Sparse Federated Learning With Hierarchical Personalization Models

Xiaofeng Liu, Qing Wang, Member, IEEE, Yunfeng Shao, Member, IEEE, and Yinchuan Li, Member, IEEE

Abstract—Federated learning (FL) can achieve privacy-safe and reliable collaborative training without collecting users’ private data. Its excellent privacy security potential promotes a wide range of FL applications in Internet of Things (IoT), wireless networks, mobile devices, autonomous vehicles, and cloud medical treatment. However, the FL method suffers from poor model performance on non-independent and identically distributed (non-i.i.d.) data and excessive traffic volume. To this end, we propose a personalized FL algorithm using a hierarchical proximal mapping based on the moreau envelop, named sparse federated learning with hierarchical personalized models (sFedHP), which significantly improves the global model performance facing diverse data. A continuously differentiable approximated ℓ1-norm is also used as the sparse constraint to reduce the communication cost. Convergence analysis shows that sFedHP’s convergence rate is state-of-the-art with linear speedup and the sparse constraint only reduces the convergence rate to a small extent while significantly reducing the communication cost. Experimentally, we demonstrate the benefits of sFedHP compared with the federated averaging (FedAvg), hierarchical fedavg (HierFAVG), and personalized FL methods based on local customization, including FedAMP, FedProx, per-FedAvg, pFedMe, and pFedGP.

Index Terms—Cloud computing, federated learning (FL), machine learning, non-independent and identically distributed (non-i.i.d.) data, privacy preservation.

I. INTRODUCTION

MACHINE learning methods have proliferated in real-life applications thanks to the tremendous number of labeled samples [1]. Typically, these samples collected on users’ devices, such as mobile phones, are expected to send to a centralized server with mighty computing power to train a deep model [2]. However, users are often reluctant to share personal data due to privacy concerns, which motivates the emergence of federated learning (FL) [3]. Federated averaging (FedAvg) [3] is known as the first FL algorithm to build a global model (GM) for different clients while protecting their data locally. Moreover, FL has been used in the Internet of Things (IoT), wireless networks, mobile devices, autonomous vehicles, and cloud medical treatment for its excellent potential in privacy security [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Unfortunately, the distribution of local data stored in different clients varies greatly, and FedAvg performs unwise when meeting non-independent and identically distributed (non-i.i.d.) data. In particular, generalization errors of the GM increase significantly with the data’s statistical diversity increasing [14], [15]. To address this problem, personalized FL based on multitask learning [16] and personalization layers [17], and local customization [18], [19], [20], [21], [22] have been proposed. Most personalized FL algorithms prioritize the personalized model (PM) performance of individual clients, while overlooking the performance of the GM. However, this approach contradicts the fundamental purpose of FL, which aims to build a high-quality GM. Disregarding the GM’s performance may lead to a suboptimal model and hinder the inclusion of new clients. Additionally, frequent communication between clients and the server is typically necessary in FL to ensure convergence performance, which can be hampered by high latency and limited bandwidth.

To address these challenges, we introduce a novel hierarchical personalized FL framework. Our approach includes a personalized edge server that minimizes differences between models during GM aggregation, leading to significant improvements in GM performance and allowing for personalized user models. Furthermore, our hierarchical personalized FL architecture covers the client-edge-cloud, reducing direct communication between clients and the cloud server and greatly decreasing communication overhead.

A. Main Contributions

Our main contributions in this article are summarized as follows.

1) We propose a novel personalized FL framework, named sparse FL with hierarchical personalization models [sparse federated learning with hierarchical personalized models (sFedHP)]. Our approach employs a hierarchical proximal mapping technique based on the moreau envelop. This method separates the optimization of...
client and edge models from that of the GM, promoting personalized models for clients and edge servers while keeping them close to the reference model. As a result, our approach enhances the performance of the GM on non-i.i.d data.

2) The hierarchical architecture of sFedHP reduces direct communication between clients and the cloud server, leading to a significant decrease in communication overhead. The continuously differentiable approximated $\ell_1$-norm constraints in sFedHP with sparse version generate sparse models, which further reduce communication costs.

3) We present the convergence analysis of sFedHP by exploiting the convexity-preserving and smoothness-enabled properties of the loss function, which characterizes two notorious issues (client-sampling and client-drift errors) in FL [23]. With carefully tuned hyperparameters, theoretical analysis shows that sFedHP’s convergence rate is state-of-the-art with linear speedup.

4) We empirically evaluate the performance of sFedHP using different data sets that capture the statistical diversity of clients’ data. We show that sFedHP obtained the state-of-the-art performance while greatly reducing the number of parameters by 80%. Moreover, sFedHP with nonsparse version outperforms FedAvg, the hierarchical fedavg (HierFAVG) [24], and other local customization-based personalized FL methods [18], [20], [21], [22] in terms of GM.

B. Organization and Main Notation

The remainder of this article is organized as follow. Section II undertake a complete literature review to illustrate the existing research findings. The problem formulation and algorithm are formulated in Section III. Section IV shows the convergence analysis with some important lemmas and theorems. Section V presents the experimental results, followed by some analysis. Finally, conclusion and discussions are given in Section VI. The main notations used are listed in Table I.

Table I: Main Notations

| Definition                        | Notation |
|----------------------------------|----------|
| Number of edge servers           | $N$      |
| Number of clients for each edge server | $J$    |
| Global model of the cloud server | $w$      |
| Edge global model of i-th edge server | $w_i$ |
| Edge personalized model of i-th edge server | $\varphi_i$ |
| Local edge model of i-th edge server and j-th client | $\varphi_{i,j}$ |
| Local personalized model          | $\theta_{i,j}$ |
| Expected loss over the data of the i-th server and j-th client | $l_{i,j}$ |
| Training data of j-th client in i-th server | $z_{i,j}$ |
| Hyperparameter for sparsity       | $\gamma_1, \gamma_2$ |
| Hyperparameter for personalization | $\lambda_1, \lambda_2$ |
| Hyperparameter for aggregation    | $\beta$  |
| Global training round             | $T$      |
| Local training round              | $R$      |
| Number of edges to aggregate      | $S$      |
| Learning rate for updating $w_i$  | $\eta_1$ |
| Learning rate for updating $\theta_{i,j}$ | $\eta_2$ |
| Expectation function              | $E_1$    |
| Same-order infinitesimal function | $O(1)$   |

II. Related Work

Traditional machine learning methods require local training data to be uploaded to a central server for centralized training. However, in practical scenarios like training on mobile phone data, sensor data in the IoT, and data from external companies like banks, uploading local data poses a significant privacy risk. Thus, participants are often hesitant to expose their local data during training. To address these issues, McMahan et al. [3] proposed an FL framework and its model aggregation algorithm, FedAvg. They also developed a network protocol for FL in 2019 [25]. The framework involves multiple participants and a central server, where participants use their local data for training and upload model updates to a parameter server. The GM is obtained by aggregating these updates, enabling multiparty collaborative machine learning while protecting local data. FL has been successfully applied in various scenarios, including smart apps, such as Google keyboard, Siri speech classifier, and QuickType keyboard, as well as cross-silo applications, such as drug discovery, financial risk prediction, and smart manufacturing.

In FL, data is supposed to be independent and identically distributed, meaning each user’s mini-batch sample must be statistically the same as a sample drawn uniformly from the entire training data set of all users. However, due to differences in devices, participants, enterprises, and scenarios, statistical diversity often exists among users, resulting in non-i.i.d. data. Non-i.i.d. data includes skewed feature distribution, skewed label distribution, different features of the same label, and different labels of the same feature [26]. Recent research on non-i.i.d. data in FL has primarily focused on non-i.i.d. data in FL has primarily focused on skewed label distribution, where different users have different label distributions. To construct non-i.i.d. training samples, researchers partition existing “flat” data sets based on labels [27]. However, global models trained on non-i.i.d. data sets often struggle to generalize well [28]. To address this problem, various personalized FL methods have been proposed, including local customization-based methods [18], [19], [20], [21], [22], personalized layer-based methods [17], knowledge distillation-based methods [29], [30], multitask learning-based methods [31], and lifelong learning-based methods [32].

In particular, local customization-based methods fine-tune the local model to customize a PM. Fedprox [18] achieves good personalization through $\ell_2$-norm regularization. pFedMe [19] can decouple PM optimization from the GM learning in a bi-level problem stylized based on Moreau envelopes. per-FedAvg [21] sets up an initial meta-model that can be updated effectively after one more gradient descent step. FedAMP and HeurFedAMP [20] utilize the attentive message passing to facilitate more collaboration with similar customers. pFedGP [22] is proposed as a solution to PFL based on Gaussian processes with deep kernel learning. The proposed sFedHP is a local customization-based personalized FL method, so we compare the performance of sFedHP with other local customization-based personalized FL methods, including [18], [19], [20], [21], and [22].
Fig. 1. Sparse client-edge-cloud FL with hierarchical personalized models.

However, personalized FL algorithms often prioritize the performance of individual client models over the GM’s performance, which contradicts the fundamental objective of FL to develop a high-quality GM and hinder the inclusion of new clients. Additionally, due to the need for frequent communication between clients and the cloud server to ensure convergence performance, FL suffers from high latency and limited bandwidth. As a result, we propose sFedHP, a communication-friendly personalized FL scheme that produces a high-performance GM.

III. SPARSE FEDERATED LEARNING WITH HIERARCHICAL PERSONALIZATION MODELS (sFedHP)

A. sFedHP: Problem Formulation

Consider a client-edge-cloud framework with one cloud server, $N$ edge servers and $NJ$ clients, i.e., each edge server collects data from $J$ clients as shown in Fig. 1. In sFedHP, we aims to find a sparse GM $w$ based on local sparse personalized models $\theta_{i,j}$, $i = 1, \ldots, N$, $j = 1, \ldots, J$, which can greatly reduce the model size and communication costs, by minimizing

$$\text{sFedHP: } \min_{w \in \mathbb{R}^d} \left\{ F(w) \triangleq \frac{1}{N} \sum_{i=1}^{N} F_i(w) \right\}$$

with

$$F_i(w) \triangleq \frac{1}{J} \sum_{j=1}^{J} F_{i,j}(w)$$

$$F_{i,j}(w) = \min_{\phi_{i,j}, \theta_{i,j} \in \mathbb{R}^d} \ell_{i,j}(\theta_{i,j}) + \lambda_1 \| \theta_{i,j} - \phi_{i,j} \|_2^2 + \gamma_1 \phi_{p}(\theta_{i,j}) + \gamma_2 \phi_{p}(w)$$

where $\ell_{i,j}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ denotes the expected loss over the data distribution of the $i$th edge server and the $j$th client; $\lambda_1$ and $\lambda_2$ denoting regularization parameters that control the strength of $w$ to the PM; $\phi_{i,j}$ is the local edge model of the $i$th edge server and the $j$th client; $\gamma_1$ and $\gamma_2$ denoting weight factors that control the sparsity level; and $\phi_{p}(\cdot)$ being a twice continuously differentiable approximation for $\|x\|_1$ [33], which is given by

$$\phi_{p}(x) = \rho \sum_{n=1}^{d} \log \cosh \left( \frac{\|x\|_1}{\rho} \right)$$

(4)

where $x_n$ denotes the $n$th element in $x$ and $\rho$ is a weight parameter, which controls the smoothing level. Note that we use $\phi_{p}(\cdot)$ instead of $\| \cdot \|_1$ to exploit the sparsity in $w$ and $\theta$.

sFedHP utilizes the moreau envelope twice to partition the optimization problem into three stages, enabling independent optimization of the GM $w$, client models $\theta_{i,j}$, and edge server models $\phi_{i,j}$. The first application of the moreau envelope decouples the optimization of the GM from that of the edge server models, while the second separates the optimization of the client models from that of the edge server models. Leveraging the moreau envelope allows us to achieve personalized models for clients and edge servers while maintaining proximity to the reference model, resulting in enhanced performance on non-i.i.d data. To encourage sparsity in the GM $w$, we permit clients to discover their own sparse models $\theta_{i,j}$ using continuously differentiable approximated $\ell_1$-norm constraints within a reasonable distance from the reference point $w$.

In the first stage, the optimal local PM $\hat{\theta}_{i,j}$ is obtained by minimizing (3) w.r.t. the data distribution of the $i$th edge server and $j$th client in the third stage as the following:

$$\hat{\theta}_{i,j}(\phi_{i,j}) = \arg \min_{\theta_{i,j} \in \mathbb{R}^d} L_{i,j}(\theta_{i,j}) + \frac{\lambda_1}{2} \| \theta_{i,j} - \phi_{i,j} \|_2^2$$

(5)

where $L_{i,j}(\theta_{i,j}) \triangleq \ell_{i,j}(\theta_{i,j}) + \gamma_1 \phi_{p}(\theta_{i,j})$.

In the second stage, the edge PM $\phi_{i}$ and the local edge model $\phi_{i,j}$ are determined by minimizing (3) w.r.t. the client models of the $i$th edge server

$$\hat{\phi}_{i} = \frac{1}{J} \sum_{j=1}^{J} \hat{\phi}_{i,j}$$

(6)

$$\hat{\phi}_{i,j} = \arg \min_{\phi_{i,j} \in \mathbb{R}^d} \lambda_1 \frac{1}{2} \| \theta_{i,j} - \phi_{j} \|_2^2 + \frac{\lambda_2}{2} \| \phi_{i,j} - w \|_2^2$$

In the third stage, $w$ is determined by utilizing the sparse model aggregation from multiple edges.

Assumption 1 (Strong Convexity and Smoothness): Assume that $\ell(w) : \mathbb{R}^d \rightarrow \mathbb{R}$ is (a) $\mu$-strongly convex or (b) nonconvex and $L$-smooth on $\mathbb{R}^d$, then we, respectively, have the following inequalities:

$$\ell(w) \geq \ell(\tilde{w}) + \langle \nabla \ell(\tilde{w}), w - \tilde{w} \rangle + \frac{\mu}{2} \| w - \tilde{w} \|_2^2$$

(6a)

$$\| \nabla \ell(w) - \nabla \ell(\tilde{w}) \|_2 \leq L \| w - \tilde{w} \|_2$$

(6b)

Assumption 2 (Bounded Variance): The variance of stochastic gradients (sampling noise) in each client is bounded by

$$\mathbb{E}_Z \left[ \| \tilde{\ell}_{i,j}(w; Z_{i,j}) - \nabla \ell_{i,j}(w) \|_2^2 \right] \leq \gamma_2^2$$

where $\gamma_2$ denotes the variance of the stochastic gradient.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( Z_{i,j} \) is the training data randomly drawn from the distribution of client \( i \) and edge server \( j \).

**Assumption 3 (Bounded Diversify):** The diversity of client’s data distribution is bounded by

\[
\frac{1}{NJ} \sum_{i,j=1}^{N,J} \| \nabla \ell_{i,j}(w) - \nabla \ell(w) \|_2^2 \leq \sigma_{\ell}^2.
\]

While Assumption 1, 2, and 3 are widely used in FL gradient calculation and convergence analysis [19], [21], [23], [34].

**B. \( s\)FedHP: Algorithm**

The pseudocode for \( s\)FedHP is outlined in Algorithm 1. During the 4th communication round, the cloud server broadcasts the GM \( w^t \) to all edge servers. Subsequently, the edge servers and their clients conduct \( R \) rounds of iterative training and upload their edge GM \( w^t_j \) to aggregate a new GM \( w^{t+1} \). As previously mentioned, \( s\)FedHP decomposes the optimization process into three steps.

In the first level, the local PM \( \hat{\theta}_{i,j} \) of the \( i \)-th edge server and the \( j \)-th client is determined by solving (5), whose parameters are sparse and can reduce the communication load between clients and edge servers. Note that (5) can be easily solved by many first-order approaches, for example, Nesterov’s accelerated gradient descent, based on the gradient. However, to calculate the exact \( \nabla \ell_{i,j}(\theta_{i,j}) \) requires the distribution of \( Z_{i,j} \), hence we use the unbiased estimate by sampling a mini-batch of data \( D_{i,j} \)

\[
\nabla \ell_{i,j}(\theta_{i,j}, D_{i,j}) = \frac{1}{|D_{i,j}|} \sum_{Z_{i,j} \in D_{i,j}} \nabla \ell_{i,j}(\theta_{i,j}, Z_{i,j})
\]

such that \( \mathbb{E}[\nabla \ell_{i,j}(\theta_{i,j}, D_{i,j})] = \nabla \ell_{i,j}(\theta_{i,j}) \). Thus, we solve the following minimization problem instead of solving (5) to obtain an approximated local PM

\[
\tilde{\theta}_{i,j}(\phi^{r}_{i,j}) = \text{arg min}_{\theta_{i,j} \in \Theta} H(\theta; \phi^{r}_{i,j}, D_{i,j})
\]

where \( H(\theta_{i,j}; \phi^{r}_{i,j}, D_{i,j}) = \ell_{i,j}(\theta_{i,j}, D_{i,j}) + \gamma_1 \phi_\theta(\theta_{i,j}) + (\lambda_1/2) \| \theta_{i,j} - \tilde{\phi}_{i,j} \|_2^2 \).

Similarly, (8) can be solved by Nesterov’s accelerated gradient descent, we let the iteration goes until the condition

\[
\| \nabla H(\tilde{\theta}_{i,j}; \phi^{r}_{i,j}, D_{i,j}) \|_2^2 \leq \nu
\]

is reached, where \( \nu \) is an accuracy level.

In the second level, after clients’ local personalized models are updated, the local edge models are determined by

\[
\phi^{r+1}_{i,j} = \frac{\lambda_1 \tilde{\theta}^{r}_{i,j} + \lambda_2 w^{r}_{i,j}}{\lambda_1 + \lambda_2}
\]

where \( \tilde{\phi}_i = (1/J) \sum_{j=1}^J \tilde{\phi}_{i,j} \) is the current edge personal model of the 4th edge server.

In the third level, once edge personalized models are updated, the edge global models can be updated by stochastic gradient descent as follows:

\[
w^{r+1}_{i,j} = w^r_{i,j} - \eta \nabla F_i(w^r_{i,j}) + \gamma_2 \nabla \phi_\theta(w^r_{i,j})
\]

where \( \eta_1 \) is a learning rate and \( w^r_{i,j} \) is the edge GM of \( i \)-th edge server at the global round \( r \) and edge round \( r \). The GM \( w^{r+1} \) is determined by utilizing the edge GM \( w^{r+1}_{i,j} = w^r_{i,j} \) aggregation from multiple edges. Moreover, similarly with [19] and [23], an additional parameter \( \beta \) is used for GM update to improve the convergence performance.

Note that communication between the clients and edge servers is more efficient than between the clients and the cloud server. The latter’s communication is high cost and latency because the distance is relatively long.

**IV. CONVERGENCE ANALYSIS**

**A. Convergence Theorems**

In this section, we present the convergence or \( s\)FedHP. We first prove Theorem 1 under Assumptions 1–3, which presents the smoothness and strong convexity properties of \( F_{i,j}(w) \).

**Theorem 1:** If \( \ell_{i,j} \) is convex or nonconvex with L-Lipschitz \( \nabla \ell_{i,j} \), then \( F_{i,j} \) is \( L_F \)-smooth with \( L_F = \lambda_2 + (\gamma_2/\rho) \) (with the condition that \( \lambda_2 > 4L + (4\gamma_2/\rho) \) for nonconvex \( L \)-smooth \( \ell_{i,j} \)) and if \( \ell_{i,j} \) is \( L \)-smooth then \( F_{i,j} \) is \( L \)-strongly convex with \( L_F = (\lambda_1 L_\ell + \lambda_2 L + \lambda_1 \lambda_2) \).

**Proof:** We first prove some interesting character of \( \phi_\theta(x) \) in (4), which is convex smooth approximation to \( \| x \|_1 [33] \) and

\[
0 \leq \nabla^2 \phi_\theta(x) = \frac{1}{\rho} (1 - (\tanh(x/\rho))^2) \leq \frac{1}{\rho}
\]

which yields \( \nabla \phi_\theta(x) - \nabla \phi_\theta(x) \leq (1/\rho)(x - \hat{x}) \), then we have
Combining (12) and (13) yields

\[ L_{ij}(a \theta_{ij} + (1 - a) \tilde{\theta}_{ij}) \leq a \ell_{ij}(\theta_{ij}) + (1 - a) \ell_{ij}(\tilde{\theta}_{ij}) - \frac{\mu}{2} (1 - a) \| \theta_{ij} - \tilde{\theta}_{ij} \|_2^2, \quad (12) \]

In addition, by noting that \( \phi_\rho \) is convex function, we have

\[ \phi_\rho(a \theta_{ij} + (1 - a) \tilde{\theta}_{ij}) \leq a \phi_\rho(\theta_{ij}) + (1 - a) \phi_\rho(\tilde{\theta}_{ij}). \quad (13) \]

Using the character of \( \phi_\rho \) mentioned before, we have the similarly conclusion that if \( G_i \) is \( \mu_G \)-strongly convex, \( F_{ij} \) is \( \mu_- \)-strong convex with \( \mu_F = \mu_G = (\lambda_1 \lambda_2) / \lambda_1 \mu + \lambda_2 \mu + \lambda_1 \lambda_2 / 2 \), and if \( G_i \) is \( L_G \)-smooth, \( F_{ij} \) is \( L_F \)-smooth with \( L_F = L_G + (\gamma_2 / \rho) \). Finally, combining the first-order condition \( \lambda_1 (\dot{\gamma}_{ij} - \dot{\theta}_{ij}) + \lambda_2 (\dot{y}_{ij} - \bar{w}) = 0 \), we complete the proof.

For unique solution \( w^* \) to \( s_{FedHP} \), we have the following convergence theorems, Theorems 2 and 3, based on Assumptions 1–3 and Theorem 1.

**Theorem 2 (Strongly Convex s_{FedHP}'s Convergence):** Let Assumptions 1(a) and 2 hold, there exists an \( \tilde{n} \leq \min\{(1/1, \alpha, (1/3L_F + 128L_F / \mu_{\beta});)\} \), such that

\[ \frac{1}{T} \sum_{t=0}^{T-1} E \left[ \| F(w^t) - F(w^*) \|_2^2 \right] \leq \frac{\Delta F}{\eta T} + M_1 + \eta M_2 \]

where \( \Delta F \leq \| w_0 - w^* \|_2^2, M_1 = (8\delta^2 \lambda^2 / \mu_{_F}), M_2 = 6(1 + 64(L_F / \mu_{\beta})\sigma_{F_1}^2 + (128L_F \delta^2 \lambda^2 / \mu_{\beta}) \), \( \lambda \leq (\lambda_1 \lambda_2 / \sqrt{\lambda_1^2 + \lambda_2^2}), \lambda \leq (\lambda_1 \lambda_2 / \lambda_1 + \lambda_2) \) and \( D_1 = \sigma_{F_1}^2 / L_F + \gamma_2^2 d_i^2 / L_F + \delta^2 \) with \( \sigma_{F_1}^2 \leq (1/N) \sum_{j=1}^{N} \| V F_{ij}(w^*) \|_2^2 \), being the sampling noise and \( d_i \) denoting the number of nonzero elements in the GM \( w \).

**Theorem 3 (Nonconvex and Smooth s_{FedHP}'s Convergence):** Let Assumptions 1(b), 2, and 3 hold, there exists an \( \tilde{n} \leq \min\{(1/588L_F \lambda^2), (\beta^2 / 2L_F)\} \), with \( \lambda \leq \sqrt{16(L + (\gamma_1 / \rho)^2) + 1} \), such that

\[ \left( \frac{1}{T} \sum_{t=0}^{T-1} E \left[ \| F(w^t) - F(w^*) \|_2^2 \right] \right) \leq 4 \left( \frac{\Delta F}{\eta T} + G_1 + \eta G_2 \right) \]

\[ \left( \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} E \left[ \| \hat{\ell}_{ij}(w^*) - w^* \|_2^2 \right] \right) \leq \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} E \left[ \| \hat{\ell}_{ij}(w^*) - w^* \|_2^2 \right] + \mathcal{O} (D_2) \]

where \( \Delta_F \leq F(w_0) - F^*, G_1 = 3 \beta^2 \lambda^2 (2^2, G_2 = (3L_F (49R \sigma_{F_1}^2 + 16 \delta^2 \lambda^2 / \lambda)^2) / R ) \) and \( D_2 = \sigma_{F_1}^2 / \lambda + \gamma_2^2 d_i^2 / \lambda + \delta^2 \) with \( \sigma_{F_1}^2 \leq (1/N) \sum_{j=1}^{N} \| V F_{ij}(w^*) \|_2^2 \), being the sampling noise and \( d_i \) denoting the number of nonzero elements in the GM \( w \).
results show that a good sparsity can be obtained by only using a tiny $\gamma_2$, yields $\frac{2}{\lambda_1}d_i^2 / \bar{x}_i^2 \ll \{\sigma_{F_1}^2 / \lambda_2, \sigma_{F_2}^2 / \lambda_2\}$, i.e., the convergence speed cost of the sparse constraints can be omitted. Note that, according to our experiments, limiting the learning rate does not increase the training time. This technique is both useful and common in FL analysis. Similar to the works in [19] and [23], we have applied this technique to our theorem analysis. The proofs of Theorems 2 and 3 are presented in Appendix.

B. Some Important Lemmas

In this section, we present some important lemmas, which are well used in the proof of Theorems 2 and 3, to help better understand the conclusion of the theorems. We re-write the local update in (11) as follows:

$$w_i^t = w_i^{t-1} - \eta_1 \lambda_2 \sum_{r=0}^{R-1} \left[ \left( w_i^{r,t} - \bar{y}_i^{r,t} \right) + \gamma_2 \nabla \phi_p \left( w_i^{r,t} \right) \right],$$

which implies

$$\eta_1 \sum_{r=0}^{R-1} g_i^{r,t} = \sum_{r=0}^{R-1} \left( w_i^{r,t} - w_i^{r,t+1} \right) = w_i^t - w_i^{t,R},$$

where $g_i^{r,t}$ can be interpreted as the biased estimate of $\nabla F_i(w_i^{r,t})$ since $E(g_i^{r,t}) \neq \nabla F_i(w_i^{r,t})$. Then, we rewrite the global update as

$$w_i^{t+1} = w_i^t - \eta_1 \beta R \frac{1}{SR} \sum_{r=0}^{R-1} \sum_{i=0}^{n} g_i^{r,t},$$

where $\bar{\eta}$ and $\bar{g}$ can be, respectively, considered as the step size and approximate stochastic gradient of the global update, which cause drift error in one-step update of the GM formulated in Lemma 1.

Lemma 1 (One-Step Global Update): Let Assumption 1(b) holds. We have

$$E \left[ \left\| w_i^{t+1} - w_i^t \right\|_2^2 \right] \leq E \left[ \left\| w_i^t - w_i^* \right\|_2^2 \right] - \bar{\eta} (2 - 6L_\beta \bar{\eta}) E \left[ F(w_i^t) - F(w_i^*) \right] + \frac{\bar{\eta} (3\bar{\eta} + 1) \beta R}{N^2} \sum_{i=0}^{N} E \left[ \left\| g_i^{r,t} - \nabla F_i(w_i^t) \right\|_2^2 \right] + 2\bar{\eta}^2 \frac{1}{N^2} \sum_{i=0}^{N} E \left[ \left\| \nabla F_i(w_i^t) - \nabla F(w_i^t) \right\|_2^2 \right]^2,$$

where $\bar{\eta} (3\bar{\eta} + 1) \beta R$ is required for when $A.1(1)(b)$ holds and $A.1(1)(a)$ holds.

V. EXPERIMENTAL RESULTS

A. Performance Comparison

To empirically highlight the performance of the proposed method, we first compare FL algorithms in nonsparse setting with FedAvg [3], HierFLAV [24] and local customization personalized FL methods, including Fedprox [18], per-FedAvg [21], pFedMe [19], HuerFedAMP [20], and pFedGP [22]. Based on MNIST [35], fashion-MNIST (FMNIST) [36], and CIFAR-10 [37] data sets. For non-i.i.d. setups, we follow the strategy in [19] for $N = 20$ clients and assign each client a unique local data with only five out of ten labels. For client-edge-cloud framework, we set four edge servers and each edge server manages five clients. All 20 clients are selected to generate the GM in following experiments.

Furthermore, we use two different data set split settings to validate the performance of the above algorithms. In Setting 1, we used 200, 200, and 100 training samples and 800, 800, and 400 test samples in each class for MNIST, FMNIST, and CIFAR-10 data sets, respectively. In Setting 2, 900, 900, and
TABLE II
GM PERFORMANCE OF DIFFERENT ALGORITHMS ON MNIST, FMNIST, AND CIFAR-10. WE MAINTAIN $|D| = 20$, $T = 800$, $K = 5$, AND $\eta_1 = \eta_2 = 0.05$ ACROSS ALL ALGORITHMS AND FINE-TUNE OTHER HYPERPARAMETERS. BEST RESULTS ARE BOLDED.

| Dataset | Method | Setting 1 (%) | Setting 2 (%) |
|---------|--------|---------------|---------------|
| MNIST   | FedAvg [3] | 91.63 ± 0.12  | 94.31 ± 0.08  |
|         | HierFAGV [24] | 93.22 ± 0.03  | 95.05 ± 0.07  |
|         | Fedprox [18] | 93.43 ± 0.15  | 90.04 ± 0.08  |
|         | pFedMe [19] | 89.34 ± 0.04  | 93.04 ± 0.14  |
|         | sFedHP (Ours) | 94.93 ± 0.03  | 97.66 ± 0.06  |
| FMNIST  | FedAvg [3] | 79.11 ± 0.09  | 84.48 ± 0.06  |
|         | HierFAGV [24] | 83.55 ± 0.15  | 85.48 ± 0.08  |
|         | Fedprox [18] | 78.67 ± 0.04  | 84.33 ± 0.10  |
|         | pFedMe [19] | 80.57 ± 0.07  | 85.14 ± 0.11  |
|         | sFedHP (Ours) | 85.64 ± 0.04  | 89.29 ± 0.05  |
| CIFAR-10| FedAvg [3] | 42.36 ± 0.19  | 81.74 ± 0.17  |
|         | HierFAGV [24] | 52.76 ± 0.21  | 87.28 ± 0.13  |
|         | Fedprox [18] | 39.10 ± 0.23  | 72.98 ± 0.35  |
|         | pFedMe [19] | 58.66 ± 0.15  | 90.31 ± 0.22  |
|         | sFedHP (Ours) | 78.44 ± 0.08  | 94.09 ± 0.10  |

Table II shows the GM performance of each algorithm. We maintain $|D| = 20$, $T = 800$, $K = 5$, and $\eta_1 = \eta_2 = 0.05$ across all algorithms and fine-tune other fundamental hyperparameters. We found that hierarchical FL methods, such as sFedHP and HierFAGV, significantly outperformed other FL methods, including FedAvg, FedProx, and pFedMe, in the device-cloud structure on non-i.i.d. data. The proposed algorithm, sFedHP, outperforms the comparative algorithms concerning the GM in all settings by more than 1% (Setting 1 on MNIST), 2% (Setting 2 on MNIST), 2% (Setting 1 on FMNIST), 3% (Setting 2 on FMNIST), 19% (Setting 1 on CIFAR-10), and 3% (Setting 2 on CIFAR-10).

We present more detailed results for fine-tuning hyperparameters, including methods without a GM, in Tables III–VI.
model that does not interact with the GM in HeurFedAMP. For HeurFedAMP, we tune \( \alpha \in [0.1, 0.5] \), where \( \alpha \) represents the proportion of the client model that does not interact with the GM. For Fedprox, we tune \( \lambda \in \{0.001, 0.01, 0.1, 1\} \) following the setting in [18]. For pFedMe and \( s\text{FedHP} \), we use the same setting as in [19] with \( \lambda \in \{5, 15, 20, 25, 30\} \). For pFedGP, we use the same setting as in [22] for other basic hyperparameters. We find that \( s\text{FedHP} \) outperforms all compared algorithms in the GM and achieves state-of-the-art results in the PM.

### B. More Results

We compare the GM between \( s\text{FedHP} \), pFedMe [19] and HierF AVG [24] on \( \mu \)-strongly convex and nonconvex situations. Similarly with [19], in each round of local training, the client uses \( K = 5 \) gradient-based iterations to obtain an approximated optimal local model, i.e., solve (8) in \( s\text{FedHP} \). An \( \ell_2 \)-regularized multinomial logistic regression (MLR) model is used for \( \mu \)-strong convex situation, while a DNN with two hidden layers of size \([500, 200]\) is used for nonconvex situation. In our experiments, \((3/4)\) data sets are for training and the others are for testing. All experiments were conducted on an NVIDIA Quadro RTX 6000 environment, and the code based on PyTorch is available online.

We use both sparsity and nonsparsity settings for \( s\text{FedHP} \). In the sparsity setting, we set \( \gamma_1 = \gamma_2 = 0.001 \) at the beginning, while set \( \gamma_1 \) and \( \gamma_2 \) to a tiny number, respectively, when the client model sparsity is lower than 0.2 and after training 100 global rounds. Since according to our theoretical results, a tiny \( \gamma_1, \gamma_2 \) can reduce the convergence cost. In nonsparsity setting, we set \( \gamma_1 = \gamma_2 = 0 \).

Fig. 2 shows the performance of \( s\text{FedHP} \), HierFAVG and pFedMe in \( \mu \)-strongly convex situation. Since the MLR model is already lightweight enough, we set \( \gamma_1 = \gamma_2 = 0.001 \) at the beginning, while set \( \gamma_1 \) and \( \gamma_2 \) to a tiny number, respectively, when the client model sparsity is lower than 0.2 and after training 100 global rounds. Since according to our theoretical results, a tiny \( \gamma_1, \gamma_2 \) can reduce the convergence cost. In nonsparsity setting, we set \( \gamma_1 = \gamma_2 = 0 \).

We use both sparsity and nonsparsity settings for \( s\text{FedHP} \). In the sparsity setting, we set \( \gamma_1 = \gamma_2 = 0.001 \) at the beginning, while set \( \gamma_1 \) and \( \gamma_2 \) to a tiny number, respectively, when the client model sparsity is lower than 0.2 and after training 100 global rounds. Since according to our theoretical results, a tiny \( \gamma_1, \gamma_2 \) can reduce the convergence cost. In nonsparsity setting, we set \( \gamma_1 = \gamma_2 = 0 \).

Fig. 3 shows the performance of \( s\text{FedHP} \), HierFAVG and pFedMe in nonconvex situation. We test sparsity and nonsparsity settings for \( s\text{FedHP} \) in accuracy, model sparsity and the accumulative communication. We set 20% sparsity, the
proportion of nonzero parameters, as a lower bound to preserve performance. According to Fig. 3, the proposed sFedHP can reduce the communication cost while achieve good test accuracy simultaneously. Specifically, sFedHP achieves similar performance in a GM with pFedMe while reducing communication costs by 80% and obtains higher performance when using a nonsparsity setting.

Remark 1: The accumulated communication cost in Fig. 3 is calculated based on one client in the client-cloud FL or one edge server in the client-edge-cloud FL since the client and the edge server are nearby between which low-cost communication is possible. Consider the DNN model with 79,510 parameters, whose nonzero parameters are quantized in 64 bits while zero parameters are quantized in 1 bit. 79,510 bits location parameters are needed to mark the positions of zero parameters in sFedHP. This section aims to demonstrate the excellent sparsity of sFedHP qualitatively. We focused on analyzing the ability of sFedHP to process non-i.i.d. data under different settings. More rigorous communication cost analysis needs to cooperate with the encoding and decoding process.

In summary, sFedHP performs better than pFedMe and HierFAVG in test accuracy, convergence rate, and communication cost in μ-strongly convex and nonconvex situations. Furthermore, the hierarchical personalization scheme is more suitable for solving statistical diversity problems.

C. Effect of Hyperparameters

We empirically study the effect of different hyperparameters in sFedHP.

Effects of λ: According to Fig. 4, properly increasing λ can effectively improve the test accuracy and convergence rate for sFedHP. And we find that an oversize λ1 and λ2 may cause gradient explosion.

Effects of γ: Fig. 5 shows the relationship of γ with the model sparsity and the convergence speed, we can see that increasing γ will reduce the communication speed and global accuracy.

VI. Conclusion

In this article, we propose sFedHP as a sparse hierarchical personalized FL algorithm that can greatly remiss the statistical diversity issue to improve the FL performance and reduce the communication cost in FL. Our approach uses an approximated ℓ1-norm and the hierarchical proximal mapping to generate the loss function. The hierarchical proximal mapping enables the personalized edge-model, and client-model optimization can be decomposed from the GM learning, which allows sFedHP parallelly optimizes the personalized edge-model and client-model by solving a tri-level problem. Theoretical results present that the sparse constraint in sFedHP only reduces the convergence speed to a small extent. Experimental results further demonstrate that sFedHP outperforms the client-edge-cloud HierFAVG and many other personalized FL methods based on local customization under different settings.

APPENDIX

Proof of the Results

Proof of Lemma 1: First, we have

\[
E \left[ \|\mathbf{w}^{t+1} - \mathbf{w}^* \|^2 \right] = E \left[ \|\mathbf{w}^t - \bar{\mathbf{g}}^t - \mathbf{w}^* \|^2 \right] = E \left[ \|\mathbf{w}^t - \mathbf{w}^* \|^2 \right] - 2\bar{\eta}E \left[ \langle \mathbf{g}^* , \mathbf{w}^t - \mathbf{w}^* \rangle \right] + \bar{\eta}^2 E \left[ \|\mathbf{g}^* \|^2 \right]
\] (14)

and the second term of (14) is as follows:

\[
- E \left[ \langle \mathbf{g}^* , \mathbf{w}^t - \mathbf{w}^* \rangle \right] = - \frac{1}{NR} \sum_{i \in R} \sum_{l \in P} ((g^*_{i,l} - \nabla F_i (\mathbf{w}^t)). (\mathbf{w}^t - \mathbf{w}^*))
\]

\[
= \frac{1}{2NR} \sum_{i \in R} \sum_{l \in P} \left\| g^*_{i,l} - \nabla F_i (\mathbf{w}^t) \right\|^2_{\mu_F} + \mu_F \left\| \mathbf{w} - \mathbf{w}^* \right\|^2_2
\] (15)
where (a) follows by the Peter Paul inequality and the Jensen’s inequality and (b) is due to Theorem 1.

From [19, eqs. (18) and (19)], we have

\[ \left\| g'_t \right\|^2 \leq \frac{3}{NR} \sum_{i,r} \left\| g'_t \right\|_2^2 \]

\[ + 3 \left\{ \frac{1}{S} \sum_{i \in S^t} \left\| F_i(w') - F_i(w) \right\|^2 \right. \]

\[ + 6L_F (F(w') - F(w)) \]  

(16)

Taking expectation of the second term of (16) w.r.t. edge server sampling, we have

\[ \mathbb{E}_{S_t} \left[ \frac{1}{S} \sum_{i \in S^t} \left\| F_i(w') - F_i(w) \right\|^2 \right] \]

\[ \leq \frac{1}{S} \sum_{i,j} \mathbb{E}_{S_t} \left( \sum_{i \in S^t} \left\| F_i(w') - F_i(w) \right\|^2 \right) \]

\[ \leq \frac{1}{S} \sum_{i,j} \mathbb{E}_{S_t} \left( \left\| F_i(w') - F_i(w) \right\|^2 \right) \]

(17)

where (a) is due to Jensen’s inequality; \( \mathbb{1}_X \) in (b) is an indicator function of an event \( X \); and \( \mathbb{E}_{S_t} [\mathbb{1}_{i \in S^t}] = (S/N) \).

By substituting (15), (16), and (17) into (14), we finish the proof of Lemma 1.

Proof of Lemma 2: By noting that the last term of (8) only w.r.t. \( y_i^{r,j} \) we have

\[ \tilde{\theta}_{i,j} (y_i^{r,j}) = \arg \min_{\theta_{i,j} \in \mathbb{R}^d} \tilde{L}_{i,j}(\theta_{i,j}) \]

\[ + \frac{\lambda_1}{2} \left\| \theta_{i,j} - y_i^{r,j} \right\|^2_2 + \gamma_1 \phi_p (\theta_{i,j}) \]

\[ \hat{\theta}_{i,j} (y_i^{r,j}) = \arg \min_{\theta_{i,j} \in \mathbb{R}^d} L_{i,j}(\theta_{i,j}) \]

\[ + \frac{\lambda_1}{2} \left\| \theta_{i,j} - y_i^{r,j} \right\|^2_2 + \gamma_1 \phi_p (\theta_{i,j}) \]

(18)

Denote \( h_{i,j}(\theta_{i,j}; y_i^{r,j}) = \ell_{i,j}(\theta_{i,j}) + \gamma_1 \phi_p (\theta_{i,j}) + (\lambda_1/2) \| \theta_{i,j} - y_i^{r,j} \|^2_2 \). Then, \( h_{i,j}(\theta_{i,j}; y_i^{r,j}) \) is \((\lambda_1 + \mu)-\)strongly convex when A.1(a) holds and \((\lambda_1 - L - (\gamma_1 \rho))\)-strongly convex when A.1(b) and \( \lambda_1 > L + (\gamma_1 \rho) \) holds. Follow the proof in [19], we obtain the result in Lemma 2.

Proof of Lemma 3:

\[ \mathbb{E} \left[ \left\| g'_t \right\|- \mathbb{E} \left[ \left\| g'_t \right\| \right] \right]^2 \]

\[ \leq 2 \mathbb{E} \left[ \left\| g'_t \right\| - \mathbb{E} \left[ \left\| g'_t \right\| \right] \right]^2 \]

(19)

where (a) follows by Jensen’s inequality; (b) is due to Theorem 1 and the first-order conditions \( \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \lambda_2 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) = 0 \) from (3) with \( \lambda = (\lambda_1 \lambda_2 / \lambda_1 + \lambda_2) \). Moreover, the last term is bounded by

\[ \mathbb{E} \left[ \left\| w_t^{r,j} - w_t^{r,j} \right\| \right] \]

\[ \leq \frac{8 \tilde{\eta}}{\beta L_F} \left[ 3 \mathbb{E} \left[ \left\| F_i(w') - F_i(w) \right\| \right] + \frac{2 \lambda_2 \beta^2}{R} \right] \]

where (a) follows by our proof of [38, Lemma 2]. Thus, we finish the proof of Lemma 3.

Proof of Lemma 4: We first prove case (a), we have

\[ \frac{1}{Nj} \sum_{i=1,j=1}^{Nj} \left\| F_i(w) - F_i(w) \right\|^2 \]

\[ \leq \frac{1}{Nj} \sum_{i=1,j=1}^{Nj} \left\| F_i(w) - F_i(w) \right\|^2 \]

\[ \leq \frac{2}{Nj} \sum_{i=1,j=1}^{Nj} \left\| F_i(w) - F_i(w) \right\|^2 \]

\[ \leq 4L_F (F(w) - F(w)) + \frac{2}{Nj} \sum_{i=1,j=1}^{Nj} \left\| F_i(w) - F_i(w) \right\|^2 \]

(20)

where (a) follows by \( \mathbb{E}[\| x - \mathbb{E}[x] \|^2] = \mathbb{E}[\| x - \mathbb{E}[x] \|^2] - \mathbb{E}[\| x \|^2] \); (b) follows by Jensen’s inequality; (c) follows by Theorem 1, thus \( F_i \) is \( \mu_F \)-strong convex and \( L_F \)-smooth with \( \mathbb{E} (F(w) - F(w)) = 0 \).

We next prove case (b). For the minimization problem in (3), we have the first-order conditions \( \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \lambda_2 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) = 0 \) and \( \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \gamma_1 \phi_p (\tilde{\theta}_{i,j}) = 0 \), then we have

\[ \nabla F_{i,j}(w) = \lambda_2 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \gamma_2 \phi_p (w) \]

\[ = \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \gamma_2 \phi_p (w) \]

\[ = \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \gamma_2 \phi_p (w) \]

\[ = \lambda_1 (\hat{\theta}_{i,j} - \tilde{\theta}_{i,j}) + \gamma_2 \phi_p (w) \]

where \( L_{i,j} = \ell_{i,j}(\cdot) + \gamma_1 \phi_p (\cdot) \).
Hence, we have
\[
\left\| \nabla F_{i,j}(\omega) - \nabla F(\omega) \right\|^2_2 \\
\leq \frac{2}{N^2} \sum_{i,j=1}^{N_J} \left( \left\| \nabla L_{i,j}(\hat{\theta}_{i,j}) - \frac{1}{N_J} \sum_{p,q=1}^{N_J} \nabla L_{p,q}(\hat{\theta}_{i,j}) \right\|^2_2 + \frac{1}{N_J^2} \sum_{i,j=1}^{N_J} \left( \left\| \nabla L_{p,q}(\hat{\theta}_{i,j}) - \nabla L_{p,q}(\hat{\theta}_{p,q}) \right\|^2_2 \right) \right)
\]

where (a) follows by (19) and Jensen’s inequality. Taking the average over the number of clients, we have
\[
\frac{1}{N_J} \sum_{i,j=1}^{N_J} \left\| \nabla F_{i,j}(\omega) - \nabla F(\omega) \right\|^2_2 \\
\leq 2\sigma^2 + \frac{2}{N^2} \sum_{i,j=1}^{N_J} \sum_{p,q=1}^{N_J} \left( \left\| \nabla L_{i,j}(\hat{\theta}_{i,j}) - \frac{1}{N_J} \sum_{p,q=1}^{N_J} \nabla L_{p,q}(\hat{\theta}_{i,j}) \right\|^2_2 + \frac{1}{N_J^2} \sum_{i,j=1}^{N_J} \left( \left\| \nabla L_{p,q}(\hat{\theta}_{i,j}) - \nabla L_{p,q}(\hat{\theta}_{p,q}) \right\|^2_2 \right) \right)
\]

(20)

where (a) follows by Assumption 3 and Jensen’s inequality; (b) is due to the \((L + (\gamma_1/\rho))-\)smoothness of \(L_{i,j}(\cdot)\); and the last term
\[
\sum_{i,j=1}^{N_J} \sum_{p,q=1}^{N_J} \left\| \hat{\theta}_{i,j} - \hat{\theta}_{p,q} \right\|^2_2 \\
\leq 4 \sum_{i,j=1}^{N_J} \sum_{p,q=1}^{N_J} \left( \left\| \hat{\theta}_{i,j} - \hat{y}_{i,j} - \frac{\gamma_2}{\lambda_1} \nabla \varphi_\rho(\omega) \right\|^2_2 - \left\| \hat{y}_{i,j} - \omega - \frac{\gamma_2}{\lambda_2} \nabla \varphi_\rho(\omega) \right\|^2_2 + \left\| \hat{y}_{i,j} - \omega - \frac{\gamma_2}{\lambda_2} \nabla \varphi_\rho(\omega) \right\|^2_2 \right)
\]

where (a) is due to Jensen’s inequality; (b) follows by (19) and rearranging the terms. Substituting (21) back to (20) we have
\[
\frac{1}{N_J} \sum_{i,j=1}^{N_J} \left\| \nabla F_{i,j}(\omega) - \nabla F(\omega) \right\|^2_2 \\
\leq 4 \sum_{i,j=1}^{N_J} \sum_{p,q=1}^{N_J} \left( \left\| \hat{\theta}_{i,j} - \hat{y}_{i,j} - \frac{\gamma_2}{\lambda_1} \nabla \varphi_\rho(\omega) \right\|^2_2 - \left\| \hat{y}_{i,j} - \omega - \frac{\gamma_2}{\lambda_2} \nabla \varphi_\rho(\omega) \right\|^2_2 + \left\| \hat{y}_{i,j} - \omega - \frac{\gamma_2}{\lambda_2} \nabla \varphi_\rho(\omega) \right\|^2_2 \right)
\]

(21)

where (a) follows by Jensen’s inequality and (19); and (b) is due to Theorem 1. In addition, the second term of (23) is bounded by
\[
E \left[ \left\| \nabla \varphi_\rho(\omega) \right\|^2_2 \right] = E \sum_i \left( \frac{\sinh(|w^T_n|/\rho)}{\cosh(|w^T_n|/\rho)} \right)^2 \leq d^2_n
\]

(24)

where \((\sinh(|w^T_n|/\rho)/\cosh(|w^T_n|/\rho)) \in (-1, 1), d^2_n \) is the \(n\)th element of \(w^T\) and \(d^2_n\) denotes the number of nonzero elements in \(w\). Due to the \(\mu_F\)-strong convex of \(F\), we have
\[
E \left[ \left\| w^T - w^* \right\|^2_2 \right] \leq \frac{2(E[F(w^T) - F(w^*)])}{\mu_F}
\]

(25)

By substituting (22), (24), and (25) into (23) we complete the proof. □
Proof of Theorem 3: We first prove part (a). Similar with the proof in [19], we rewrite the recursive formula due to the $L$-smooth of $F(\cdot)$

$$E\left[F(w^{t+1}) - F(w^t)\right]$$

\[
\leq E\left[\nabla F(w^t), w^{t+1} - w^t\right] + \frac{L_F}{2} E\left[\|w^{t+1} - w^t\|^2\right]
\]

\[
= -\eta E[\nabla F(w^t), g^t] + \frac{\eta^2 L_F}{2} E\left[\|g^t\|^2\right]
\]

\[
= -\eta E\left[\nabla F(w^t)\|^2\right] - \eta E[\nabla F(w^t), g^t - \nabla F(w^t)]
\]

\[
+ \frac{\eta^2 L_F}{2} E\left[\|g^t\|^2\right]
\]

\[
\leq -\eta E\left[\nabla F(w^t)\|^2\right] + \frac{\eta^2 L_F}{2} E\left[\|g^t\|^2\right]
\]

\[
= \frac{\eta}{2} \frac{\sum_{i=1}^{N_J} \sum_{l,r} g^t_{i,l,r} - \nabla F_i(w^t)}{2^2} E\left[\|g^t\|^2\right]
\]

\[
+ \frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2}
\]

\[
\leq -\eta E[\nabla F(w^t)] + \eta G_1 + \eta G_2
\]

where (b) is due to the Cauchy-Swarz inequalities; (c) is due to Jensen’s inequality; (d) follows by $1 + 3L_F \eta \leq 3\beta$ since $\eta \leq (\beta/2L_F)$, and $G_1 = 3\beta^2 \lambda^2$, $G_2 = (3L_F^2(49\beta^2 + 16\lambda^2)/\rho)$.

Next, let $\lambda^2 \leq 16(L + (\gamma_1/\rho))^2 \geq 1$ and $\hat{\eta} \leq (1/588L_F^2)$

$$G = \frac{1}{2} - \frac{\sum_{j=1}^{N_J} \sum_{i,l,r} g^t_{i,l,r} - \nabla F_i(w^t)}{2^2}$$

we hence have

$$\frac{\sum_{j=1}^{N_J} \sum_{i,l,r} g^t_{i,l,r} - \nabla F_i(w^t)}{2^2} \leq \frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2}
\]

we therefore have

$$\frac{1}{T} \sum_{t=0}^{T-1} E\left[\nabla F(w^t)\|^2\right] \leq \frac{\left(\frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2} + \frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2}
\]

\[
\leq 4 \left(\frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2} + \frac{\lambda^2}{2} \frac{\sum_{i=1}^{N_J} \sum_{j} \left(E_{\gamma_{i,j}}[\|\nabla F_i(w^t) - \nabla F(w^t)\|^2]\right)}{2^2}
\]

where (a) is due to Jensen’s inequality, (b) is due to (19). And the last term is bounded by

$$\frac{1}{NJ} \sum_{i=1}^{N_J} \left(\|F_i(w^t) - F(w^t)\|^2\right)$$

where (a) is due to $E[\|x - E[x]\|^2] = E[\|x - E[x]\|^2] + E[\|E[x] - E[x]\|^2]$, (b) follows by Lemma 4, (c) is by setting $\lambda^2 \leq 16(L + (\gamma_1/\rho))^2 \leq 1$. By substituting (27) into (26), we finished the proof.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[2] J. Chen and X. Ran, “Deep learning with edge computing: A review,” Proc. IEEE, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” Proc. Artif. Intell. Stat., vol. 2017, pp. 1273–1282.
[4] W. Zhang et al., “Blockchain-based federated learning for device failure detection in industrial IoT,” IEEE Internet Things J., vol. 8, no. 7, pp. 5926–5937, Apr. 2021.
[5] B. Ghimire and D. B. Rawat, “Recent advances on federated learning for cybersecurity and cybersecure for federated learning for Internet of Things,” IEEE Internet Things J., vol. 9, no. 11, pp. 8229–8249, Jun. 2022.
[6] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, “Communication-efficient edge AI: Algorithms and systems,” IEEE Commun. Surveys Tuts., vol. 22, no. 4, pp. 2167–2191, 4th Quart., 2020.
[7] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” IEEE Trans. Wireless Commun., vol. 20, no. 1, pp. 269–283, Jan. 2021.
[8] J. Le, X. Lei, N. Mu, H. Zhang, K. Zeng, and X. Liao, “Federated continuous learning with broad network architecture,” IEEE Trans. Cybern., vol. 51, no. 8, pp. 3874–3888, Aug. 2021.
[9] S. R. Pokhrel and J. Choi, “Federated learning with blockchain for autonomous vehicles: Analysis and design challenges,” IEEE Trans. Commun., vol. 68, no. 8, pp. 4734–4746, Aug. 2020.
[10] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, “Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis,” IEEE Access, vol. 9, pp. 138509–138542, 2021.
[11] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges,” IEEE Commun. Surveys Tuts., vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.
[12] Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, “Federated learning for automatic modulation classification under class imbalance and varying noise condition,” IEEE Trans. Cogn. Commun. Netw., vol. 8, no. 1, pp. 86–96, Mar. 2022.
L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical personalized federated learning," 2020, arXiv:1912.00818.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, arXiv:1812.06127.

C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 21394–21405.

Y. Huang et al., "Personalized cross-silo federated learning on non-iid data," in Proc. Assoc. Adv. Artif. Intell., vol. 35, 2021, pp. 7865–7873.

A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, arXiv:2002.07948.

I. Achituve, A. Shamsian, A. Navon, G. Chechik, and E. Fetaya, "Personalized federated learning with Gaussian processes," in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 8392–8406.

S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in Proc. Int. Conf. Mach. Learn., 2020, pp. 5132–5143.

L. Liu, J. Zhang, S. Song, and K. B. Letlafet, "Client-edge-cloud hierarchical federated learning," in Proc. IEEE Int. Conf. Commun., 2020, pp. 1–6.

K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, arXiv:1902.01046.

K. Hsieh, A. Phanshaye, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in Proc. Int. Conf. Mach. Learn., 2020, pp. 4387–4398.

P. Kairouz et al., "Advances and open problems in federated learning," Found. Trends® Mach. Learn., vol. 14, nos. 1–2, pp. 1–210, 2021.

Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," 2020, arXiv:2002.10619.

G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, arXiv:1503.02531.

T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 2351–2363.

Y. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 4424–4434.

N. Shoham et al., "Overcoming forgetting in federated learning on non-iid data," 2019, arXiv:1910.07796.

J. Sun, Q. Qu, and J. Wright, "Complete dictionary recovery over the sphere I: Overview and the geometric picture," IEEE Trans. Inf. Theory, vol. 63, no. 2, pp. 853–884, Feb. 2017.

A. Beck, First-Order Methods in Optimization, Philadelphia, PA, USA: SIAM, 2017.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, arXiv:1708.07747.

A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

X. Liu, Y. Li, Q. Wang, X. Zhang, Y. Shao, and Y. Geng, "Sparse personalized federated learning," IEEE Trans. Neural Netw. Learn. Syst., 2023, early access, Mar. 8, 2023, doi: 10.1109/TNNLS.2023.3250658.

Yunfeng Shao (Member, IEEE) received the B.S. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009, and the Ph.D. degree in electronic engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2014. He is currently an Expert with Huawei Noah’s Ark Lab, Beijing. His research interests include machine learning with privacy protection, federated learning, transfer learning, and their applications in telecommunication network.

Yinchuan Li (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in electronic engineering from Tianjin University, Tianjin, China, in 2004, 2007, and 2010, respectively. From November 2017 to November 2019, he was a visiting scholar with the Department of Electrical Engineering, Columbia University, New York, NY, USA. From December 2019 to August 2020, he was a Senior Technical Consultant with Santé Ventures, Austin, TX, USA. He currently works with Noah’s Ark Lab, Huawei Technologies, Beijing, as an AI Researcher. His current research interests include machine learning, deep learning, reinforcement learning, and sparse signal processing.

Xiaofeng Liu received the B.Eng. degree from Tianjin University, Tianjin, China, in 2018, where he is currently pursuing the Ph.D. degree in information and communication engineering with the School of Electrical and Information Engineering. His research interests include federated learning, model compression, and intelligent signal processing.

Qing Wang (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in electronic engineering from Tianjin University, Tianjin, China, in 2004, 2007, and 2010, respectively. She is currently a Full Professor with the School of Electrical and Information Engineering, Tianjin University. From 2007 to 2009, she was a visiting student with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. From 2018 to 2019, she was a Visiting Scholar with Columbia University, New York, NY, USA. Her current research interests include intelligent signal processing, array signal processing, and integrated sensing and communication systems.

Prof. Wang served as the Vice Dean of the School of Marine Science and Technology and an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS.