How to Prevent the Poor Performance Clients for Personalized Federated Learning?

Zhe Qu[1], Xingyu Li[2]*, Xiao Han[3], Rui Duan[3], Chengchao Shen[1], and Lixing Chen[4]
[1]Central South University, [2]Mississippi State University,
[3]University of South Florida, [4]Shanghai Jiaotong University
{zhe_qu, scc.cs}@csu.edu.cn, xl292@msstate.edu,
{xiaoh, ruiduan}@usf.edu, lxchen@sjtu.edu.cn

Abstract

Personalized federated learning (pFL) collaboratively trains personalized models, which provides a customized model solution for individual clients in the presence of heterogeneous distributed local data. Although many recent studies have applied various algorithms to enhance personalization in pFL, they mainly focus on improving the performance from averaging or top perspective. However, part of the clients may fall into poor performance and are not clearly discussed. Therefore, how to prevent these poor clients should be considered critically. Intuitively, these poor clients may come from biased universal information shared with others. To address this issue, we propose a novel pFL strategy, called Personalize Locally, Generalize Universally (PLGU). PLGU generalizes the fine-grained universal information and moderates its biased performance by designing a Layer-Wised Sharpness Aware Minimization (LWSAM) algorithm while keeping the personalization local. Specifically, we embed our proposed PLGU strategy into two pFL schemes concluded in this paper: with/without a global model, and present the training procedures in detail. Through in-depth study, we show that the proposed PLGU strategy achieves competitive generalization bounds on both considered pFL schemes. Our extensive experimental results show that all the proposed PLGU based-algorithms achieve state-of-the-art performance.

1. Introduction

Federated Learning (FL) is a popular collaborative research paradigm that trains an aggregated global learning model with distributed private datasets on multiple clients [16, 29]. This setting has achieved great accomplishments when the local data cannot be shared due to privacy and communication constraints [36]. However, because of the non-IID/heterogeneous datasets, learning a single global model to fit the "averaged distribution" may be difficult to propose a well-generalized solution to the individual client and slow the convergence results [24]. To address this problem, personalized federated learning (pFL) is developed to provide a customized local model solution for each client based on its statistical features in the private training dataset [5, 9, 11, 34]. Generally, we can divide existing pFL algorithms into two schemes: (I) with a global model [5, 23, 25, 40] or (II) without a global model [27, 28, 37].

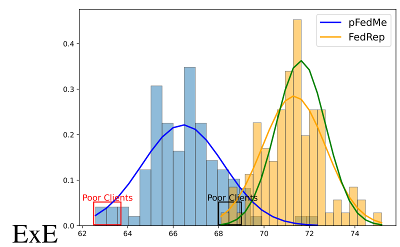


Figure 1. Toy example in a heterogeneous pFL on CIFAR10, which includes 100 clients and each client obtains 3 labels.

Though many pFL algorithms make accomplishments by modifying the universal learning process [41, 47] or enhancing the personalization [3, 27, 37], they may lead part of clients to fall into poor learning performance, where the personalization of local clients performs a large statistical deviation from the "averaged distribution". To the best of our knowledge, none of the existing studies explore how to prevent clients from falling into poor personalized performance on these two schemes. For example, the poor medical learning models of some clients may incur serious medical malpractice. To better present our concerned problem, we introduce a toy example in Figure 1, which is learned by two pFL algorithms representing these two schemes: pFedMe [40] and FedRep [3]. Though both algorithms achieve high averaged local model performance of 66.43% and 71.35%, there also 15% of clients are less than 64% and 14% clients are less than 69%, respectively.

*Corresponding author.

This motivates us to exploit an effective strategy to prevent clients from falling into poor performance while without degrading others, e.g., the green curve.

Intuitively, we consider this phenomenon oftentimes comes from the biased universal information towards the clients with better learning performance. For scheme I, a simple-averaged aggregation may not perfectly handle data heterogeneity, as it generates serious bias between the global and local clients. For scheme II, abandoning the universal contribution may dismiss some information from other clients. Instead of designing a new pFL algorithm, we propose a novel pFL strategy on existing pFL studies: generalizing the universal learning for unbiased local adaptation as well as keeping the local personalized features, called Personalize Locally, Generalize Universally (PLGU). The main challenge of PLGU is to generalize universal information without local feature perturbation, as the statistical information is only stored locally. In this paper, we tackle this challenge by developing a fine-grained perturbation method called Layer-Wised Sharpness-Aware-Minimization (LWSAM) based on the SAM optimizer [7, 33], which develops a generalized training paradigm by leveraging linear approximation. Furthermore, we present how to embed this PLGU strategy with the perturbed universal generalization on both the two pFL schemes.

For scheme I (with the global model), we propose the PLGU-Layer Freezing (LF) algorithm. As illustrated in [21, 31, 48], each layer in a personalized model shares a different contribution: the shallow layers focus more on local feature extraction (personalization), and the deeper layers are for extracting global features (universal). Specifically, the PLGU-LF first explores the personalization score of each layer. Then, PLGU-LF freezes the important layer locally for personalization and uses the LWSAM optimizer with the consideration of obtained layer importance score for universal generalization. For scheme II (without the global model), we mainly focus on our proposed PLGU strategy FedRep algorithm [3], named PLGU-GRep. It generalizes the universal information by smoothing personalization in the representation part. To show the extensibility, we present that we can successfully extend our PLGU strategy to pFedHN [37], called PLGU-GHN, to improve learning performance, especially for poor clients. Furthermore, we analyze the generalization bound on PLGU-LF, PLGU-GRep, and PLGU-GHN algorithms in-depth. Extensive experimental results also show that all three algorithms successfully prevent poor clients and outperform the average learning performance while incrementally reducing the top-performance clients.

2. Related Work

Various algorithms to realize pFL can be classified by different measurements. From the universal learning perspective, we can divide the existing pFL algorithms into two schemes: with or without a global shared model [19, 41]. Typically, algorithms on the scheme I (with a global model) are mainly extended from the conventional FL methods, i.e., FedAvg [29] or FedProx [24], which combines the adaption of local personalized features on local training updates procedure, such as fine-tuning [1, 30], regularized loss function [23, 40], model mixture [2, 4, 28, 32], and meta-learning [5, 14].

Scheme II of pFL, i.e., without a global model, has more diverse algorithms. [11, 28] propose to train multiple global models at the server, where similar clients are clustered into several groups and different models are trained for each group. However, these algorithms may incur huge communication costs. In addition, some algorithms collaboratively train the customized model with only layer-wised transfer universally to enhance the personalization [27, 37]. Specifically, other algorithms address heterogeneity in pFL by sharing some data [52] or generating additional data [13] on the server, which may violate the privacy policy [41].

The generalization of deep neural networks has been studied as an important topic, which can avoid the learning model to overfit the training dataset. Previous algorithms usually add auxiliary changes to the standard training process, e.g., dropout [39], and normalization [12, 45], which require the acknowledgment of training data that are not feasible in pFL. More specifically, solving the minimax objective will incur large computational costs. Recently, some studies in centralized learning [17, 51] observe that generalization is positively correlated to the sharpness of the training loss landscape. Motivated by this, [7] develops Sharpness-Aware Minimization (SAM), which uses approximated weight perturbation to leverage generalization by leveraging the first-order Taylor expansion. Moreover, [15, 33] extend SAM to FL and graph neural network.

3. Problem Formulation and Strategy Design

3.1. Problem Formulation

The goal of pFL is to collaboratively train personalized models on multiple clients by only sharing the model information while preserving the private local data. Generally, we conclude the pFL into schemes: scheme I (with global model) [2, 3, 25, 31] and scheme II (without global model) [27, 37]. Let $\mathcal{N}$ be the set of clients with the size of N, where the non-IID distributed training data on i-th client is denoted as $\mathcal{D}_i = \{(x_i, y_i)\}, i \in \mathcal{N}$, x_i, y_i are the corresponding data pair. For scheme I, let $\boldsymbol{\theta}_i$ denote the personalized model of client i, the objective of both schemes pFL can be formulated as follows:

$$\min_{\boldsymbol{w}, \{\boldsymbol{\theta}_i\}_{i=1}^N} \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{w}; \boldsymbol{\theta}_i), \tag{1}$$

where F_i is the loss function of the client i associated with its dataset $\mathcal{D}_i$, typically represented by the cross-entropy loss F_{CE} between the predicted and true label as $F_i(\boldsymbol{\theta}_i) = \frac{1}{m_i}\sum_{j=1}^{m_i} F_{\text{CE}}(\boldsymbol{\theta}_i; x_i^j, y_i^j)$, and m_i is the number of data samples. Note that the loss function $F_i(\boldsymbol{w}; \boldsymbol{\theta}_i)$ denotes that the pFL objective takes either the global model $\boldsymbol{w}$ or personalized model $\boldsymbol{\theta}_i$ as the target model for classification tasks. Note that the objective function on scheme II can be formulated as $\min_{\{\boldsymbol{\theta}_i\}_{i=1}^N} \frac{1}{N}\sum_{i=1}^N F_i(\boldsymbol{\theta}_i)$.

3.2. Personalize Locally, Generalized Universally

Although existing pFL studies have great accomplishments by enhancing personalization [2,3,25,27,31,37], they cannot guarantee that all clients can achieve desired learning performance. Especially, as shown in Figure 1, only focusing on personalization can lead to a biased pFL result, where poor clients that perform much lower learning performance suffer from the large client deviation. This phenomenon happens because the universal information shared across all clients may be not general enough or biased towards some typical clients. Although common sense is that the header of the neural network stores the personalization and other layers obtain the universal information [3, 31], only a few explore the importance of universal information on personalization and show how to moderate the personalization in the universal information. Therefore, in this paper, we aim to propose a pFL strategy to address this issue and prevent poor clients, called Personalize Locally, Generalize Universally (PLGU). Specifically, PLGU has two main parts: (i) extracting the features towards personalization and keeping them locally and (ii) generalizing the universal information.

To generalize the universal information, we leverage the Sharpness Aware Minimization (SAM) algorithm [7] to be the local optimizer. In SAM, the parameters of $\boldsymbol{w}_i$ whose neighbors within the ℓ_p ball are perturbed for a low training loss $F_{\mathcal{D}_i}$ through the following objective function:

$$F_{\mathcal{D}_i}(\tilde{\boldsymbol{\theta}}_i) = \max_{\|\boldsymbol{\epsilon}_i\|_p \le \rho} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i), \tag{2}$$

where $p \ge 0$, ρ is the radius of the ℓ_p ball, $\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i$, and $F_{\mathcal{D}_i}(\tilde{\boldsymbol{\theta}}_i)$ is the loss function of SAM on client i. Considering the non-trivial effort to calculate the optimal solution for the inner maximization, SAM uses one extra back-forward gradient ascent step to approximate $\tilde{\boldsymbol{\epsilon}}_i$:

$$\tilde{\boldsymbol{\epsilon}}_i = \rho \frac{\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)}{\|\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)\|} \approx \operatorname*{arg\,max}_{\|\boldsymbol{\epsilon}_i\|_p \le \rho} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i). \tag{3}$$

As such, SAM computes the perturbed model $\boldsymbol{\theta}_i + \tilde{\epsilon}_i$ for the gradient in objective (2) as $\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\tilde{\boldsymbol{\theta}}_i) \approx \nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)|_{\boldsymbol{\theta}_i + \tilde{\boldsymbol{\epsilon}}_i}$. However, SAM adds perturbation on the entire local model [7, 26, 33], which dismisses the interior impact of universal information. To generalize the fine-grained universal information, we develop a Layer-Wise SAM (LWSAM) to be the local training optimizer inspired by [26, 49]. Instead of simply applying the scaling to guide the training update [26], we leverage the inner maximization in LWSAM for the layer-wised scaling based on the property of all clients. Let $\boldsymbol{\Lambda}_i$ denote a diagonal $L \times L$ matrix, $\boldsymbol{\Lambda}_i = \text{diag}(\xi_{i,1}, \dots, \xi_{i,L})$, where $\xi_{i,l}$ is the layer personalization score. We apply the adopted scaling method to the inner maximization of SAM on client i as follows:

Algorithm 1 PLGU($\tilde{\boldsymbol{\theta}}_i^{t,0}, \tilde{\boldsymbol{w}}_i^{t,0}, \boldsymbol{\Lambda}_i^t, K, \eta$).

1: **Input:** personalized model $\tilde{\boldsymbol{\theta}}_i^{t,0}$, global model $\tilde{\boldsymbol{w}}_i^{t,0}$, scaling matrix $\boldsymbol{\Lambda}_i^t$, number of local epochs K, learning rate η;
2: **for** $k = 0, \dots, K-1$ **do**
3:     Sample mini-batch $\mathcal{B}_i$ on client i;
4:     Calculate unbiased gradient $\boldsymbol{g}_i^{t,k} = \nabla_{\tilde{\theta}_i^{t,k}} F_{\mathcal{B}_i}(\tilde{\boldsymbol{\theta}}_i^{t,k})$;
5:     Update personalized model $\tilde{\boldsymbol{\theta}}_i^{t,k+1} = \tilde{\boldsymbol{\theta}}_i^{t,k} - \eta \boldsymbol{g}_i^{t,k}$;
6:     Calculate perturbation $\tilde{\boldsymbol{\epsilon}}_i^{t,k}$ by (5);
7:     Calculate unbiased gradient approximation for LWSAM $\tilde{\boldsymbol{g}}_i^{t,k} = \nabla_{\tilde{\boldsymbol{w}}_i^{t,k} + \tilde{\boldsymbol{\epsilon}}_i^{t,k}} F_{\mathcal{B}_i}(\tilde{\boldsymbol{\theta}}_i^{t,k} + \tilde{\boldsymbol{\epsilon}}_i^{t,k})$;
8:     Update global model $\tilde{\boldsymbol{w}}_i^{t,k+1} = \tilde{\boldsymbol{w}}_i^{t,k} - \eta \tilde{\boldsymbol{g}}_i^{t,k}$;
9: **end for**

$$F_{\mathcal{D}_i}(\tilde{\boldsymbol{\theta}}_i) = \max_{\|\boldsymbol{\Lambda}_i \boldsymbol{\epsilon}_i\| \le \rho} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i + \boldsymbol{\Lambda}_i \boldsymbol{\epsilon}_i). \tag{4}$$

Note that the advantage of LWSAM can obtain fine-grained universal information, because it adds more perturbation to the personalized layers, i.e., a higher value of $\xi_{i,l}$ scales more perturbation and moderates the personalization from the global model perspective. The layer-wised weight perturbation in LWSAM is also solved by the first-order approximation of (4). Considering the added $\boldsymbol{\Lambda}_i$, the approximate inner solution of LWSAM can be written as follows:

$$\tilde{\boldsymbol{\epsilon}}_i = \rho \text{sign}(\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)) \boldsymbol{\Lambda}_i \frac{|\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)|^{q-1}}{(\|\nabla_{\boldsymbol{\theta}_i} F_{\mathcal{D}_i}(\boldsymbol{\theta}_i)\|_q^q)^{p-1}}, \tag{5}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. (5) provides the layer-wise calculation of $\tilde{\boldsymbol{\epsilon}}_i$ to scale up the batch size on client i. Specifically, the PLUG strategy based on the LWSAM algorithm is illustrated in Algorithm 1. In SAM, the first SGD step is only to calculate the perturbation. But the LWSAM algorithm can efficiently leverage the two output models. In particular, Lines 4-5 can obtain the personalized model, which aim to seek the optimal point of the model parameter $\tilde{\boldsymbol{\theta}}_i^{t,k}$. Lines 6-8 aim to seek the loss land surface $\tilde{\boldsymbol{w}}_i^{t,k}$. As such, both the two SGD steps obtain the models $\tilde{\boldsymbol{\theta}}_i^t$ and $\tilde{\boldsymbol{w}}^t$ for our learning goal. In the following two sections, we will present how to calculate the layer personalization score $\boldsymbol{\Lambda}_i$ and how to embed our proposed PLGU strategy into the two schemes separately.

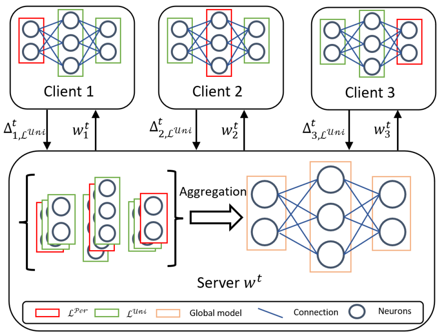


Figure 2. Illustration of PLGU-LF algorithm.

4. PLGU for Scheme I

4.1. PLGU-LF Algorithm

Although we also consider the pFL scheme I [23, 25, 31, 40], which includes a global model $\boldsymbol{w}$ and N personalized local models $\boldsymbol{w}_i$ at the same time, we should carefully design a more general global model $\boldsymbol{w}$ while not influencing the personalization, i.e., both global and personalized models can prevent poor clients. For extracting personalization features, we propose a distance metric for the l-th layer between the corresponding global model and local models to determine the personalization degree of each layer on the local model. The personalization score $\xi_{i,l}^t$ of the scaling matrix $\boldsymbol{\Lambda_i}$ can be calculated as follows:

$$\xi_{i,l}^t = \frac{\|\tilde{\boldsymbol{\theta}}_{i,l}^t - \tilde{\boldsymbol{w}}_l^t\|}{\text{dim}(\tilde{\boldsymbol{w}}_l^t)}, \tag{6}$$

where $\tilde{\boldsymbol{\theta}}_{i,l}^t$ and $\tilde{\boldsymbol{w}}_l^t$ are the model parameters at the l-th layer of the personalized model $\tilde{\boldsymbol{\theta}}_i^t$ and the global model $\tilde{\boldsymbol{w}}^t$. $\text{dim}(\cdot)$ denotes the number of parameters on layer l, which can normalize the values as $\sum_{l=1}^L \xi_{i,l} = 1$. Note that leveraging (6) to extract the personalization information does not incur huge computational costs compared to inference networks-based feature extraction [1, 25]. In addition, calculating the personalization score $\boldsymbol{\Lambda_i}$ in (6) uses simple linear algebra, and thus the main computational cost of the PLGU strategy comes from two steps SGD in LWSAM.

In this consideration, $\xi_{i,l}^t$ measures the l-th layer difference between the personalized model $\tilde{\boldsymbol{\theta}}_i^t$ and the global model $\tilde{\boldsymbol{w}}^t$, which quantifies the personalized contributions of i-th local client to the l-th layer on the global model at the communication round t. Intuitively, a larger value of ξ_l^i indicates that the l-th layer of $\tilde{\boldsymbol{\theta}}_i^t$ deviates further from $\tilde{\boldsymbol{w}}^t$, which indicates more contributions to personalization. On the contrary, a layer with a smaller value of ξ_l^i has more contributions to the universal information.

Hence, based on the personalization score $\xi_{i,l}^t$, we can divide the layers of $\tilde{\boldsymbol{\theta}}_i^t$ into two categories, where the number of D largest values of $\xi_{i,l}^t$ are categorized into the set of personalized layers $l \in \mathcal{L}_i^{\text{Per}}$, and others are into the set of universal layers $l \in \mathcal{L}_i^{\text{Uni}}$. To protect the personalization locally, when client i receives the global model $\tilde{\boldsymbol{w}}^t$, it will replace the universal layers in the $\boldsymbol{w}_{\mathcal{L}^{\text{Uni}}}^t$ and freeze its personalized layers to obtain a new personalized model $\tilde{\boldsymbol{\theta}}_i^{t,0}$, i.e., $\tilde{\boldsymbol{\theta}}_{i,\mathcal{L}^{\text{Uni}}}^{t,0} = \tilde{\boldsymbol{w}}_{\mathcal{L}^{\text{Uni}}}^t$ and $\tilde{\boldsymbol{w}}_{i,\mathcal{L}^{\text{Per}}}^{t,0} = \tilde{\boldsymbol{w}}_{i,\mathcal{L}^{\text{Per}}}^t$. We call this algorithm for the scheme I as PLGU-Layer Freezing (PLGU-LF). Thus, the objective of PLGU-LF for the scheme I is:

Algorithm 2 Scheme I: PLGU-LF algorithm.

Input: communication upper bound T, client set $\mathcal{N}$, number of local epochs K, learning rate η;
Output: personalized model $\tilde{\boldsymbol{w}}_i^T$ and global model $\tilde{\boldsymbol{w}}^T$;
for $t = 0, \ldots, T-1$ **do**
  Sample a set of clients $\mathcal{C}^t \subseteq \mathcal{N}$ with the size of C;
  for each client $i \in \mathcal{C}^t$ in parallel **do**
    Calculate $\boldsymbol{\Lambda}_i^t$ by (6);
    Select D layers with largest ξ_i^t values to be the set as $\mathcal{L}_{i,\text{Per}}^t$ and other layers are set as $\mathcal{L}_{i,\text{Uni}}^t$;
    $\tilde{\boldsymbol{\theta}}_{i,\mathcal{L}^{\text{Uni}}}^{t,0} = \tilde{\boldsymbol{w}}_{\mathcal{L}^{\text{Uni}}}^t$ and $\tilde{\boldsymbol{\theta}}_{i,\mathcal{L}^{\text{Per}}}^{t,0} = \tilde{\boldsymbol{\theta}}_{i,\mathcal{L}^{\text{Per}}}^t$
    PLGU($\tilde{\boldsymbol{\theta}}_i^{t,0}, \tilde{\boldsymbol{w}}_i^{t,0}, \boldsymbol{\Lambda}_i^t, K, \eta$)
    $\Delta_i^t = \tilde{\boldsymbol{w}}_i^{t,K} - \tilde{\boldsymbol{w}}_i^{t,0}$;
    $\tilde{\boldsymbol{w}}^{t+1} = \tilde{\boldsymbol{w}}^{t+1} + \frac{1}{C}\sum_{i \in \mathcal{C}^t} \Delta_i^t$;
  end for
end for

$$\min_{\boldsymbol{w}, \{\boldsymbol{\theta}_i\}_{i=1}^N} \max_{\{\|\boldsymbol{\Lambda}_i \boldsymbol{\epsilon}_i\| \le \rho\}_{i=1}^N} \frac{1}{N}\sum_{i=1}^N F_i(\tilde{\boldsymbol{w}}; \tilde{\boldsymbol{\theta}}_i), \tag{7}$$

where $\tilde{\boldsymbol{w}} = \boldsymbol{w} + \tilde{\boldsymbol{\epsilon}}_i$, $\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \hat{\boldsymbol{\epsilon}}_i$, and $\hat{\boldsymbol{\epsilon}}_{i,l} = \boldsymbol{0}, \forall l \in \mathcal{L}_i^{\text{Uni}}$.

We show the learning framework of PLGU-LF with 3 clients to learn a personalized 3-layered network in Figure 2, where each client freezes one personalized layer for personalization, i.e., $D = 1$. In Algorithm 2, we introduce the training procedure of PLGU-LF in detail. As a result, the global model should be more suitable compared to simply using SAM to be the local optimizer, because PLGU-LF algorithm does not lose much universal information sharing across all clients. From the personalization perspective, each client can receive more generalized universal information as well as keep its personal features in order to improve the performance of poor clients. Note that if $|\mathcal{L}^{\text{Per}}| = 0$, PLGU-LF is equal to FedSAM [33]; otherwise, i.e., $|\mathcal{L}^{\text{Per}}| = L$, it is the same as FedAvg [29]. Specifically, suppose that the number of local training epochs of $\tilde{\boldsymbol{\theta}}_i^t$ is equal to $\tilde{\boldsymbol{w}}_i^t$, the computational cost of PLGU-LF is equal to state-of-the-art pFL in scheme I [23, 40]. Thus, our proposed PLGU-LF does not incur more computational cost.

4.2. Generalization Analysis of PLGU-LF

In this subsection, we aim to analyze the generalization bound of the PLGU-LF algorithm, which can be presented as follows. Firstly, we define the generalization gap of PLGU-LF algorithm as $\left|\sum_{i=1}^{N}\frac{m_i}{m}\left(\min_{\boldsymbol{w}_i,i\in[N]}\max_{\|\boldsymbol{\epsilon}_i\|_2\le\rho,i\in[N]}F_i(\tilde{\boldsymbol{w}};\tilde{\boldsymbol{w}}_i)-\min_{\boldsymbol{w},\boldsymbol{w}_i,i\in[N]}\max_{\|\boldsymbol{\epsilon}_i\|_2\le\rho,i\in[N]}\bar{F}_i(\tilde{\boldsymbol{w}};\tilde{\boldsymbol{w}}_i)\right)\right|$, where $F_i(\cdot;\cdot)=\mathbb{E}_{P_i}[F_{\text{CE}}(\tilde{\boldsymbol{w}};\tilde{\boldsymbol{w}}_i),\boldsymbol{x},y]$, P_i denotes the local data distribution of client i and $\bar{F}_i(\cdot;\cdot)=\frac{1}{m_i}\sum_{j=1}^{m_i}[F_{\text{CE}}(\tilde{\boldsymbol{w}};\tilde{\boldsymbol{w}}_i),\boldsymbol{x}_j,y_j]$ denotes the empirical distribution. The following theorem aims to bound the difference between the empirical and underlying margin-based error for a general deep neural network function based on [8, 28, 43, 46]. The bound is based on the spectral norms of the model parameter matrices across layers which provide the upper bound for the Lipschitz and smoothness coefficients of the corresponding neural network.

Theorem 1. *Suppose that the loss function F is β-Lipschitz and the input data $\boldsymbol{x}$ has ℓ_2-norm bounded by B. For depth-L and width-d neural networks, suppose that the model parameter matrices in each of the L layers have spectrum norm bounded by M_l. Then, $\forall\gamma\in(0,1)$, with probability at least $1-\gamma$, we can upper bound the following generalization gap as* $\mathcal{O}\Bigg(\beta\bigg(\frac{B\sqrt{L}\prod_{l\in\mathcal{L}}M_l}{\gamma\sqrt{m}}+\frac{(B+\rho)d\sqrt{L\log L}\prod_{l\in\mathcal{L}}M_l}{\gamma\sqrt{m}}+\frac{B\sqrt{L}\prod_{l\in\mathcal{L}}M_l}{\gamma N\sqrt{\hat{m}}}+\frac{dD(L-D)\sqrt{\log D}\prod_{l\in\mathcal{L}^{Per}}BM_l\sqrt{\log(L-D)}\prod_{l\in\mathcal{L}^{Uni}}(B+\rho)M_l}{\gamma N\sqrt{\hat{m}}}\bigg)+\sqrt{m\log\frac{1}{\gamma}}\Bigg)$*, where $\hat{m}=\min m_i,\forall i\in[N]$.*

Different from the generalization bound of existing pFL algorithms with two main items [4, 6, 28], the result in Theorem 1 has four main items. The additional items come from the perturbation. The first two items are from the global model, which depends on the total number of data samples m. The third and fourth items are based on the personalized model, and hence they depend on the local dataset and the number of clients N. In addition, the first and third items are due to the marginal-based error [35]. And the second and fourth terms are because of the perturbation (adversarial error), which depends on the number of perturbed layers, i.e., $L-D$. The adversarial error of the global model is on all the layers, and the personalized model depends on the number of universal layers.

5. PLGU for Scheme II

5.1. PLGU-GRep

In this section, we focus on instantiating the PLGU strategy on scheme II without obtaining a global model. Existing studies for scheme II aim to share the learned universal information (usually not the full local models) across all clients, e.g., cluster [42], multi-task learning [38], and [37] hyper-network. However, none of studies discuss whether the learned universal information is general enough or not. FedRep [3] is one of the most popular pFL scheme II, which is based on representation learning. The motivation of FedRep is that even if local datasets are highly heterogeneous, we can still seek to share the common low-dimensional universal feature representation across all clients. In the main paper, we will set our PLGU strategy on FedRep to improve the generalization of the universal representation and prevent the poor clients, named PLGU-GRep. To show the extensibility, we will embed our PLGU strategy on pFedHN [37], named pFed-GHN, and present the detailed training procedure and generalization analysis in supplementary.

Let $\boldsymbol{\theta}_{\boldsymbol{\phi}}$ denote the global representation, i.e., universal information, which is a function parameterized by $\boldsymbol{\phi}\in\Phi$, and the specific heads $\boldsymbol{\theta}_{\boldsymbol{h}_i}$, which are parameterized by $\boldsymbol{h}_i\in\mathcal{H},\forall i\in\mathcal{N}$. Specifically, the personalized model of client i can be decomposed by the low-dimensional header and representation $\boldsymbol{\theta}_i=(\boldsymbol{\theta}_{h_i}\circ\boldsymbol{\theta}_{\boldsymbol{\phi}})$. Therefore, the objective for FedRep can be formulated as follows:

$$\min_{\boldsymbol{\phi}\in\Phi}\frac{1}{N}\sum_{i=1}^{N}\min_{h_i\in\mathcal{H}}F_i(\boldsymbol{h}_i,\boldsymbol{\phi}),\tag{8}$$

where the function $F_i(h_i,\boldsymbol{\phi}):=F_i(\boldsymbol{\theta}_{h_i}\circ\boldsymbol{\theta}_{\boldsymbol{\phi}})$. Although the success of straightforwardly leveraging representation to pFL has been demonstrated in [3, 44], they do not consider which layers are more dominant on the universal information in Φ. As such, FedRep cannot guarantee that all clients achieve the desired accuracy due to the biased representation $\boldsymbol{\phi}$ towards part of clients. Specifically, as shown in the toy example in Figure 1, 22% of clients cannot achieve 72% accuracy using the FedRep algorithm. Therefore, we aim to seek a more generalized representation $\tilde{\boldsymbol{\phi}}^t$ to prevent more clients from falling into poor performance, and the objective function of (8) is re-formulated as $\min_{\tilde{\boldsymbol{\phi}}\in\Phi}\frac{1}{N}\sum_{i=1}^{N}\min_{h_i\in\mathcal{H}}F_i(\boldsymbol{h}_i,\tilde{\boldsymbol{\phi}})$. Note that the header h_i mainly obtains the personalization for client i, we can update it by K epochs SGD as follows:

$$\boldsymbol{g}_i^{t,k}=\nabla_{\boldsymbol{h}_i^{t,k}}F_{\mathcal{B}_i}(\boldsymbol{h}_i^{t,k},\tilde{\boldsymbol{\phi}}^t),\;\;\boldsymbol{h}_i^{t,k+1}=\boldsymbol{h}_i^{t,k}-\eta\boldsymbol{g}_i^{t,k}\tag{9}$$

When the local header $\boldsymbol{h}_i$ updates finish, PLGU-GRep comes into the representation $\tilde{\boldsymbol{\phi}}$ updates phase. Our design goal is to generalize some specific layers towards personalization. As such, we propose a two-step update to achieve this. Firstly, similar to PLGU-LF, we also explore the personalization score $\boldsymbol{\Lambda}_i^t=\text{diag}(\xi_{i,1}^t,\ldots,\xi_{i,L^{\boldsymbol{\phi}}}^t),\forall l\in\mathcal{L}^{\boldsymbol{\phi}_i}$, where $\mathcal{L}^{\boldsymbol{\phi}_i}$ is the layers set of $\boldsymbol{\phi}_i$ with the size of $L^{\boldsymbol{\phi}}$ on client i at communication round t, to determine the personalization contribution, i.e.,

$$\xi_{i,l}^t=\frac{\|\tilde{\boldsymbol{\phi}}_{i,l}^t-\tilde{\boldsymbol{\phi}}_l^t\|}{\dim(\tilde{\boldsymbol{\phi}}_l^t)},\tag{10}$$

Then, by leveraging PLGU strategy in Algorithm 1, we can calculate the layer-wised perturbation $\tilde{\boldsymbol{\epsilon}}_i^t$ and obtain a more

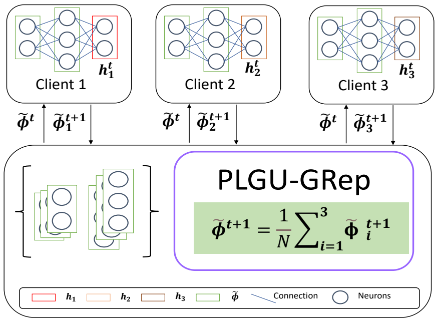


Figure 3. Illustration of PLGU-GRep.

generalized $\tilde{\boldsymbol{\phi}}_i^{t+1}$ representation on client i at communication t as follows:

$$\boldsymbol{g}_i^t =\nabla_{\tilde{\boldsymbol{\phi}}^t} F_{\mathcal{B}_i}(\boldsymbol{h}_i^{t,K}, \tilde{\boldsymbol{\phi}}^t), \;\; \boldsymbol{\phi}_i^t = \tilde{\boldsymbol{\phi}}^t - \eta \boldsymbol{g}_i^t, \tag{11}$$

$$\tilde{\boldsymbol{\epsilon}}_i^t =\rho \mathrm{sign}(\nabla_{\boldsymbol{\phi}_i^t} F_{\mathcal{B}_i}(\boldsymbol{\phi}_i^t))\boldsymbol{\Lambda}_i^t \frac{|\nabla_{\boldsymbol{\phi}_i^t} F_{\mathcal{B}_i}(\boldsymbol{\phi}_i^t)|^{q-1}}{(\|\nabla_{\boldsymbol{\phi}_i^t} F_{\mathcal{B}_i}(\boldsymbol{\phi}_i^t)\|_q^q)^{p-1}}, \tag{12}$$

$$\tilde{\boldsymbol{g}}^t =\nabla_{\boldsymbol{\phi}_i^t} F_{\mathcal{B}_i}(\boldsymbol{h}_i^{t,K}, \boldsymbol{\phi}_i^t + \tilde{\boldsymbol{\epsilon}}_i^t), \;\; \tilde{\boldsymbol{\phi}}_i^t = \boldsymbol{\phi}_i^t - \eta \boldsymbol{g}_i^t. \tag{13}$$

Based on (10)-(13), we can moderate the personalization in the representation by adding more perturbation. The description of our proposed PLGU-GRep is shown in Figure 3 and its detailed training procedure is illustrated in Algorithm 3. Note that we only use one more SGD step to update the $\tilde{\boldsymbol{\phi}}_i^t$, and hence PLGU-GRep does not incur a huge computational cost.

5.2. Generalization Analysis for PLGU-GRep

Here, we demonstrate the generalization bound of the PLGU-GRep algorithm. Suppose that there exists a global model $F(\boldsymbol{h}, \tilde{\boldsymbol{\phi}})$, and then obtain the bound by Rademacher complexity. The empirical loss of the global model is $\hat{F}_{\mathcal{D}}(\boldsymbol{h}, \tilde{\boldsymbol{\phi}}) = \frac{1}{N}\sum_{i=1}^N \frac{m_i}{m}\sum_{j=1}^{m_i} F_{\mathrm{CE}}(\boldsymbol{x}_j^i, y_j^i; \boldsymbol{h}_i, \tilde{\boldsymbol{\phi}})$. For the expected loss of $F(\boldsymbol{V}, \tilde{\boldsymbol{\phi}})$, $F(\boldsymbol{h}, \tilde{\boldsymbol{\phi}}) = \frac{1}{N}\sum_{i=1}^N \frac{m_i}{m}\mathbb{E}_{P_i}[F_{\mathrm{CE}}(\boldsymbol{x}, y; \boldsymbol{h}_i, \tilde{\boldsymbol{\phi}})]$. We assume that the function $F(\cdot)$, $\boldsymbol{h}_i, \forall i \in [N]$ and $\tilde{\boldsymbol{\phi}}$ are β-, β_h- and β_ϕ-Lipschitz.

Theorem 2. *We assume that the local training model is a depth-L and width-d neural network, the input data sample is bounded by B, the model parameter matrices in each of the layers have spectrum norm bounded by $M_l, \forall l \in \mathcal{L}^\phi$, and the model parameter matrices of header $\boldsymbol{h}$ can be bounded by M_h. $\forall \gamma \in (0,1)$ we can bound the generalization gap of PLGU-GRep with probability at least $1-\gamma$ as* $\mathcal{O}\Bigg(\beta\Bigg(\frac{\beta_\phi (B+\rho) d\sqrt{(L-1)\log(L-1)}\prod_{l\in\mathcal{L}^\phi} M_l}{\gamma\sqrt{m}} + \frac{\beta_h B\sqrt{L}M_h}{\gamma N\sqrt{\hat{m}}} + \frac{\beta_\phi (B+\rho) d\sqrt{(L-1)\log(L-1)}\prod_{l\in\mathcal{L}^\phi} M_l}{\gamma N\sqrt{\hat{m}}}\Bigg) + \sqrt{m\log\frac{1}{\gamma}}\Bigg)$, *where* $\hat{m} = \min m_i, \forall i \in [N]$.

Algorithm 3 Scheme II: PLGU-GRep algorithm.

Input: communication upper bound T, client set $\mathcal{N}$, number of header local epochs K, learning rate η;
Output: personalized model $\tilde{\boldsymbol{\theta}}_i^T$;
for $t = 0, \ldots, T-1$ **do**
  Sample a set of clients $\mathcal{C}^t \subseteq \mathcal{N}$ with the size of C;
  Download global representation $\tilde{\boldsymbol{\phi}}^t$ to client $i \in \mathcal{C}^t$;
  for each client $i \in \mathcal{C}^t$ in parallel **do**
    for $k = 0, \ldots, K-1$ **do**
      Sample mini-batch $\mathcal{B}_i \subset \mathcal{D}_i$;
      Update the header $\boldsymbol{h}_i^{t,k}$ by (9);
    end for
    Sample mini-batch $\mathcal{B}_i \subset \mathcal{D}_i$;
    Update the generalized representation $\tilde{\boldsymbol{\phi}}_i^t$ by (10)-(13);
  end for
  Server updates the new representation $\tilde{\boldsymbol{\phi}}^{t+1} = \frac{1}{C}\sum_{i\in\mathcal{C}^t}\tilde{\boldsymbol{\phi}}_i^t$;
end for

Theorem 2 shows insights into the effect of perturbation for PLGU-GRep, and indicates the generalization gap. The first item is based on the aggregation of local generalized representation $\tilde{\boldsymbol{\phi}}_i$, which depends on the number of total data samples m. Based on the local training by SGD on the header $\boldsymbol{h}_i$ and generalized local representation $\tilde{\boldsymbol{\phi}}_i$, the second and third items depend on the number of data samples on each client $\hat{m}$ and the number of clients N. More specifically, because the third item is related to LWSAM, i.e., with perturbation, it includes the value of ρ, compared to the second item.

6. Experiments

6.1. Experimental Setups

We use CIFAR10, CIFAR100 [18], and Tiny-ImageNet (TmgNet) [20] datasets with various heterogeneous levels and participation rates. We set up 100 clients for CIFAR10 and CIFAR100 datasets, and 20 clients for TmgNet dataset. For the non-IID dataset, we simulate three non-IID data simulations by assigning a fixed number of labels. Note that the default number of labels on CIFAR10 is 3, CIFAR100 is 10, and on TmgNet is 30. The number of data samples on each client is uniformly distributed. We set the number of local epochs $K = 5$, a number of universal layers $|\mathcal{L}^{\mathrm{Uni}}| = 5$ and local batch size as 50, and the perturbation $\rho = 0.05$ by default. We consider two participation schemes $C = 10$ and 100 for CIFAR10 and CIFAR100, respectively.

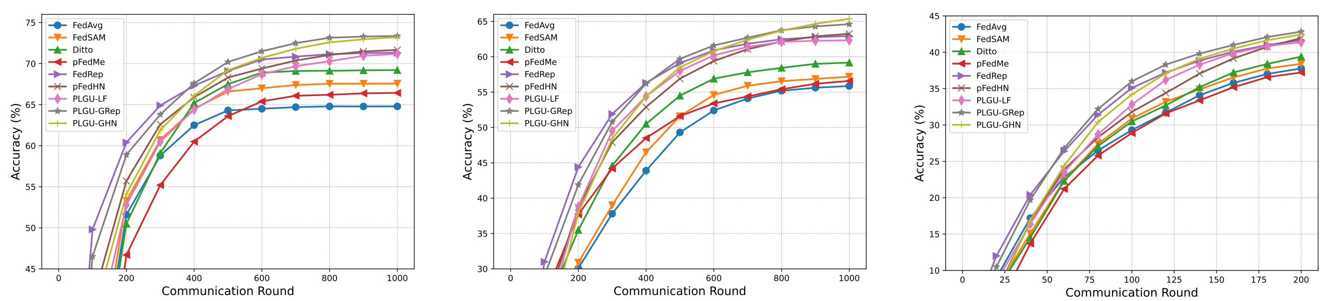
We compare PLGU-LF, PLGU-GRep, and PLGU-GHN algorithms with several state-of-the-art FL and pFL benchmarks: FedAvg [29], FedSAM [33], Ditto [23], pFedMe

Table 1. Testing accuracy by the global model (averaged, top 5%, and lowest 5% accuracy) under three datasets.

Datasets	C	FedAvg	FedSAM	Ditto	pFedMe	PLGU-LF
CIFAR10	10	64.79	67.57	64.15	64.36	69.36
		60.69 68.45	64.76 69.93	62.08 69.45	61.13 70.46	67.34 71.69
	100	67.15	69.49	68.24	68.31	71.48
		65.99 71.80	67.04 71.12	61.64 71.08	65.25 71.64	68.92 73.57
CIFAR100	10	55.86	57.19	56.35	55.17	59.74
		51.21 60.75	54.82 59.55	52.73 59.28	50.41 58.35	56.85 61.49
	100	58.00	59.39	58.93	57.24	61.07
		55.16 60.67	56.73 61.20	56.64 60.79	52.96 60.53	58.25 62.25
TmgNet	20	37.78	38.42	38.09	36.43	40.61
		32.26 43.73	34.61 42.02	34.75 42.84	31.79 42.90	35.41 43.56

Table 2. Testing accuracy by the personalized model (averaged, top 5%, and lowest 5% accuracy) under three datasets.

Datasets	C	Ditto	pFedMe	FedRep	pFedHN	PLGU-LF	PLGU-GRep	PLGU-GHN
CIFAR10	10	69.21	66.43	71.32	71.66	71.18	72.87	72.64
		65.33 72.86	62.97 71.10	68.15 74.04	68.06 73.95	68.22 73.27	69.98 74.91	69.70 74.63
	100	71.23	69.19	75.30	74.95	74.23	76.94	76.17
		69.93 74.46	66.84 71.58	73.11 77.74	73.23 77.58	72.66 75.87	74.23 77.61	73.79 77.82
CIFAR100	10	59.17	56.58	62.92	63.25	62.33	64.61	65.37
		57.81 63.06	52.92 60.14	59.70 65.95	59.86 65.79	60.08 65.24	62.58 66.62	63.02 66.94
	100	62.52	58.57	65.08	65.54	64.67	66.79	66.30
		59.48 64.81	55.79 61.73	62.60 67.96	63.30 68.09	63.72 67.75	64.81 68.59	64.27 68.02
TmgNet	20	39.41	37.22	41.68	41.96	41.39	42.84	42.45
		35.88 43.49	32.76 42.91	37.53 45.07	37.94 45.19	37.46 44.57	40.29 45.18	39.92 44.89



(a) CIFAR10.

(b) CIFAR100.

(c) TmgNet.

Figure 4. Convergence results evaluation of personalized models under three datasets.

[40], FedRep [3], and pFedHN [37]. To clearly show the performance of our proposed algorithms on different learning models, we use ResNet-18 [10], WideResNet28-10 [50], and ResNet-50 [10] for CIFAR10, CIFAR100, and TmgNet, with group batch. The HN includes 3 fully connected layers. More experimental setups and results will be presented in the supplementary.

6.2. Basic Performance Evaluations

We first evaluate the performance of the proposed algorithms by analyzing the achieved model accuracy on pFL. The results in Table 1 indicate that, compared to other benchmarks on all three datasets, the proposed PLGU-LF algorithm reaches the best global model accuracy (increasing at least 2.55% than others) and does not degrade the top clients. More importantly, we successfully prevent the clients from falling into poor performance, where the 5% lowest clients can achieve 56.85% accuracy with $C = 10$, and increase accuracy by at least 2.03% for the lowest 5% clients on CIFAR100. And the results of the personalized model accuracy evaluation are shown in Table 2, where the proposed algorithms outperform others, e.g., PLGU-GRep achieves 72.64%, 69.70%, and 74.63%, on average, top 5%, and lowest 5% on CIFAR10 with $C = 10$.

To show the learning performance from the convergence perspective, we present the convergence curves in Figure 4. It is easy to observe that the proposed PLGU-GRep and PLGU-GHN achieve the best two convergence speeds.

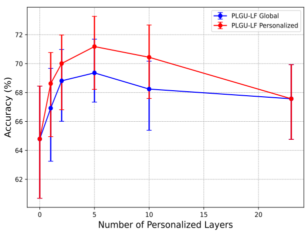


Figure 5. Impact of the size of the personalized layer set D.

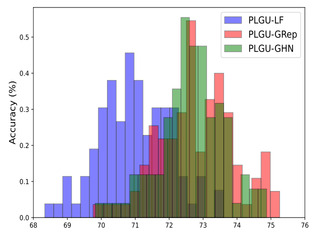
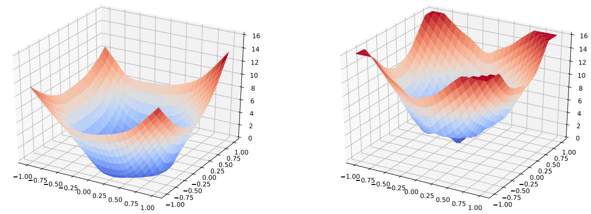


Figure 6. Distribution of personalized models.



(a) Global model. (b) Personalized model

Figure 7. Loss landscapes visualization of PLGU-LF.

Table 3. Impact of number of local epochs K with $C = 10$.

K	1		5		10	
PLGU-LF (G)	65.31		69.36		68.42	
	62.08	68.20	67.24	71.13	66.23	71.39
PLGU-LF (P)	67.43		71.18		70.40	
	64.25	70.01	68.22	73.17	65.76	73.54
PLGU-GRep	70.82		72.87		71.65	
	67.25	73.28	69.98	74.91	68.04	74.11
PLGU-GHN	69.93		72.64		72.91	
	66.89	73.15	69.70	74.63	72.91	75.48

Table 4. Impact of perturbation ρ with $C = 10$.

ρ	0.05		0.1		0.5	
PLGU-LF (G)	69.36		68.62		66.73	
	67.24	71.13	65.71	71.88	63.95	68.31
PLGU-LF (P)	71.18		70.96		69.05	
	68.22	73.17	69.15	74.68	66.89	73.50
PLGU-GRep	72.87		72.01		70.80	
	69.98	74.91	71.49	75.93	66.73	73.65
PLGU-GHN	72.64		70.49		67.30	
	69.70	74.63	68.18	74.73	64.84	74.69

6.3. Further Performance Evaluations

Due to the space limitation, we leverage the CIFAR10 for the ablation performance study for all the proposed algorithms. Figure 5 studies the impact of the empirical number of $\mathcal{L}^{\text{Per}}$. Through observation, we can notice that when $D = 5$, the proposed PLGU-LF algorithm achieves the best performance in both personalized and global models. More specifically, when $D = 1$ or 10, the performance does not have obvious degradation. However, when $D = 0$ or 23, it does not achieve the desired performance, which indicates that no perturbation, e.g., FedAvg, and, e.g., full layers perturbation, e.g., FedSAM, are not efficient solutions.

In Figure 6, we show the results of personalized model distribution across all clients under PLGU-LF, -GHN, and -GRep. Compared to the toy example in Figure 1, we can see that our proposed algorithms can significantly decrease the deviation and prevent more poor clients (the accuracy of all clients is larger than 75%), while not clearly reducing the top clients.

We investigate the impact of local epoch number K on the performance in Table 4, which achieve the best when $K = 5$. Note that we use "G" and "P" to represent the global and personal model performance of PLGU-LF. We then explore the impact of ρ on LWSAM in Table 3. The best performance is to set $\rho = 0.05$. In addition, when we increase $\rho = 0.5$, the learning performance incurs large degradation, which matches the results in [7, 26]. Therefore, it is necessary to properly set the values of K and ρ to achieve better performance.

Lastly, to visualize the universal generalization ability of the proposed PLGU-LF algorithm, we show the loss surfaces for the global and personalized models, following the settings in [22]. The results show that the global model is more smooth, i.e., generalizing more universal information, and the personalized model is sharper, i.e., protecting model personalization on clients.

7. Conclusion

In this paper, we propose a novel PLGU strategy to prevent clients from falling into poor performance without obviously downgrading the average and top personalized performance. This strategy aims to generalize the universal information while protecting the personalized features locally. Specifically, we embed the PLGU strategy on two pFL schemes and propose three algorithms, PLGU-LF, PLGU-GRep, and PLGU-GHN by keeping the personalization local and generalizing universal information. Further theoretical investigation indicates that all the PLGU-based algorithms can achieve competitive generalization bounds. The extensive experimental results show that all the proposed algorithms can successfully protect poor clients while not degrading the average learning performance.

Acknowledgement. This work is supported in part by the Natural Science Foundation of Changsha (kq2202108), the Natural Science Foundation of Hunan Province (2022JJ40636), and the National Natural Science Foundation of China (62202293). We are grateful for resources from the High Performance Computing Center of Central South University.

References

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 2, 4

[2] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2022. 2, 3

[3] 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. 1, 2, 3, 5, 7

[4] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 2, 5

[5] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020. 1, 2

[6] Farzan Farnia, Amirhossein Reisizadeh, Ramtin Pedarsani, and Ali Jadbabaie. An optimal transport approach to personalized federated learning. *arXiv preprint arXiv:2206.02468*, 2022. 5

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 2, 3, 8

[8] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018. 5

[9] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[11] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI*, pages 7865–7873, 2021. 1, 2

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 2

[13] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018. 2

[14] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 2

[15] Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. Questions for flat-minima optimization of modern neural networks. *arXiv preprint arXiv:2202.00661*, 2022. 2

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 1

[17] Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 2

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[19] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797. IEEE, 2020. 2

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[21] Sunwoo Lee, Tuo Zhang, Chaoyang He, and Salman Avestimehr. Layer-wise adaptive model aggregation for scalable federated learning. *arXiv preprint arXiv:2110.10302*, 2021. 2

[22] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 8

[23] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 1, 2, 4, 6

[24] 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. 1, 2

[25] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1, 2, 3, 4

[26] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12360–12370, 2022. 3, 8

[27] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10092–10101, 2022. 1, 2, 3

[28] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for per-

sonalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. 1, 2, 5

[29] 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. 1, 2, 4, 6

[30] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019. 2

[31] Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2021. 2, 3, 4

[32] Zhe Qu, Rui Duan, Lixxing Chen, Jie Xu, Zhuo Lu, and Yao Liu. Context-aware online client selection for hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 2022. 2

[33] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. *arXiv preprint arXiv:2206.02618*, 2022. 2, 3, 4, 6

[34] Zhe Qu, Xingyu Li, Jie Xu, Bo Tang, Zhuo Lu, and Yao Liu. On the convergence of multi-server federated learning with overlapping area. *IEEE Transactions on Mobile Computing*, 2022. 1

[35] Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. In *NeurIPS*, 2020. 5

[36] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020. 1

[37] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021. 1, 2, 3, 5, 7

[38] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017. 5

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[40] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 1, 2, 4, 7

[41] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1, 2

[42] Xueyang Tang, Song Guo, and Jingcai Guo. Personalized federated learning with clustered generalization. *arXiv preprint arXiv:2106.13044*, 2021. 5

[43] Yu-Lin Tsai, Chia-Yi Hsu, Chia-Mu Yu, and Pin-Yu Chen. Formalizing generalization and adversarial robustness of neural networks to weight perturbations. *Advances in Neural Information Processing Systems*, 34:19692–19704, 2021. 5

[44] Isidoros Tziotis, Zebang Shen, Ramtin Pedarsani, Hamed Hassani, and Aryan Mokhtari. Straggler-resilient personalized federated learning. *arXiv preprint arXiv:2206.02078*, 2022. 5

[45] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[46] Jiancong Xiao, Yanbo Fan, Ruoyu Sun, and Zhi-Quan Luo. Adversarial rademacher complexity of deep neural networks. 2021. 5

[47] Peng Xiao, Samuel Cheng, Vladimir Stankovic, and Dejan Vukobratovic. Averaging is probably not the optimum way of aggregating parameters in federated learning. *Entropy*, 22(3):314, 2020. 1

[48] Rudong Xu, Yiting Tao, Zhongyuan Lu, and Yanfei Zhong. Attention-mechanism-containing neural networks for high-resolution remote sensing image classification. *Remote Sensing*, 10(10):1602, 2018. 2

[49] Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6(12):6, 2017. 3

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 7

[51] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 2

[52] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34:10092–10104, 2021. 2