexity. Personalized methods such as FedPer [20], pFedMe [37], and FedBN [23] offer better adaptation to local data but are more prone to overfitting and incur higher computational costs. PFPS-LWC balances robustness and efficiency, making it a strong choice for environments with data heterogeneity. FedProto [41], focusing on communication efficiency, comes with the downside of slower training. Overall, the choice of method depends on the balance between the need for model personalization, computational resources, and the desired robustness to data heterogeneity.

3. Methodology

3.1. Preliminaries

Personalized federated learning is an important branch of federated learning that aims to address the issue of data heterogeneity among different clients. In a pFL system, assume there are N clients and one server. Each client, $i \in \{1, 2, \ldots, N\}$, holds a private dataset, $D_i = \{(x_j, y_j)\}_{j=1}^{M_i}$, where M_i represents the data volume of client i. Additionally, define $M = \sum_{i=1}^{N} M_i$ as the total volume of all data. Each client's decoupled model parameters are $w_i = (\theta_i, \phi_i)$, where θ_i is the parameters of the feature extractor and ϕ_i is the parameters of the classifier. On the server side, the aggregation of the shared feature extractor is completed, and the parameters of the aggregated feature extractor are denoted as θ_g.

There are two processes in pFL, the local update process and the global aggregation process. Let E denote the total number of epochs for local training in a global round, t, and T represent the total number of global training rounds, while e denotes an intermediate epoch during local training. For client i, it first updates the local feature extractor using the global aggregated feature extractor $\theta_g^{(t-1)}$, and the updated local feature extractor is θ_i^t. Then, θ_i^t is concatenated with ϕ_i^t to form the initialized model $w_i^t = (\theta_i^t, \phi_i^t)$. This model is trained locally for E epochs, resulting in $w_i^{t+E} = (\theta_i^{t+E}, \phi_i^{t+E})$. The updated local feature extractor θ_i^{t+E} is then uploaded to the server. In an ideal scenario, assuming all clients participate in training and updating and the model parameters are aggregated in proportion to the data volume, the updated global feature extractor is $\theta_g^t = \sum_{i=1}^{N} \frac{M_i}{M} \theta_i^{t+E}$, and the aggregated feature extractor is distributed to each client. Under the pFL framework, the overall optimization objective is as follows:

$$\min_{\theta,\phi_i} \left\{ F(\theta, \phi_i) := \sum_{i=1}^{N} \frac{M_i}{M} \mathcal{L}(\theta, \phi_i; D_i) \right\} \tag{1}$$

where $\mathcal{L}$ is the loss of client i on its private dataset, D_i.

3.2. Progressive Local Training Strategy

Current personalized federated learning frameworks decouple models into personalized headers and shared feature extractors. At the beginning of each round, they directly initialize the local feature extractor using the global feature extractor and concatenate it with the personalized head (which always resides locally) to complete the initialization of the local model. Essentially, they ignore the continuity of local training and cause the local knowledge to be forgotten directly. For a certain local client, if it does not participate in the global training for many rounds, the continuity of its local training will be severely damaged, leading to the direct forgetting of local knowledge. Additionally, in cases where there is high data heterogeneity among clients, this heterogeneity is transferred to the local models through the global model. As a result, clients may receive a model that significantly differs from their own data, which is detrimental to local training. To address this issue, we propose a Progressive Local Training Strategy which preserves the local model from the client's last participation in global training. This preserved model is then used to recall knowledge when initializing the global model in the next participating round.

In prior personalized federated learning frameworks, the stage where the global feature extractor is concatenated with the local classifier was commonly referred to as the model initialization stage. This stage is crucial as it sets the foundation for the subsequent training process by combining the global and local components of the model. However, we propose to introduce a new stage prior to this, called the local knowledge recall stage. This additional stage aims to leverage the knowledge gained from previous training rounds, ensuring that the model retains valuable local information before integrating the global feature extractor.

By incorporating the local knowledge recall stage, we aim to enhance the model's ability to utilize previously learned local information, thereby improving its overall performance and stability. The approach helps to mitigate the issue of discontinuity in local training and ensures a smoother transition to the model initialization stage. We refer to this overall strategy as a Progressive Local Training Strategy. This strategy not only improves the alignment between local and global models but also enhances the model's ability to generalize across diverse data distributions.

In this stage, we denote the local feature extractor from the most recent round of participation in the federated process as θ_i^l, where 'l' refers to 'local'. The global feature extractor obtained in the current round is θ_g, and the output of the data after passing through the feature extractor θ is $z_j = f(\theta, x_j)$, where f represents the mapping function from the data to the features. Next, for $x_j^i \in D_i$ belonging to client i, we utilize $z_j^i = f(\theta_i^l, x_j^i)$ as the soft label for x_j^i. Correspondingly, the output of x_j^i on θ_g is z_j^g. Based on the above definition, we establish the following objective to realize the knowledge of the θ_i^l transfer to θ_g:

$$\min_{\theta_g}\left\{F(\theta_g, \theta_i^l) := \mathcal{L}_{sim}(\theta_g, \theta_i^l, x_i^j), where\, x_i^j \in D_i\right\} \tag{2}$$

We employ the cosine similarity function to maximize the similarity between the two outputs and establish the following loss function:

$$\mathcal{L}_{sim} = 1 - \frac{z_j^i \cdot z_j^g}{||z_j^i|| \times ||z_j^g||} = 1 - \frac{f(\theta_i^l, x_j^i) \cdot f(\theta_i^g, x_j^i)}{||f(\theta_i^l, x_j^i)|| \times ||f(\theta_i^g, x_j^i)||}, \tag{3}$$

Then, the update process for θ_g^{t+e} is as follows:

$$\theta_g^{t+e} \leftarrow \theta_g^{t+e-1} - \nabla\mathcal{L}_{sim}(\theta_g, \theta_i^l, x_i^j), where\, x_i^j \in D_i \tag{4}$$

Through the mechanism described above, we facilitate the transfer of knowledge from the local model parameters, denoted as θ_i^l, to the global model parameters, θ_g, by maximizing the similarity between the outputs of two distinct networks when they process an identical dataset. This strategy enables θ_g to assimilate the learning knowledge accumulated by each client during previous training rounds, thereby ensuring the seamless continuation of the local training process, and after the process is finished, we use the updated θ_g to initialize the local feature extractor, establishing the initial local model to carry out subsequent local training. Specially, the initial model establishes a strong connection with local knowledge at this stage, providing a good starting point for local training, making later local training more efficient and effective. After the local training process has been completed, the updated local feature extractor θ_i^{t+E} is used to further refine the local parameters θ_i^l and continue to the next stage.

3.3. Local Training with Lightweight Classifier

The model's ability to generalize is a key factor that directly influences its performance and can be effectively enhanced by reducing the upper bound on its generalization. To quantify the upper bound on generalization, we apply the theory of Rademacher complexity as outlined in [44]. Let $\mathcal{G}$ be a set of functions mapping $\mathcal{Z} \rightarrow [a, b]$, and let S = $(Z_1, \ldots, Z_n)$ be i.i.d. random variables on $\mathcal{Z}$, drawn from some distribution, P. The empirical Rademacher complexity of $\mathcal{G}$ with respect to the sample $(Z_1, \ldots, Z_n)$ is defined as

$$\widehat{\mathfrak{R}_S}(\mathcal{G}) := \mathbb{E}_\sigma\left[\underset{g\in\mathcal{G}}{sup}\,\frac{1}{n}\sum_{i=1}^{n}\sigma_i g(x_i)\right], \tag{5}$$

where $\sigma = (\sigma_1, ..., \sigma_n)^T$ and $\sigma_i \backsim unif\{-1, 1\}$, which are known as Rademacher random variables. Moreover, with a probability of at least $1 - \delta$, we have, with respect to that drawn from S,

$$\forall g \in \mathcal{G}, \mathbb{E}[g(\mathcal{Z})] \leq \frac{1}{n}\sum_{i=1}^{n} g(x_i) + 2\widehat{\mathfrak{R}_S}(\mathcal{G}) + 3(b-a)\sqrt{\frac{log(2/\delta)}{2n}}, \tag{6}$$

where the right-hand side of the equation is the upper bound on the generalization of the model. As evident from Equation (6), the upper bound on the model's generalization capability is determined by three key factors: the empirical risk, the Rademacher risk, and the combined effect of the sample size and the number of categories. For neural networks, the Rademacher risk has a bound with an explicit dependence on the dimension, which is primarily influenced by the model complexity [45].

Inspired by the above theory, reducing the model's complexity can lower the Rademacher risk and improve the model's generalization ability. The classifier head, located at the end of a deep learning model, is responsible for mapping extracted features to class predictions. While it can fit local data distributions and features well, it is prone to overfitting local data [46]. In the pFL framework, the classifier is locally trained and updated, causing its parameters to become overly complex and leading to a significant increase in the model complexity. This increase in complexity tends to raise the Rademacher risk and exacerbate the overfitting phenomenon. To address this issue, we propose to introduce lightweight classifiers to limit the complexity of the model and thus reduce the Rademacher risk and then improve the generalization performance of the model.

Regularization techniques, commonly used in machine learning, help reduce overfitting and improve model generalization. Therefore, we propose adding a classifier regularization term during the local training process in pFL to alleviate the degree of overfitting of the classifier and improve the model performance. In this section, we propose utilizing L2 regularization to enhance the training process of local models. By applying the L2 norm to the classifier, we can control its parameter density, making it relatively sparse. This approach helps achieve the goal of creating lightweight classifiers. Building on the original loss function, we add an L2 regularization term for the lightweight classifier, controlled by the hyperparameter λ. Thus, the local loss consists of two parts: the empirical loss, which we use the cross-entropy loss for, and the regularization loss. For the total loss, we define it as follows:

$$\mathcal{L}_{local} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{REG}, \tag{7}$$

In the above equation, $\mathcal{L}_{CE}$ represents the cross-entropy loss, and $\mathcal{L}_{REG}$ represents the regularization loss.

Here, we define the mapping function from the features to the classifier as h. Thus, the classification output for data x can be represented as $h_\phi(f_\theta(x))$. The definition of $\mathcal{L}_{CE}$ and $\mathcal{L}_{REG}$ are as follows:

$$\mathcal{L}_{CE}(h_\phi(f_\theta(x)), y) = -\frac{1}{M_i}\sum_{j=1}^{M_i} h_\phi(f_\theta(x_j))\log(y_j), \tag{8}$$

$$\mathcal{L}_{REG} = ||\phi||_2 = \sqrt{\sum_{k=1}^{|\phi|} \phi_k^2}, \tag{9}$$

$|\phi|$ represents the number of ϕ's parameters and ϕ_k is the kth parameter of ϕ. Then, the update process of the local model is

$$w_i^{t+e}(\theta_i^{t+e}, \phi_i^{t+e}) \leftarrow w_i^t(\theta_i^t, \phi_i^t) - \nabla \mathcal{L}_{local}. \tag{10}$$

We modify the locally trained loss function and control the regularization degree of the lightweight classifier head by adjusting the weight of the λ regularization term in the loss function. By regularizing the classifiers, the parameters of the local classifiers can be made sparse, which reduces the Rademacher risk of the model to some extent and improves the generalization ability of the model.

3.4. The Convergence of PFPS-LWC

To prove the convergence of PFPS-LWC, we begin by introducing the following assumptions.

Assumption 1. *Lipschitz Smoothness. The gradients of client i's local complete heterogeneous model w_i are $L1 - Lipschitzsmooth$:*

$$\begin{gathered} \|\nabla \mathcal{L}_i^{t_1}\left(w_i^{t_1}; x, y\right) - \nabla \mathcal{L}_i^{t_2}\left(w_i^{t_2}; x, y\right)\| \leq L_1 \|w_i^{t_1} - w_i^{t_2}\|, \\ \forall t_1, t_2 > 0, i \in \{0, 1, ..., N-1\}, (x, y) \in D_i \end{gathered} \tag{11}$$

The above formulation can be further derived as

$$\mathcal{L}_i^{t_1} - \mathcal{L}_i^{t_2} \leqslant \left\langle \nabla \mathcal{L}_i^{t_2}, \left(w_i^{t_1} - w_i^{t_2}\right) \right\rangle + \frac{L_1}{2} \|w_i^{t_1} - w_i^{t_2}\|_2^2 \tag{12}$$

The smoothness assumption guarantees that the gradient updates are controlled and do not cause erratic or unstable model updates, which could hinder convergence. It provides a theoretical foundation for maintaining stability during federated model aggregation and facilitates the model's smooth convergence.

Assumption 2. *Unbiased Gradient and Bounded Variance. Client i's random gradient $g_t^{w,i} = \nabla \mathcal{L}_i^t\left(w_i^t, \mathcal{B}_i^t\right)$ ($\mathcal{B}$ is a batch of local data) is unbiased:*

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i}\left[g_t^{w,i}\right] = \nabla \mathcal{L}_i^t\left(w_i^t\right) \tag{13}$$

and the variance of the random gradient $g_t^{w,i}$ is bounded by

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i}\left[\|\nabla \mathcal{L}_i^t(w_i^t; \mathcal{B}_i^t) - \nabla \mathcal{L}_i^t(w_i^t)\|_2^2\right] \leq \sigma^2 \tag{14}$$

This assumption ensures that the model's optimization process is based on correct and stable gradient updates. It also prevents the introduction of large variance during training, which would otherwise affect the stability of the global model aggregation and hinder convergence.

Assumption 3. *Bounded Prameter Variation. The parameter variations of the homogeneous feature extractor θ_i^t and θ^t before and after aggregation are bounded as*

$$\|\theta^t - \theta_i^t\| \leq \delta^2 \tag{15}$$

This assumption ensures that despite local data heterogeneity, the model updates remain manageable and do not cause large disparities between local and global models. It helps in maintaining the stability of the federated learning system by ensuring that parameter updates are consistently within a reasonable range, facilitating convergence and preventing instability during model aggregation.

Using Assumptions 1 and 2, we can establish the following lemma.

Lemma 1. *There is an upper bound on the loss range of any client's local model, w, in the t local training round.*

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathcal{L}_t^{E+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=1}^{E}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 + \frac{L_1E\eta^2\sigma^2}{2} \tag{16}$$

Leveraging Assumption 3 and Lemma 1, we derive Lemma 2.

Lemma 2. *For the* $(t+1)$ *local training round, the loss of any client before and after aggregating local homogeneous small feature extractors on the server is bounded by*

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathbb{E}[\mathcal{L}_t^{E+1}] + \eta\delta^2 \tag{17}$$

By synthesizing Lemma 1 and 2, we ultimately arrive at the following conclusion:

$$\eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)} \tag{18}$$

Given that ϵ, $L1$, δ^2, σ^2, and E are all positive constants, it follows that η has well-defined solutions. Consequently, when the learning rate η satisfies the aforementioned condition, convergence is assured for any client's local complete heterogeneous model. The detailed certification process is presented in Appendix A.

3.5. An Overview of PFPS-LWC

To provide a concise overview of the proposed method, we present the algorithm in a pseudocode format below, as shown in Algorithm 1. The algorithm is divided into two parts: one for the client-side operations and the other for the server-side operation. Before local training begins on the client side, the model needs to undergo an initialization process. In this step, we optimize the global feature extractor by minimizing the discrepancy between its output and the output of the previous round's feature extractor for this client. The optimized feature extractor is concatenated with the classifier to form the initial model for local training, which is then used in this process. On the server side, we perform aggregation based on the proportion of data from each client. By calculating the data proportion for each client, we determine their aggregation weights, which are then used to aggregate the global feature extractor.

Algorithm 1 PFPS-LWC.

```
Input: Global feature extractor $\theta_g$, local feature extractor that participated in the last round of global training $\theta_i^l$, local classifier $\phi_i^t$, dataset $D_i$, local epoch $E$, global round T, the number of participating clients $N_t$, learning rate $\eta$.
Client Training:
for $i = 1$ to $N_t$ do
    if $\theta_i^l \neq NULL$ then
        Localize global feature extractor $\theta_g$:
        $\theta_g^{t+e} \leftarrow \theta_g^{t+e-1} - \nabla \mathcal{L}_{sim}(\theta_g, \theta_i^l, D_i)$
    end if
    Initialize local model: $(\theta_i^t, \phi_i^t) = (\theta_g, \phi_i^t)$
    for $e = 1$ to $E$ do
        $(\theta_i^{t+e}, \phi_i^{t+e}) \leftarrow ((\theta_i^{t+e-1}, \phi_i^{t+e-1})) - \eta \nabla \mathcal{L}_{local}$
    end for
    Get trained local feature extractor: $\theta_i^{t+E}$
    Update the local feature extractor in the last round
    of global training: $\theta_i^l = \theta_i^{t+E}$
    return $\theta_i^{t+E}$ to server
end for
Server Aggregation:
for $t = 1$ to $T$ do
    $\left\{\theta_1^{t+E}, \theta_2^{t+E}, \ldots, \theta_{N_t}^{t+E}\right\} \leftarrow$ Client Training()
    Compute aggregation weights $[\alpha_1, \alpha_2, \ldots, \alpha_{N_t}]$
    Get the aggregated global feature $\theta_g$
    Send $\theta_g$ to clients who are selected in round t + 1.
end for
```

3.6. *Discussion of the Proposed PFPS-LWC Method*

PFPS-LWC introduces several key innovations that enhance the personalization and robustness of federated learning. One of the most significant contributions is the Progressive Local Training Strategy combined with the local knowledge recall stage, which ensures that each client retains and utilizes its local knowledge across multiple rounds of global training. Unlike traditional methods, where local models are initialized using global feature extractors without consideration for previous local training, PFPS-LWC allows clients to recall previous local knowledge before integrating new global information. This effectively reduces catastrophic forgetting and enables a smoother transition between local and global training phases, significantly improving the model's ability to adapt to the unique characteristics of each client's data.

Additionally, PFPS-LWC improves model generalization through the use of lightweight classifiers. By limiting the classifier complexity, the method reduces the risk of overfitting to local data, which is a common challenge in federated learning systems. This lightweight classifier not only enhances personalization but also mitigates the model complexity, helping to maintain a balance between personalization and generalization.

4. Experiments

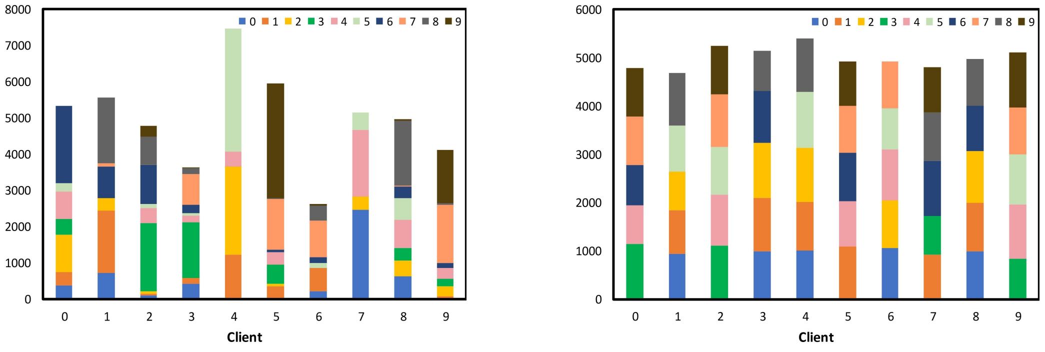
4.1. *Experiment Setup*

Datasets. To evaluate the effectiveness of the proposed federated learning method, we conducted comprehensive experiments on two widely adopted and challenging visual benchmark datasets:

- Cifar10 [47]: The CIFAR10 dataset contains 60,000 32 × 32 color images, evenly distributed across 10 different classes, with each class comprising 6000 images. Among them, 5000 were used for training and 1000 for testing.

- Cifar100 [47]: An extension of CIFAR10, the CIFAR100 dataset includes 60,000 32 × 32 color images, evenly distributed across 100 classes, with each class containing 600 images. Among them, 500 were used for training and 100 for testing.

For each dataset, we employed two data partitioning methods. The first method used the Dirichlet distribution to partition the data, which were then allocated to each client. In Dirichlet partitioning, the parameter controlling the Dirichlet distribution, denoted as β, can be adjusted to control the degree of Non-IID data among different clients. A smaller β indicated a higher degree of Non-IID data, allowing us to simulate varying levels of heterogeneity by adjusting the value of β. The second method was pathological Non-IID partitioning, a manually designed Non-IID data partitioning method. This method involved dividing the dataset into multiple small shards and randomly assigning these shards to different clients, resulting in significant heterogeneity in the data distribution across clients. We conducted experiments under these two heterogeneity simulation settings. This extensive experimental setup simulated real-world scenarios and ensured a comprehensive evaluation of the robustness of our method. In order to more intuitively understand the two partitioning methods, we present a figure showing two data distributions created using Dirichlet partitioning and pathological Non-IID partitioning on the cifar10 dataset below, where the number of clients, N, was 10 and the Dirichlet distribution parameter β = 0.5. The specific distribution after partitioning is shown in Figure 4 below.



(**a**) (**b**)

Figure 4. The distribution of the cifar10 dataset under two data partitioning methods. (**a**) shows the data distribution after applying Dirichlet partitioning on cifar10 when $N = 10$ and $\beta = 0.5$, while (**b**) illustrates the data distribution after applying Pathological Non-IID partitioning on cifar10 when $N = 10$.

Models. For classification tasks on the above datasets, we used a five-layer CNN network. The first four layers of the CNN served as feature extractors, while the final layer functioned as the classifier. For the CIFAR10 dataset, the output dimension of the final layer was set to 10, while for the CIFAR100 task, the output dimension of the final layer was set to 100.

Counterparts. To demonstrate the effectiveness of the proposed method, we select a representative and superior personalized federated learning method including FedPer [20], FedBN [23], pFedMe [37] and FedProto [41]. Additionally, to showcase the superiority of personalized federated learning on Non-IID data, we also compared our method with the representative of traditional personalized federated learning methods such as FedAvg [29] and FedProx [14].

Implementation details. To ensure the fairness of the comparison methods, we used the same hyperparameters in our experiments. In the experiments, we set the number of clients, N, to 10, the number of global training rounds, T, to 20, and the number of local training epochs, E, to 5. We used an SGD optimizer with a learning rate of 0.01 and a batch size of 64. Additionally, we set the parameter λ in the PFPS-LWC method to 0.02. The top-one accuracy was used for the evaluation.

4.2. A Comparison with State-of-the-Art Methods

To validate the effectiveness of our method, we compared it with several state-of-the-art methods in the field of federated learning and conducted a comprehensive evaluation. We used P to represent the proportion of clients participating in training each round. The smaller the P, the fewer clients participated in training, increasing the likelihood of clients being offline for extended periods. In our experiments, we set P to [0.3, 0.5, 0.7] to simulate collaborative training scenarios under different participation rates. This variation in participation rates helped us understand how our method performs under different levels of client availability.

Additionally, for the Dirichlet partitioning method, the smaller the β, the greater the degree of data heterogeneity. We set β to [0.1, 0.3, 0.5] to simulate different levels of data heterogeneity using Dirichlet partitioning methods. These parameter settings effectively simulated environments with high data heterogeneity at different participation rates, allowing us to thoroughly evaluate the robustness of our method. For the pathological Non-IID partitioning method, we set the number of classes to 5 for cifar10 and 30 for cifar100. This approach involved dividing the dataset into multiple small shards and randomly assigning these shards to different clients, resulting in significant heterogeneity in the data distribution across clients. This setup was designed to mimic real-world scenarios where data distributions can vary widely among different clients.

Extensive experiments were conducted in the above simulated environments. The results are shown in Tables 2–4. We report the accuracy of different methods on the cifar10 and cifar100 datasets under various conditions. These tables provide a comprehensive comparison of our method against existing approaches, highlighting its effectiveness in handling data heterogeneity and varying participation rates. In Table 2, we show the experimental results in the case where the dataset was cifar10 and the data partitioning method was Dirichlet partitioning. Notably, when β = 0.5 and P = 0.3, PFPS-LWC surpassed the baseline by an impressive 4.18%. Moreover, PFPS-LWC achieved optimal performance in all settings, compared to other methods. When the dataset was cifar100, with β = 0.1 and P = 0.5, as shown in Table 3, our method surpassed the baseline by an impressive 4.03%. This significant improvement highlights the effectiveness of our approach in scenarios with strong data heterogeneity and lower client participation rates.

Table 2. Prediction accuracy of the proposed method compared with that of state-of-the-art methods on Cifar10 using different Dirichlet settings.

Method	**Accuracy⇑ Predicted with Different Settings of β and P on Cifar10.**								
	$\beta = 0.1$			$\beta = 0.3$			$\beta = 0.5$		
	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$
FedAvg [29]	33.03	34.54	52.3	49.17	56.5	58.95	56.15	56.36	54.14
FedProx [14]	36.35	40.03	53.84	52.59	58.79	59.2	57.34	58.67	56.86
FedPer [20]	89.36	87.09	87.48	79.74	79.01	79.94	75.59	76.21	76.4
pFedMe [37]	87.68	86.08	86.04	77.59	75.73	69.67	70.8	71.8	71.16
FedBN [23]	67.91	84.66	85.29	72.72	74.61	73.85	70.27	72.41	72.96
FedProto [41]	88.56	88.42	89.11	76.96	77.71	77.08	72.31	72.15	72.36
PFPS-LWC	90.58	91.03	90.91	82.36	82.81	83.18	79.77	80.43	80.33
Δ	**1.22**	**3.94**	**3.43**	**2.62**	**3.8**	**3.24**	**4.18**	**4.22**	**3.93**

Table 3. Prediction accuracy of the proposed method compared with that of state-of-the-art methods on Cifar100 using different Dirichlet settings.

Method	**Accuracy⇑ Predicted with Different Settings of β and P on Cifar100.**								
	$\beta = 0.1$			$\beta = 0.3$			$\beta = 0.5$		
	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$
FedAvg [29]	18.68	21.22	23.64	22.81	25.55	26.73	24.05	25.6	27.76
FedProx [14]	19.5	22.03	24.31	23.45	25.72	24.77	23.78	26.11	27.25
FedPer [20]	43.18	43.97	44.92	37.57	37.55	36.71	31.23	31.05	31.32
pFedMe [37]	32.89	34.98	34.04	22.34	24.41	24.58	19.01	20.84	21.19
FedBN [23]	21.17	26.88	29.57	25.71	30.05	32.03	28.36	31.45	32.51
FedProto [41]	35.57	34.31	35.43	20.66	21.5	21.18	13.14	13.27	13.8
PFPS-LWC	46.84	48	48.69	40.32	39.86	40.57	33.59	34.1	33.72
Δ	**3.66**	**4.03**	**3.77**	**2.75**	**2.31**	**3.86**	**2.36**	**3.05**	**2.4**

Table 4. Prediction accuracy of the proposed method compared with that of state-of-the-art methods on Cifar10 and Cifar100 using different pathological settings.

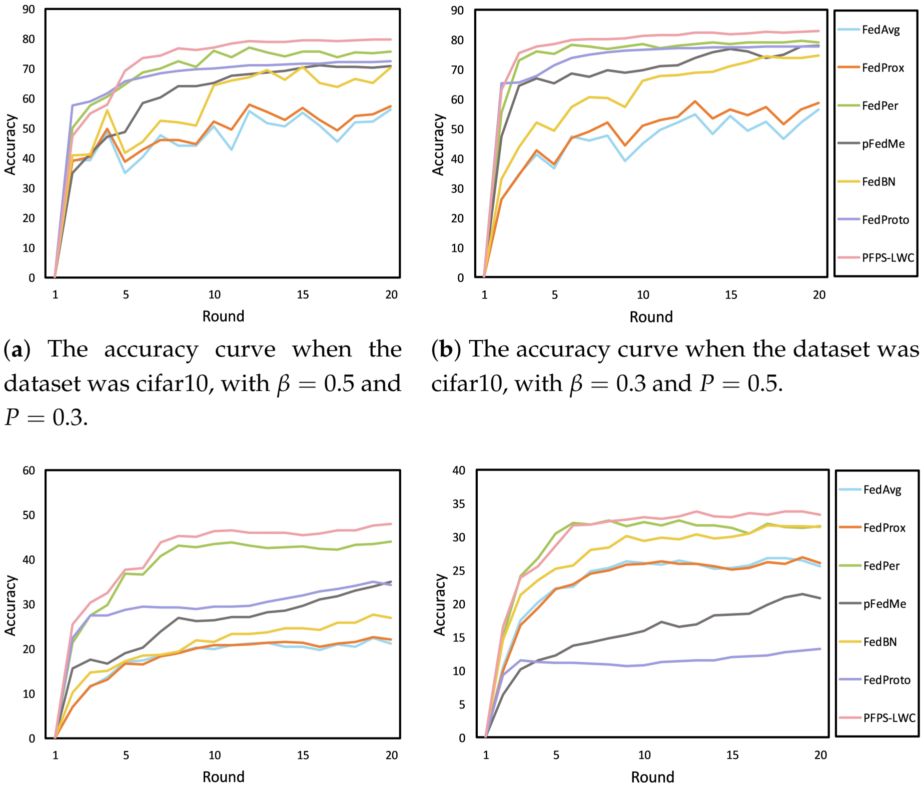
Method	**Accuracy⇑ Predicted with Different Pathological Settings**					
	Cifar10			**Cifar100**		
	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$
FedAvg [29]	44.78	59.55	60.99	20.35	22.64	25.12
FedProx [14]	49.28	60.48	62.76	20.68	23.94	26.01
FedPer [20]	74.68	74.68	74.66	39.12	40.13	40.32
pFedMe [37]	65.17	65.9	66.78	24.68	26.43	25.38
FedBN [23]	65.16	70.5	71.75	26.6	29.56	31.88
FedProto [41]	71.05	70.74	71.01	37.19	34.31	36.86
PFPS-LWC	76.06	76.29	76.64	41.9	43.7	43.35
Δ	**1.38**	**1.61**	**2.22**	**2.78**	**3.57**	**3.03**

In Table 4, we present the experimental results for cifar10 and cifar100 under pathological data partitioning. As shown in the table, our method achieved optimal results in this scenario. Specifically, when the dataset was cifar100 with $p = 0.5$, PFPS-LWC surpassed FedPer's accuracy by up to 3.57%. This demonstrates that our method is effective when the data volume is similar but the types of data differ. It also highlights the importance of retaining local knowledge to mitigate the interference from other types of knowledge.

A thorough examination of Tables 2–4 reveals a compelling observation: under various environmental settings, PFPS-LWC consistently achieved the best results on both the cifar10 and cifar100 datasets, outperforming other methods. This substantial evidence demonstrates the robustness of PFPS-LWC in highly heterogeneous environments and with varying client participation rates. Under different environmental settings, PFPS-LWC also significantly outperformed the baseline, further validating its versatility and robustness. This remarkable advantage underscores the necessity of PFPS-LWC's improvements to the original pFL framework. It supports our viewpoint that simply using the global model to initialize local models is unreasonable. Instead, our approach, which focuses on improving the continuity of local training and alleviating local knowledge forgetting, proves to be more effective. By ensuring that local models retain valuable knowledge from previous training rounds and seamlessly integrate global information, PFPS-LWC enhances the overall effectiveness of federated model training.

Additionally, compared to traditional federated learning methods like FedAvg [29] and FedProx [14], our method significantly outperformed them. In extreme heterogeneous conditions with $\beta = 0.1$ and $P = 0.3$, as shown in Table 3, our method exceeded their accuracy on the cifar10 dataset by an impressive 57.55% and 54.23%, respectively. These substantial improvements highlight the effectiveness of our approach in handling highly heterogeneous data distributions and low client participation rates. This further underscores the potential of PFPS-LWC as a robust solution for federated learning in extremely heterogeneous environments. The ability of our method to maintain high accuracy under such challenging conditions demonstrates its versatility and robustness.

In addition, we present the accuracy curves of different methods during the training process in Figures 5 and 6. These figures provide a direct comparison of the convergence behavior and stability of our method versus those of others across different heterogeneous environments. As shown in both figures, our method achieved faster convergence compared to that of the other methods, highlighting its efficiency in reaching optimal performance. This is particularly evident in the rapid rise in accuracy in the early stages of training, which suggests that our model is able to quickly adapt to local data distributions.



(**a**) The accuracy curve when the dataset was cifar10, with $\beta = 0.5$ and $P = 0.3$.

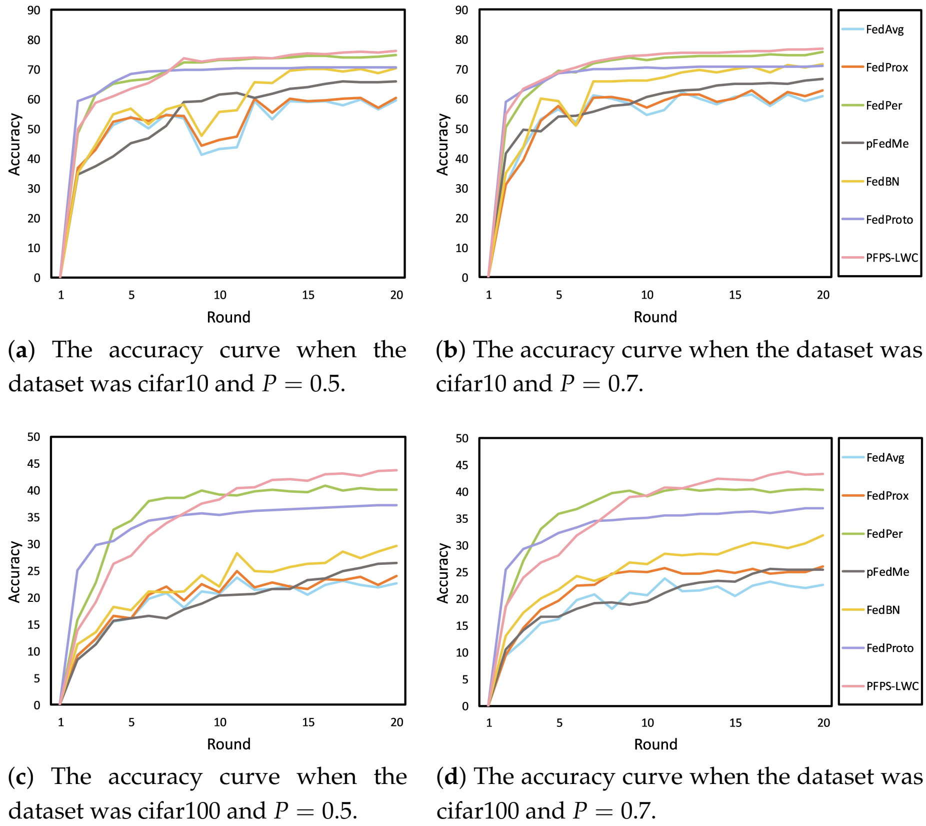
(**b**) The accuracy curve when the dataset was cifar10, with $\beta = 0.3$ and $P = 0.5$.

(**c**) The accuracy curve when the dataset was cifar100, with $\beta = 0.1$ and $P = 0.5$.

(**d**) The accuracy curve when the dataset was cifar100, with $\beta = 0.5$ and $P = 0.5$.

Figure 5. The accuracy curves for cifar10 and cifar100 under different Dirichlet partitioning settings.

Moreover, both figures consistently demonstrate smaller fluctuations in the accuracy curves for our method. This stability across both figures is a key feature indicating that our model maintains consistent performance throughout the training process, avoiding common issues such as overfitting or underfitting. The reduced fluctuations across both figures can be interpreted as a result of incorporating more local knowledge into the model, which helps it better adapt to the specific characteristics of each client's data. This dual advantage, faster convergence and improved stability, further reinforces the effectiveness and robustness of our approach, making it well suited to handle the challenges posed by data heterogeneity in federated learning environments.



(**a**) The accuracy curve when the dataset was cifar10 and $P = 0.5$.

(**b**) The accuracy curve when the dataset was cifar10 and $P = 0.7$.

(**c**) The accuracy curve when the dataset was cifar100 and $P = 0.5$.

(**d**) The accuracy curve when the dataset was cifar100 and $P = 0.7$.

Figure 6. The accuracy curves for cifar10 and cifar100 under different pathological partitioning settings.

4.3. Ablation Experiments

To validate the effectiveness of Module 3.2 (the Progressive Local Training Strategy) and Module 3.3 (the lightweight classifier), we conducted a series of ablation experiments. These experiments were performed on the cifar10 dataset with $\beta = 0.5$ and $P = [0.3, 0.5, 0.7]$, as well as on the cifar100 dataset with $\beta = 0.5$ and $P = [0.3, 0.5, 0.7]$. By varying these parameters, we aimed to simulate different levels of data heterogeneity and client participation rates, providing a comprehensive evaluation of our proposed modules.

The results of these ablation experiments are presented in Table 5 below. This table illustrates the overall effectiveness of the different modules, highlighting the contributions of each component to the performance of our method. By isolating the impact of the Progressive Local Training Strategy and the lightweight classifier, we could better understand their individual and combined effects on the model accuracy and stability.

Table 5. Ablation study: ✓ indicates module usage, × indicates module non-usage.

Module		**Cifar10 ($\beta = 0.5$)**			**Cifar100 ($\beta = 0.1$)**		
PLT	**LWC**	$P = 0.3$	$P = 0.5$	$P = 0.7$	$P = 0.3$	$P = 0.5$	$P = 0.7$
×	×	75.59	76.21	76.4	43.18	43.97	44.92
✓	×	79.13 (3.54)	79.32 (3.11)	78.9 (2.52)	45.87 (2.69)	45.9 (1.93)	46.64 (1.72)
×	✓	76.87 (1.28)	77.15 (0.94)	78.26 (1.86)	44.31 (1.13)	46.12 (2.15)	46.95 (2.03)
✓	✓	79.77 (4.18)	80.43 (4.22)	80.33 (3.93)	46.84 (3.66)	48 (4.03)	48.69 (3.77)

From the table, we can observe that both modules improved the algorithm's performance to varying degrees. For instance, on the cifar10 dataset, the Progressive Local Training (PLT) module, with $\beta = 5$ and $P = 0.3$, showed a 3.53% improvement over the baseline. This improvement highlights the effectiveness of the PLT module in enhancing the model's ability to retain and utilize local knowledge, thereby improving its performance in heterogeneous environments. On the cifar100 dataset, the lightweight classifier (LWC) module, with $\beta = 0.1$ and $P = 0.5$, demonstrated a 2.15% improvement over the baseline. This result underscores the importance of regularizing the classifier head, which undergoes prolonged local training. By incorporating an LWC, the classifier's generalization capability is enhanced, leading to better overall performance.

These experiments demonstrate that both modules significantly enhance the performance of our method, particularly in environments with high data heterogeneity and varying client participation rates. The Progressive Local Training Strategy helps to maintain continuity in local training, while the LWC improves the generalization capability of the classifier. Together, these modules contribute to a more robust and effective federated learning framework.

4.4. Hyperparameter Sensitivity Analysis

Our method involves only one hyperparameter, λ, which controls the weight of the L2 regularization term in the LWC. To determine the optimal value of λ, we conducted a series of experiments on the cifar10 dataset with $\beta = 0.5$ and $P = 0.5$. We explored a range of values for λ to observe its impact on the model performance. In the range [0, 0.1], we used a step size of 0.02 to finely tune the regularization weight. For the range [0.1, 0.3], we increased the step size to 0.05 to cover a broader spectrum of values. Finally, in the range [0.3, 1], we used a step size of 0.1 to efficiently explore higher values of λ. This systematic approach allowed us to comprehensively evaluate the effect of λ on the model's accuracy.

As illustrated in Figure 7, the algorithm exhibited relatively stable performance and notable improvement when the λ parameter fell within the range of [0, 0.2]. This indicates that moderate regularization helps enhance the model's generalization capability without overly constraining it. However, as λ increased beyond this range, the performance showed a fluctuating decline. This degradation in performance with larger λ values can be primarily attributed to the increased weight of the regularization term, which causes the optimization of the overall loss function to overly favor the regularization of the classifier. This excessive regularization can hinder the model's ability to learn effectively from the local data.

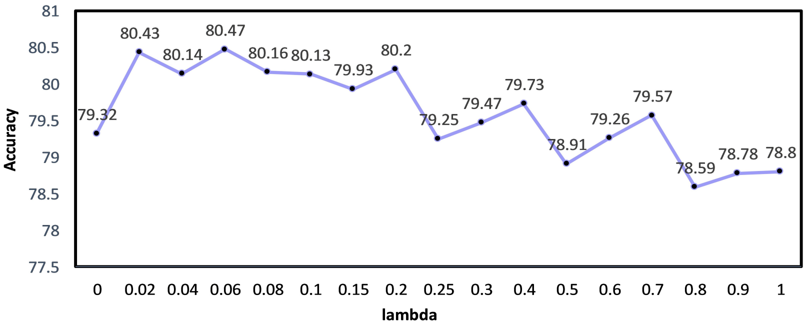


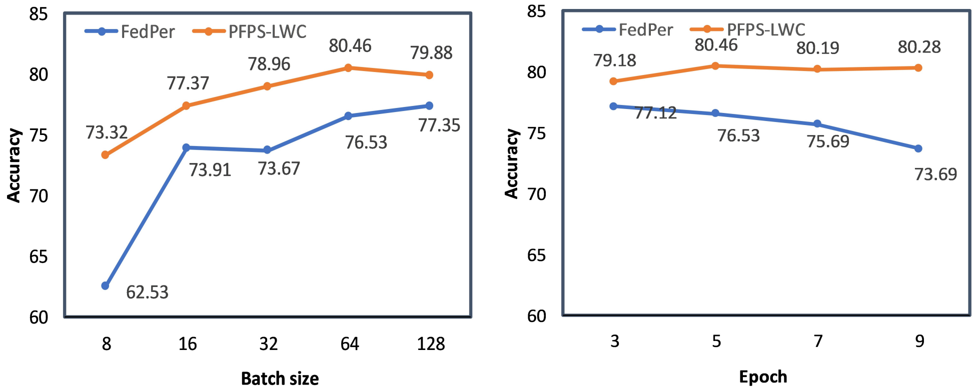
Figure 7. The curve for cifar10 with different lambda settings.

Therefore, it is recommended to utilize λ values within a smaller range to achieve optimal performance. Furthermore, the overall fluctuation range of the algorithm's performance remained relatively small across different ranges of λ, indicating the algorithm's stability and providing a higher fault tolerance for the selection of the λ parameter. This stability is crucial for ensuring consistent performance across different training scenarios and data distributions.

4.5. Discussion of Other Parameters

To further validate the robustness of our method under highly heterogeneous conditions, we conducted experiments with parameter settings that were independent of our algorithm. Specifically, we chose the parameters batch size and local training epochs, E. By varying the ranges of the batch size and E, we assessed the performance of the algorithm under different conditions. These experiments were conducted on the CIFAR10 dataset with β set to 0.5 and P set to 0.5. In the experiments, the $batch - sizes$ were set to [8, 16, 32, 64, 128], and the values for E were set to [3, 5, 7, 9]. By exploring these different configurations, we aimed to understand how our method performs with varying batch sizes and training epochs, which are critical factors in the training process. The results of the experiment are shown in the figure below. These results provide insights into the stability and effectiveness of our method across different training settings.

Firstly, the batch size has a significant impact on model training. Generally, larger batch sizes lead to better training performance; however, this can be difficult to achieve for clients with limited computational resources. As shown in Figure 8a, when the batch size was small, the traditional personalized federated learning method FedPer performed poorly in local training. In contrast, PFPS-LWC significantly improved the local training performance with smaller batch sizes, surpassing FedPer in accuracy by 10.79%. This indicates that our method effectively mitigates the shortcomings of traditional personalized federated learning frameworks in resource-constrained environments. Furthermore, PFPS-LWC consistently outperformed FedPer across different batch sizes, demonstrating the robustness of our method under various settings.



(**a**) The accuracy curve for varying batch sizes. (**b**) The accuracy curve for varying epochs.

Figure 8. The accuracy curves for changing batch sizes and epochs.

The local epochs also play a crucial role in the local training performance. As shown in Figure 8b, the performance of traditional personalized federated learning was highly sensitive to the number of local epochs, with FedPer's accuracy fluctuating significantly across different ranges. In contrast, PFPS-LWC exhibited much smaller fluctuations and consistently outperformed FedPer in terms of accuracy. Notably, when E = 9, PFPS-LWC surpassed FedPer by 6.59% in terms of accuracy. This not only highlights the stability of PFPS-LWC across different numbers of local epochs but also demonstrates its superiority over FedPer.

These findings underscore the effectiveness of our method in various training scenarios. By maintaining high performance with smaller batch sizes and exhibiting stable accuracy across different numbers of local epochs, PFPS-LWC proves to be a robust and adaptable solution. This robustness is particularly valuable in real-world federated learning environments, where computational resources and data distributions can vary widely among clients.

4.6. Analysis of Results and Future Research

Through extensive experiments, we have validated the effectiveness of the PFPS-LWC approach in personalized federated learning, particularly in addressing critical challenges such as data heterogeneity and fluctuations in client participation rates. We conducted experiments on the cifar10 and cifar100 datasets, using various data partitioning methods, including Dirichlet and pathological Non-IID partitioning. The results show that PFPS-LWC performs exceptionally well across different environmental settings, significantly outperforming existing methods, thus demonstrating the robustness of our approach.

In scenarios with varying data distributions and participation rates, PFPS-LWC exhibits greater adaptability compared to traditional federated learning methods. By employing Progressive Local Training Strategies and lightweight classifiers, PFPS-LWC effectively enhances the client's local model performance and significantly improves the accuracy and stability of the global model. Moreover, in environments with limited computational resources and high data heterogeneity, PFPS-LWC shows notable advantages, demonstrating its strong robustness and generalization capability.

Despite its strong performance, PFPS-LWC has some limitations. In scenarios with extremely high data heterogeneity, there is still potential for further performance improvements. Future work could focus on refining the knowledge recall mechanism to enhance the model's adaptability and stability in such challenging environments. Additionally, while PFPS-LWC has been evaluated on visual datasets like cifar10 and cifar100, its applicability could be extended to more complex, domain-specific datasets, such as medical or IoT data, to assess its broader effectiveness.

Regarding changes in the model architecture, we recognize that different network architectures can impact both the results and conclusions of the model. The five-layer CNN architecture we are currently using performed well in our experiments and is widely applied in related research. However, with changes in the network architecture (such as adding or removing layers or altering the number of neurons per layer), the model's performance and the convergence speed of the global model may vary. Altering the architecture could lead to improvements or declines in performance, depending on the model's complexity, the nature of the training data, and computational resource limitations. Theoretically, however, the PFPS-LWC method is a systematic approach, and its performance is less affected by structural changes. Therefore, future research could explore the impact of different architectures on the effectiveness of PFPS-LWC and further optimize the architectural design to enhance performance across different data distributions and application scenarios.

Future research can focus on several key areas: first, further optimizing the knowledge recall mechanism to address challenges posed by extreme data heterogeneity; second, enhancing the computational efficiency of the model, particularly for edge computing and IoT devices, to reduce computational and storage burdens; and finally, exploring the potential of PFPS-LWC in cross-domain applications (such as healthcare, autonomous driving, etc.), with a focus on data privacy protection and real-time adaptability.

5. Conclusions

In this paper, we explored the critical issues of local knowledge discontinuity and local classifier overfitting in existing personalized federated learning frameworks. These challenges can significantly hinder the performance and generalizability of federated learning models, especially in environments with highly heterogeneous data. To address these problems, we introduced the PFPS-LWC method, which specifically targets and alleviates these challenges. By employing a Progressive Local Training Strategy, we achieved the local correction of the global model's parameters, thereby enhancing the continuity of local knowledge. This strategy ensures that the model retains valuable local information across training rounds, improving its ability to generalize and perform well on diverse data distributions. Additionally, we incorporated a lightweight classifier by adding a regularization term to the loss function. This approach mitigates the overfitting of local classifiers, ensuring that they do not become overly specialized to the local training data. By balancing the regularization term, we enhanced the generalization capability of the classifiers, leading to more robust and effective models.

Through extensive experiments, we comprehensively validated the effectiveness and robustness of PFPS-LWC. Our experimental results demonstrated significant improvements in the model performance across various settings, highlighting the method's ability to handle data heterogeneity and varying client participation rates. We hope that our work provides new insights for future research on personalized federated learning in the context of highly heterogeneous data. By addressing the inherent flaws in existing frameworks and proposing a robust solution, we aim to advance the field and contribute to the development of more effective federated learning systems.

Author Contributions: Conceptualization, J.L.; formal analysis, W.G.; funding acquisition, J.G.; methodology, J.L.; project administration, J.G.; software, J.L.; supervision, J.G. and J.R.; validation, J.L. and Z.F.; visualization, W.G.; writing—original draft, J.L.; writing—review and editing, W.G. All authors have read and agreed to the published version of the manuscript.

Funding: The present study was partly supported by the Natural Science Foundation of Shandong Province under Grant ZR2023MF041, the National Natural Science Foundation of China under Grant 62072469, the Shandong Data Open Innovative Application Laboratory, and the Brazilian National Council for Scientific and Technological Development—CNPq, via Grant No. 306607/2023-9.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used for the proposed method were the publicly available cifar10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html) (accessed on 15 January 2025) and cifar100 dataset (https://www.cs.toronto.edu/~kriz/cifar.html) (accessed on 15 January 2025).

Acknowledgments: Wenjuan Gong acknowledges the support of the Natural Science Foundation of Shandong Province under Grant ZR2023MF041, the National Natural Science Foundation of China under Grant 62072469, and the Shandong Data Open Innovative Application Laboratory. Jordi Gonzàlez acknowledges the support of the Spanish Ministry of Economy and Competitiveness (MINECO) and the European Regional Development Fund (ERDF) under Project No. PID2020-120611RBI00/AEI/10.13039/501100011033.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. The Convergence Proof Process for PFPS-LWC

Assumption A1. *Lipschitz Smoothness. The gradients of client i's local complete heterogeneous model w_i are $L1 - Lipschitzsmooth$:*

$$\|\nabla\mathcal{L}_i^{t_1}\left(w_i^{t_1};x,y\right)-\nabla\mathcal{L}_i^{t_2}\left(w_i^{t_2};x,y\right)\|\leq L_1\|w_i^{t_1}-w_i^{t_2}\|,\\ \forall t_1,t_2>0,i\in\{0,1,...,N-1\},(x,y)\in D_i \tag{A1}$$

The above formulation can be further derived as

$$\mathcal{L}_i^{t_1}-\mathcal{L}_i^{t_2}\leqslant\left\langle\nabla\mathcal{L}_i^{t_2},\left(w_i^{t_1}-w_i^{t_2}\right)\right\rangle+\frac{L_1}{2}\|w_i^{t_1}-w_i^{t_2}\|_2^2 \tag{A2}$$

Assumption A2. *Unbiased Gradient and Bounded Variance. Client i's random gradient* $g_{w,i}^t = \nabla\mathcal{L}_i^t(w_i^t, \mathcal{B}_i^t)$ *($\mathcal{B}$ is a batch of local data) is unbiased:*

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i}[g_{w,i}^t] = \nabla\mathcal{L}_i^t(w_i^t) \tag{A3}$$

and the variance of the random gradient $g_{w,i}^t$ *is bounded by*

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i}\left[\|\nabla\mathcal{L}_i^t(w_i^t; \mathcal{B}_i^t) - \nabla\mathcal{L}_i^t(w_i^t)\|_2^2\right] \leq \sigma^2 \tag{A4}$$

Assumption A3. *Bounded Prameter Variation. The parameter variations of the homogeneous small feature extractor* θ_i^t *and* θ^t *before and after aggregation are bounded as*

$$\|\theta^t - \theta_i^t\| \leq \delta^2 \tag{A5}$$

Lemma A1. *There is an upper bound on the loss range of any client's local model, w, in the t local training round.*

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathcal{L}_t^{E+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=1}^{E}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 + \frac{L_1 E\eta^2\sigma^2}{2} \tag{A6}$$

Lemma A2. *For the* $(t+1)$ *local training round, the loss of any client before and after aggregating local homogeneous small feature extractors on the server is bounded by*

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathbb{E}[\mathcal{L}_t^{E+1}] + \eta\delta^2 \tag{A7}$$

Based on the above assumptions, we can perform a further derivation. For convenience, we write an arbitrary client i's local model as w, and w can be updated by $w_{t+1} = w_t - \eta g_{w,t}$ in the (t + 1) round, and following Assumption A1, we can obtain

$$\mathcal{L}_t^{E+1} - \mathcal{L}_t^{E+0} \leq \left\langle \nabla\mathcal{L}_t^{E+0}, (w_t^{E+1} - w_t^{E+0}) \right\rangle + \frac{L_1}{2}\|w_t^{E+1} - w_t^{E+0}\|_2^2 \tag{A8}$$

$$\mathcal{L}_t^{E+1} \leq \mathcal{L}_t^{E+0} - \eta\left\langle \nabla\mathcal{L}_t^{E+0}, g_{w,t}^{E+0} \right\rangle + \frac{L_1\eta^2}{2}\|g_{w,t}^{E+0}\|_2^2. \tag{A9}$$

Taking the expectation of both sides of the inequality concerning the random variable $\mathcal{E}_t^{E+0}$, we obtain

$$\mathbb{E}[\mathcal{L}_t^{E+1}] \leq \mathcal{L}_t^{E+0} - \eta\mathbb{E}[\left\langle \nabla\mathcal{L}_t^{E+0}, g_{w,t}^{E+0} \right\rangle] + \frac{L_1\eta^2}{2}\mathbb{E}[\|g_{w,t}^{E+0}\|_2^2] \tag{A10}$$

And then, based on the functions (A3) and (A4) from Assumption A2 and $Var(x) = \mathbb{E}[x^2] - (\mathbb{E}[x]^2)$, we can derive that

$$\eta\mathbb{E}[\left\langle \nabla\mathcal{L}_t^{E+0}, g_{w,t}^{E+0} \right\rangle] = \eta\|\nabla\mathcal{L}_t^{E+0}\|_2^2 \tag{A11}$$

$$\frac{L_1\eta^2}{2}\mathbb{E}[\|g_{w,t}^{E+0}\|_2^2] = \frac{L_1\eta^2}{2}((\mathbb{E}[\|g_{w,t}^{E+0}\|]_2^2) + Var(g_{w,t}^{E+0})) \tag{A12}$$

$$\frac{L_1\eta^2}{2}((\mathbb{E}[\|g_{w,t}^{E+0}\|]_2^2)) = \frac{L_1\eta^2}{2}\|\nabla\mathcal{L}_t^{E+0}\|_2^2 \tag{A13}$$

Using functions (A11)–(A13), we can obtain that

$$\mathbb{E}[\mathcal{L}_t^{E+1}] \leq \mathcal{L}_t^{E+0} - \eta\|\nabla\mathcal{L}_t^{E+0}\|_2^2 + \frac{L_1\eta^2}{2}(\|\nabla\mathcal{L}_t^{E+0}\|_2^2 + \sigma^2) \tag{A14}$$

Merge items of the same type as above:

$$\mathbb{E}[\mathcal{L}_t^{E+1}] \leq \mathcal{L}_t^{E+0} + (\frac{L_1\eta^2}{2} - \eta)\|\nabla\mathcal{L}_t^{E+0}\|_2^2 + \frac{L_1\eta^2\sigma^2}{2} \tag{A15}$$

Taking the expectation of both sides of the inequality for the model w over E iterations, we obtain

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathcal{L}_t^{E+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=1}^{E}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 + \frac{L_1E\eta^2\sigma^2}{2} \tag{A16}$$

This concludes the proof of Lemma A1.

Next, we are going to prove Lemma A2.

$$\mathcal{L}_{t+1}^{E+0} = \mathcal{L}_{t+1}^{E} + \mathcal{L}_{t+1}^{E+0} - \mathcal{L}_{t+1}^{E} \approx \mathcal{L}_{t+1}^{E} + \eta\|\theta_{t+1}^{E+0} - \theta_{t+1}^{E}\|_2^2 \leq \mathcal{L}_{t+1}^{E} + \eta\delta^2 \tag{A17}$$

Taking the expectation of both sides of the inequality, we obtain

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathbb{E}[\mathcal{L}_t^{E+1}] + \eta\delta^2 \tag{A18}$$

This concludes the proof of Lemma A2.

Next, we are going to prove the conclusion. Substituting Lemma A1 into the right side of Lemma A2's inequality, we obtain

$$\mathbb{E}[\mathcal{L}_{t+1}^{E+0}] \leq \mathcal{L}_t^{E+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=1}^{E}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2 \tag{A19}$$

Through transformation, we can obtain

$$\sum_{e=1}^{E}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 \leq \frac{\mathcal{L}_t^{E+0} - \mathbb{E}[\mathcal{L}_{t+1}^{E+0}] + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} \tag{A20}$$

Taking the expectation of both sides of the inequality over rounds $t = [0, T-1]$ for w, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 \leq \frac{\frac{1}{T}\sum_{t=0}^{T-1}[\mathcal{L}_t^{E+0} - \mathbb{E}[\mathcal{L}_{t+1}^{E+0}]] + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} \tag{A21}$$

Let $\Delta = \mathcal{L}_{t=0} - \mathcal{L}^* > 0$; then $\sum_{t=0}^{T-1}[\mathcal{L}_t^{E+0} - \mathbb{E}[\mathcal{L}_{t+1}^{E+0}]] \leq \Delta$, and we can obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_t^{E+e}\|_2^2 \leq \frac{\frac{\Delta}{T} + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} \tag{A22}$$

If the above equation converges to a constant ϵ,

$$\frac{\frac{\Delta}{T} + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} \leq \epsilon \tag{A23}$$

then

$$T > \frac{\Delta}{\epsilon(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1E\eta^2\sigma^2}{2} - \eta\delta^2} \tag{A24}$$

Since $T > 0$ and $\Delta > 0$, we can obtain

$$\epsilon(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1 E\eta^2\sigma^2}{2} - \eta\delta^2 > 0 \tag{A25}$$

Solving the above inequality yields

$$\eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)} \tag{A26}$$

Since ϵ, $L1$, δ^2, σ^2, and E are all constants greater than 0, η has solutions. Therefore, when the learning rate η satisfies the above condition, any client's local complete heterogeneous model can converge.

References

1. Antunes, R.S.; André da Costa, C.; Küderle, A.; Yari, I.A.; Eskofier, B. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–23. [CrossRef]
2. Fu, W.; Wang, H.; Gao, C.; Liu, G.; Li, Y.; Jiang, T. Privacy-Preserving Individual-Level COVID-19 Infection Prediction via Federated Graph Learning. *ACM Trans. Inf. Syst.* **2024**, *42*, 82. [CrossRef]
3. Zhang, F.; Kreuter, D.; Chen, Y.; Dittmer, S.; Tull, S.; Shadbahr, T.; Schut, M.; Asselbergs, F.; Kar, S.; Sivapalaratnam, S.; et al. Recent methodological advances in federated learning for healthcare. *Patterns* **2024**, *5*, 101006 [CrossRef]
4. Qammar, A.; Karim, A.; Ning, H.; Ding, J. Securing federated learning with blockchain: A systematic literature review. *Artif. Intell. Rev.* **2023**, *56*, 3951–3985. [CrossRef] [PubMed]
5. Zhuang, W.; Wen, Y.; Zhang, X.; Gan, X.; Yin, D.; Zhou, D.; Zhang, S.; Yi, S. Performance optimization of federated person re-identification via benchmark analysis. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 955–963.
6. Fantauzzo, L.; Fanì, E.; Caldarola, D.; Tavera, A.; Cermelli, F.; Ciccone, M.; Caputo, B. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 11504–11511.
7. Zhang, S.; Li, J.; Shi, L.; Ding, M.; Nguyen, D.C.; Tan, W.; Weng, J.; Han, Z. Federated learning in intelligent transportation systems: Recent applications and open problems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 3259–3285 [CrossRef]
8. Imran, M.; Yin, H.; Chen, T.; Nguyen, Q.V.H.; Zhou, A.; Zheng, K. ReFRS: Resource-efficient Federated Recommender System for Dynamic and Diversified User Preferences. *ACM Trans. Inf. Syst.* **2023**, *43*, 1–30. [CrossRef]
9. Le, Q.; Diao, E.; Wang, X.; Anwar, A.; Tarokh, V.; Ding, J. Personalized federated recommender systems with private and partially federated autoencoders. In Proceedings of the 2022 56th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 31 October–2 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1157–1163.
10. Lin, Z.; Pan, W.; Yang, Q.; Ming, Z. A Generic Federated Recommendation Framework via Fake Marks and Secret Sharing. *ACM Trans. Inf. Syst.* **2022**, *41*, 1–37. [CrossRef]
11. Tian, C.; Xie, Y.; Chen, X.; Li, Y.; Zhao, X. Privacy-preserving Cross-domain Recommendation with Federated Graph Learning. *ACM Trans. Inf. Syst.* **2024**, *42*, 135. [CrossRef]
12. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning., PMLR, Virtual, 13–18 July 2020; pp. 5132–5143.
13. Li, Q.; He, B.; Song, D. Model-contrastive federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10713–10722.
14. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
16. Hao, W.; El-Khamy, M.; Lee, J.; Zhang, J.; Liang, K.J.; Chen, C.; Duke, L.C. Towards fair federated learning with zero-shot data augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3310–3319.
17. Li, Z.; Sun, Y.; Shao, J.; Mao, Y.; Wang, J.H.; Zhang, J. Feature matching data synthesis for non-iid federated learning. *IEEE Trans. Mob. Comput.* **2024**, *23*, 9352–9367. [CrossRef]
18. Rasouli, M.; Sun, T.; Rajagopal, R. Fedgan: Federated generative adversarial networks for distributed data. *arXiv* **2020**, arXiv:2006.07228.

19. Zhang, H.; Hou, Q.; Wu, T.; Cheng, S.; Liu, J. Data augmentation based federated learning. *IEEE Internet Things J.* **2023**, *10*, 22530–22541. [CrossRef]
20. Arivazhagan, M.G.; Aggarwal, V.; Singh, A.K.; Choudhary, S. Federated learning with personalization layers. *arXiv* **2019**, arXiv:1912.00818.
21. Li, H.; Cai, Z.; Wang, J.; Tang, J.; Ding, W.; Lin, C.T.; Shi, Y. Fedtp: Federated learning by transformer personalization. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 13426–13440. [CrossRef]
22. Vaswani, A. Attention is all you need. *srXiv* **2017**. [CrossRef]
23. Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv* **2021**, arXiv:2102.07623.
24. Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; Guan, H. Fedala: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11237–11244.
25. Liu, Q.; Wu, J.; Huang, Z.; Wang, H.; Ning, Y.; Chen, M.; Chen, E.; Yi, J.; Zhou, B. Federated User Modeling from Hierarchical Information. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–33. [CrossRef]
26. Han, S.; Park, S.; Wu, F.; Kim, S.; Wu, C.; Xie, X.; Cha, M. Fedx: Unsupervised federated learning with cross knowledge distillation. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 691–707.
27. Jiang, D.; Shan, C.; Zhang, Z. Federated learning algorithm based on knowledge distillation. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 23–25 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 163–167.
28. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
29. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
30. Acar, D.A.E.; Zhao, Y.; Navarro, R.M.; Mattina, M.; Whatmough, P.N.; Saligrama, V. Federated learning based on dynamic regularization. *arXiv* **2021**, arXiv:2111.04263.
31. Gao, L.; Fu, H.; Li, L.; Chen, Y.; Xu, M.; Xu, C.Z. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10112–10121.
32. Shi, Y.; Liang, J.; Zhang, W.; Tan, V.Y.; Bai, S. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. *arXiv* **2022**, arXiv:2210.00226.
33. Zheng, X.; Xie, H.; Guo, Y.; Bie, R. FedIR: Learning Invariant Representations from Heterogeneous Data in Federated Learning. In Proceedings of the 2023 19th International Conference on Mobility, Sensing and Networking (MSN), Nanjing, China, 14–16 December 2023; IEEE: Berlin/Heidelberg, Germany, 2023; pp. 644–651.
34. Tan, J.; Zhou, Y.; Liu, G.; Wang, J.H.; Yu, S. pFedSim: Similarity-Aware Model Aggregation Towards Personalized Federated Learning. *arXiv* **2023**, arXiv:2305.15706.
35. Ye, R.; Xu, M.; Wang, J.; Xu, C.; Chen, S.; Wang, Y. Feddisco: Federated learning with discrepancy-aware collaboration. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 39879–39902.
36. Zhang, J.; Jiang, Y. A data augmentation method for vertical federated learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 6596925. [CrossRef]
37. Dinh, C.T.; Tran, N.; Nguyen, J. Personalized federated learning with moreau envelopes. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21394–21405.
38. Li, T.; Hu, S.; Beirami, A.; Smith, V. Ditto: Fair and robust federated learning through personalization. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 6357–6368.
39. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. Exploiting shared representations for personalized federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 2089–2099.
40. Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; Alvarez, J.M. Personalized federated learning with first order model optimization. *arXiv* **2020**, arXiv:2012.08565.
41. Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2022; Volume 36, pp. 8432–8440.
42. Chen, H.Y.; Chao, W.L. On bridging generic and personalized federated learning for image classification. *arXiv* **2021**, arXiv:2107.00778.
43. 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* **2020**, arXiv:2001.01523.

44. Scott, C. Rademacher Complexity. 2014. Available online: https://web.eecs.umich.edu/~cscott/past_courses/eecs598w14/notes/10_rademacher.pdf (accessed on 10 January 2025).
45. Yin, D.; Kannan, R.; Bartlett, P. Rademacher complexity for adversarially robust generalization. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7085–7094.
46. Yang, Y.; Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19290–19301.
47. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. In *Handbook of Systemic Autoimmune Diseases*; University of Toronto: Toronto, ON, Cannada, 2009; Volume 1, pp. 1–58.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.