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

Zhe Qu\textsuperscript{1}, Xingyu Li\textsuperscript{2*}, Xiao Han\textsuperscript{3}, Rui Duan\textsuperscript{3}, Chengchao Shen\textsuperscript{1}, and Lixing Chen\textsuperscript{4}

\textsuperscript{1}Central South University, \textsuperscript{2}Mississippi State University, \textsuperscript{3}University of South Florida, \textsuperscript{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

\*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 \( N \) be the set of clients with the size of \( N \), where the non-IID distributed training data on \( i \)-th client is denoted as \( D_i = \{ (x_i, y_i) \}, i \in N, \) \( x_i, y_i \) are the corresponding data pair. For scheme I, let \( \theta_i \) denote the personalized model of client \( i \), the objective of both schemes pFL can be formulated as follows:

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

### 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, Generalized 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 $\hat{w}_i$ whose neighbors within the $\ell_p$ ball are perturbed for a low training loss $F_{D_i}$ through the following objective function:

$$F_{D_i}(\hat{\theta}_i) = \max_{\|\epsilon_i\|_p \leq \rho} F_{D_i}(\theta_i + \epsilon_i), \quad (2)$$

where $p \geq 0$, $\rho$ is the radius of the $\ell_p$ ball, $\hat{\theta}_i = \theta_i + \epsilon_i$, and $F_{D_i}(\hat{\theta}_i)$ 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 $\epsilon_i$:

$$\hat{\epsilon}_i = \rho \frac{\nabla \theta_i F_{D_i}(\hat{\theta}_i)}{\|\nabla \theta_i F_{D_i}(\hat{\theta}_i)\|} \approx \arg \max_{\|\epsilon_i\|_p \leq \rho} F_{D_i}(\theta_i + \epsilon_i). \quad (3)$$

As such, SAM computes the perturbed model $\theta_i + \hat{\epsilon}_i$ for the gradient in objective (2) as $\nabla \theta_i F_{D_i}(\hat{\theta}_i) \approx \nabla \theta_i F_{D_i}(\theta_i) |_{\theta_i + \hat{\epsilon}_i}$. 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 $\Lambda_i$ denote a diagonal $L \times L$ matrix, $\Lambda_i = \text{diag}(\xi_{i,1}, \ldots, \xi_{i,L})$, 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:

$$F_{D_i}(\hat{\theta}_i) = \max_{\|\Lambda_i \epsilon_i\| \leq \rho} F_{D_i}(\theta_i + \Lambda_i \epsilon_i). \quad (4)$$

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 $\Lambda_i$, the approximate inner solution of LWSAM can be written as follows:

$$\hat{\epsilon}_i = \rho \text{sign}(\nabla \theta_i F_{D_i}(\hat{\theta}_i)) \Lambda_i \frac{\|\nabla \theta_i F_{D_i}(\hat{\theta}_i)\|-1}{\|\nabla \theta_i F_{D_i}(\hat{\theta}_i)\|_p^{-1}}, \quad (5)$$

where $\frac{1}{p} + \frac{1}{q} = 1$. (5) provides the layer-wise calculation of $\hat{\epsilon}_i$ 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 $\hat{\theta}_i$. Lines 6-8 aim to seek the loss land surface $\hat{w}_i$. As such, both the two SGD steps obtain the models $\hat{\theta}_i$ and $\hat{w}_i$ for our learning goal. In the following two sections, we will present how to calculate the layer personalization score $\Lambda_i$ and how to embed our proposed PLGU strategy into the two schemes separately.

**Algorithm 1** PLGU($\hat{\theta}_i$, $\hat{w}_i$, $\Lambda_i$, $K$, $\eta$).

1. **Input:** personalized model $\theta_i$, global model $\hat{w}_i$, scaling matrix $\Lambda_i$, number of local epochs $K$, learning rate $\eta$.
2. for $k = 0, \ldots, K - 1$
   3. Sample mini-batch $B_i$ on client $i$;
   4. Calculate unbiased gradient $\hat{g}_{i,k} = \nabla \hat{g}_{i,k} F_{B_i}(\hat{\theta}_i)$;
   5. Update personalized model $\theta_{i,k+1} = \theta_i - \eta \hat{g}_{i,k}$;
   6. Calculate perturbation $\hat{\epsilon}_i$ by (5);
   7. Calculate unbiased gradient approximation for LWSAM $\hat{g}_{i,k} = \nabla \hat{g}_{i,k} + \eta \hat{g}_{i,k}$;
   8. Update global model $\hat{w}_{i,k+1} = \hat{w}_{i,k} - \eta \hat{g}_{i,k}$;
9. end for.
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 \(w\) and \(N\) personalized local models \(w_i\) at the same time, we should carefully design a more general global model \(w\) 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 of the personalized model \(\tilde{w}_l\) where \(\theta\) are the model parameters at the \(l\)-th layer on the global model at \(t\) and \(\tilde{w}_l\) are categorized into the set \(\mathcal{L}_l\). To protect the personalization locally, when client \(i\) receives the global model \(\tilde{w}_t\), it will replace the universal layers in the \(\tilde{w}_t\) and freeze its personalized layers to obtain a new personalized model \(\hat{\theta}_i^{t,0}\), i.e., \(\hat{\theta}_i^{t,0}_{Uni} = \tilde{w}_t^{Uni} \) and \(\hat{\theta}_i^{t,0}_{Per} = \tilde{w}_t^{Per}\). 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:

\[
\min_{\mathcal{L}_l^{Per}} \max_{\mathcal{L}_l^{Uni}} \frac{1}{N} \sum_{i=1}^{N} F_i(\tilde{w}; \hat{\theta}_i),
\]

where \(\tilde{w} = w + \epsilon_i, \hat{\theta}_i = \tilde{\theta}_i + \epsilon_i, \) and \(\epsilon_i, 1 = 0, \forall l \in \mathcal{L}_l^{Uni}\). 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 \(|\mathcal{L}_l^{Per}| = 0\), PLGU-LF is equal to FedSAM \([33]\); otherwise, i.e., \(|\mathcal{L}_l^{Per}| = L\), it is the same as FedAvg \([29]\). Specifically, suppose that the number of local training epochs of \(\hat{\theta}_i^{t,0}\) is equal to \(w_t\), 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

---

### Algorithm 2 Scheme I: PLGU-LF Algorithm

1. **Input:** communication upper bound \(T\), client set \(\mathcal{C}\), number of local epochs \(K\), learning rate \(\eta\);
2. **Output:** personalized model \(\tilde{w}_t^{Per}\) and global model \(\tilde{w}_t^{Uni}\);
3. **for** \(t = 0, \ldots, T - 1\) **do**
4. Sample a set of clients \(\mathcal{C}_t^{\tilde{C}} \subseteq \mathcal{C}\) with the size of \(\tilde{C}\);
5. **for** each client \(i \in \mathcal{C}_t^{\tilde{C}}\) in parallel **do**
6. Calculate \(\Lambda_i^t\) by (6);
7. Select \(D\) layers with largest \(\xi_i^t\) values to be the set as \(\tilde{\mathcal{L}}_i^{Per}\) and other layers are set as \(\tilde{\mathcal{L}}_i^{Uni}\);
8. \(\tilde{\theta}_i^{t,0}_{Uni} = \tilde{w}_t^{Uni}\) and \(\tilde{\theta}_i^{t,0}_{Per} = \tilde{\theta}_i^{t,0}_{Per}\);
9. PLGU(\(\tilde{\theta}_i^{t,0}_{Uni}, \tilde{w}_t^{Uni}, \Lambda_i^t, K, \eta\));
10. \(\Delta_i^t = \tilde{w}_t^{Per} - \tilde{w}_t^{Uni}\);
11. \(\tilde{w}_t^{t+1} = \tilde{w}_t^{t+1} + \frac{1}{\tilde{C}} \sum_{i \in \mathcal{C}_t^{\tilde{C}}} \Delta_i^t\);
12. **end for**
13. **end for**

---
can be presented as follows. Firstly, we define the generalization gap of PLGU-LF algorithm as
\[
\frac{1}{m} \left( \min_{w, \bar{w}} \max_{i \in [N]} \max_{\|l \|_2 \leq m, \rho, i \in [N]} F_i(w; \bar{w}) - \min_{w, \bar{w}} \max_{i \in [N]} \max_{\|l \|_2 \leq m, \rho, i \in [N]} F_i(w; \bar{w}) \right),
\]
where
\[
F_i(\cdot; \cdot) = \mathbb{E}_{P_i}[F_T(w; \bar{w}), x, y],
\]
\(P_i\) denotes the local data distribution of client \(i\) and \(F_i(\cdot; \cdot)\) 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 \(\beta\)-Lipschitz and the input data \(x\) has \(\ell_2\)-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
\[
O\left( \beta \frac{B \sqrt{\log M_l}}{\gamma^2} \frac{1}{\sqrt{m}} \right) + \left( B + \rho \right) \sqrt{\frac{\log L}{m}} \frac{1}{\gamma^2} \frac{1}{\sqrt{m}} + \left( B + \rho \right) \frac{1}{\gamma^2} \frac{1}{\sqrt{m}} + \left( B + \rho \right) \frac{1}{\gamma^2} \frac{1}{\sqrt{m}} + \left( B + \rho \right) \frac{1}{\gamma^2} \frac{1}{\sqrt{m}}
\]
where \(m = \min m_i, \forall i \in [N]\).

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 due to 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 \(\theta_\phi\) denote the global representation, i.e., universal information, which is a function parameterized by \(\phi \in \Phi\), and the specific heads \(\theta_{h_i}\), which are parameterized by \(h_i \in \mathcal{H}, \forall i \in N\). Specifically, the personalized model of client \(i\) can be decomposed by the low-dimensional header and representation \(\theta_i = (\theta_{h_i} \circ \theta_\phi)\). Therefore, the objective for FedRep can be formulated as follows:
\[
\min_{\phi \in \Phi} \frac{1}{N} \sum_{i=1}^{N} \min_{h_i \in \mathcal{H}} F_i(h_i, \phi),
\]
where the function \(F_i(h_i, \phi) := F_i(\theta_{h_i} \circ \theta_\phi\). 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 \(\Phi\). As such, FedRep cannot guarantee that all clients achieve the desired accuracy due to the biased representation \(\phi\) 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 \(\phi'\) to prevent more clients from falling into poor performance, and the objective function of (8) is re-formulated as follows:
\[
\min_{\phi \in \Phi} \frac{1}{N} \sum_{i=1}^{N} \min_{h_i \in \mathcal{H}} F_i(h_i, \phi). \quad (9)
\]

When the local header \(h_i\) updates finish, PLGU-GRep comes into the representation \(\phi\) 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 \(\Lambda_i^t = \text{diag}(\xi_i^{t, 1}, \ldots, \xi_i^{t, L_\phi}), \forall i \in \mathcal{L}^\phi_t\), where \(\mathcal{L}^\phi_t\) is the layers set of \(\phi_i\) with the size of \(L_\phi\) on client \(i\) at communication round \(t\), to determine the personalization contribution, i.e.,
\[
\xi_i^{t, l} = \frac{\|\hat{\phi}_i^{t, l} - \phi_i^t\|}{\text{dim}(\phi_i^t)},
\]

Then, by leveraging PLGU strategy in Algorithm 1, we can calculate the layer-wised perturbation \(\xi_i^{t, l}\) and obtain a more
generalized $\tilde{\phi}_t^{i+1}$ representation on client $i$ at communication $t$ as follows:

$$g_t^i = \nabla_\tilde{\phi}_t F_{\theta_t}(h_t^i, \tilde{\phi}_t), \quad \phi_t^i = \tilde{\phi}_t - \eta g_t^i,$$

(11)

$$\epsilon_t^i = \rho \text{sign}(\nabla_\phi_t F_{\theta_t}(\phi_t^i)) \Lambda_t^i \left(\frac{\|\nabla_\phi_t F_{\theta_t}(\phi_t^i)\|_q}{p-1}\right),$$

(12)

$$\hat{g}_t^i = \nabla_\phi_t F_{\theta_t}(h_t^i, \phi_t^i + \epsilon_t^i), \quad \hat{\phi}_t^i = \phi_t^i - \eta \hat{g}_t^i.$$  

(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 $\tilde{\phi}_t^i$, 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 $F(h, \phi)$, and then obtain the bound by Rademacher complexity. The empirical loss of the global model is $F_{\theta}(h, \phi) = \frac{1}{N} \sum_{i=1}^{N} m_i \sum_{t=1}^{m_i} F_{\theta}(x_t^i, y_t^i; h, \phi)$.

For the expected loss of $F_{\theta}(\cdot, \phi)$, $F_{\theta}(h, \phi) = \frac{1}{N} \sum_{i=1}^{N} \frac{m_i}{m} \mathbb{E}_{P_i} [F_{\theta}(x, y; h, \phi)]$.

We assume that the function $F(\cdot)$, $h_i, \forall i \in [N]$ and $\phi$ are $\beta_\phi$, $\beta_{\gamma^0}$- and $\beta_\phi$-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 $M_h$, $\forall h \in \mathcal{L}^h$, and the model parameter matrices of the local model $h$ 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

$$O\left(\beta (\beta_{\phi}(B + \rho)\sqrt{L - 1}) \frac{\log(L - 1)}{\gamma^0} \frac{\prod_{l \in \mathcal{L}^h} M_l}{\gamma^0} + \frac{\beta_{\phi}(B + \rho)\sqrt{L - 1}) \frac{\log(L - 1)}{\gamma^0} \frac{\prod_{l \in \mathcal{L}^h} M_l}{\gamma^0} + \frac{\beta_{\phi}(B + \rho)\sqrt{L - 1}) \frac{\log(L - 1)}{\gamma^0} \frac{\prod_{l \in \mathcal{L}^h} M_l}{\gamma^0} + \sqrt{m \log \frac{1}{\gamma}}\right),$$

where

$m = \min m_i, \forall i \in [N].$"
The results in Table 1 indicate that, compared to other algorithms by analyzing the achieved model accuracy on pFL. 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 $C = 10$. 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.
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 $L_{\text{per}}$. 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 $\rho$ 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 $\rho$ 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


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

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agura and Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. PMLR, 2017. 1, 2, 4, 6


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


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


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


