Depersonalized Federated Learning: Tackling Statistical Heterogeneity by Alternating Stochastic Gradient Descent

Yujie Zhou\textsuperscript{1,2,3}, Zhidu Li\textsuperscript{1,2,3}, Tong Tang\textsuperscript{1,2,3}, Ruyan Wang\textsuperscript{1,2,3}

\textsuperscript{1} Chongqing University of Posts and Telecommunications, School of Communication and Information Engineering, China
\textsuperscript{2} Advanced Network and Intelligent Interconnection Technology Key Laboratory of Chongqing Education Commission of China
\textsuperscript{3} Key Laboratory of Ubiquitous Sensing and Networking in Chongqing, China
Email: lizd@cqupt.edu.cn

Abstract—Federated learning (FL) enables distributed clients to cooperatively train a common machine learning (ML) model for intelligent inference without raw data sharing. However, problems in practical networks, such as non-independent-and-identically-distributed (non-iid) raw data and limited network resources, lead to slow and unstable convergence of the FL training process. To address these issues, this paper proposes a new FL method that can mitigate statistical heterogeneity through the depersonalization mechanism. Specifically, we decouple the global and local optimization objectives by alternating stochastic gradient descent, thus reducing the accumulated variance in local update phases to accelerate the FL convergence. Furthermore, the proposed FL method is analyzed in detail and proved to converge at a sublinear speed under the general non-convex setting. Finally, extensive experiments are conducted on public datasets to verify the effectiveness of the proposed method with comparisons of other representative baseline methods.

Index Terms—Federated learning, depersonalization mechanism, statistical heterogeneity, convergence analysis

I. INTRODUCTION

The ever-increasing evolution of communication and computation technologies is now introducing us into a new era of the Internet of Intelligent, where a tremendous amount of data is generated in the edge networks and requires real-time processing via machine learning (ML) [1, 2]. Due to privacy concerns, it is required to train ML models across multiple clients without sharing raw data directly. In this regard, federated learning (FL) is proposed as a distributed cooperative learning paradigm and has attracted wide attention from academia and industry.

In a typical FL, e.g., FedAvg [3], individual models are trained via several local stochastic gradient descent (SGD) steps and then uploaded to the central server for aggregation. Despite eschewing raw data sharing, traditional FL usually suffers the following challenges in practical deployment. As clients may come from various regions, the data on each client is usually non-independent-and-identically-distributed (non-iid, a.k.a. statistical heterogeneity), which has been proved that degrades the FL performance [4–6]. Additionally, as massive clients may join in an FL training process, it is impractical for the network to support all participators to upload data simultaneously, which further aggravates the statistical heterogeneity.

Recently, some efforts have been devoted to analyzing and improving FL performance on non-iid data with partial communication (i.e., client sampling). One representative research route is to reveal the relationship between the FL performance and optimization parameters through convergence analysis. For example, work [6] provided detailed convergence analyses of FedAvg with client sampling in general convex cases, and work [7] analyzed the FL performance under a dynamic network topology in both convex and non-convex cases. Another research route is to design a new update-rule to alleviate the statistical heterogeneity, then enhance the FL performance based on FedAvg [8, 9]. For instance, works [8, 9] proposed the FedProx and the SCAFFOLD methods by introducing a proximal operator or variance controller in the local update phase, respectively. However, the deviation between the global and the individual local performance caused by the data heterogeneity was not considered in the existing works [6–9], which may slow down the convergence rate of the FL training process. Additionally, a common global solution may not always work well on each client.

To better tackle statistical heterogeneity to simultaneously enhance global and local performance, we devise a new method that modifies local update-rule via the depersonalization mechanism inspired by reversely using the model-customization techniques [10–13]. Intuitively, these works extracted a group of personalized models from a trained global one to ease the negative impact of non-iid data on individual inference, where such models contain rich local information unexploited in the prepositive global training process [14]. Note that utilizing this additional information may be beneficial to reduce the negative impact of statistical
heterogeneity globally. Hence, we perform the personalization process in advance to capture extra local information from the generated temporary personalized models and then leverage the information reversely to correct the global update direction (i.e., early personalizing for global optimizing, as shown in Fig. 1). In this regard, we design an alternating SGD rule in the local update stage to let each client produce two decoupled local models (rather than an original one) for separating the global update direction from the local one. Specifically, the personalized local model is obtained by directly performing SGD on the surrogate local objective to exclude partial information contained in the personalized local model from the original one. Therefore, each sampled client can upload its globalized model in place of the original local one to reduce the accumulated local deviations for convergence acceleration and stabilization. The key contributions of this paper are summarized as follows:

- A novel method called FedDeper is proposed to improve the FL performance on non-iid data by the depersonalization update mechanism, which can be widely adapted to a variety of scenarios.
- The convergence performance of the FedDeper is proved and analyzed theoretically for both personalized local models and the aggregated global model in the general non-convex setting.
- Experimental results are provided to show the advantages of our proposed algorithm versus representative baselines and further illustrate the effectiveness of the proposed update mechanism in FL convergence acceleration.

The remainder of this paper is organized as follows. The FL backgrounds are discussed in Section II. A new FedDeper method is proposed in Section III, and its convergence is analyzed in Section IV. Experimental results are presented in Section V. The paper is concluded in Section VI.

II. PRELIMINARIES AND BACKGROUNDS

For an FL process with participating clients \( \mathcal{N} \), we have the following optimization objective [7–9, 15]

\[
\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i \in \mathcal{N}} f_i(x)
\]

(1)

where \( n := |\mathcal{N}| \) denotes the cardinal number of \( \mathcal{N} \), \( d \) denotes the dimension of the vector \( x \), and \( f_i(x) := \mathbb{E}_{\theta_i \sim D_i}[f(x; \theta_i)] \) represents the local objective function on each client \( i \). Besides, \( f_i \) is generally the loss function defined by the local ML model, and \( \theta_i \) denotes a data sample belonging to the local dataset \( D_i \). In this paper, we mainly deal with Problem (1), and all the results can be extended to the weighted version according to the techniques proposed in [6, 16].

To solve (1), we depict a round of the classical algorithm FedAvg as three phases: In the \( k \)-th round, (i) Broadcasting: The server uniformly samples a subset of \( m \) clients (i.e., \( \mathcal{U}^k \subseteq \mathcal{N} \) with \( m := |\mathcal{U}^k| \leq n \), \( \forall k \in \{0,1,...,K-1\} \) for any integer \( K \geq 1 \)) and broadcasts the aggregated global model \( x^k \) to client \( i \in \mathcal{U}^k \). (ii) Local Update: Each selected client \( i \) initializes the local model \( v_{i,0}^k \) as \( x^k \) and then trains the model by performing SGD with a step size \( \eta \) on \( f_i(\cdot) \),

\[
v_{i,j+1}^k \leftarrow v_{i,j}^k - \eta g_i(v_{i,j}^k), \ \forall j \in \{0,1,...,\tau - 1\}, (2)
\]

where \( v_{i,j}^k \) denotes the updated local model in the \( j \)-th step SGD and \( g_i(\cdot) \) represents the stochastic gradient of \( f_i(\cdot) \). During the number of local steps reaches a certain threshold \( \tau \), client \( i \) will upload its local model to the server. (iii) Global Aggregation: The server aggregates all received local models to derive a new global one for the next phase,

\[
x^{k+1} \leftarrow \frac{1}{m} \sum_{i \in \mathcal{U}^k} v_{i,\tau}^k.
\]

(3)

The whole FL process is completed when the number of communication rounds reaches the upper limit \( K \) and the final global model is obtained.

Note that the stochastic gradient \( g_i(\cdot) \) in (2) can be more precisely rewritten as \( g_i(\cdot) = \nabla f_i(\cdot; \theta_i) \) with \( \theta_i \sim D_i \). Due to the high statistical heterogeneity, local datasets \( \{D_i\}_{i \in \mathcal{N}} \) obey unbalanced data distributions, and the corresponding generated gradients are consequently different in expectation:

\[
\mathbb{E}_{\theta_i \sim D_i} [f_i(\cdot; \theta_i)] \neq \mathbb{E}_{\theta_j \sim D_j} [f_j(\cdot; \theta_j)], \ \forall i,j \in \mathcal{N}, i \neq j. (4)
\]

That means performing SGD process according to (2) leads to the deviation between the local solution \( v_i^* \) for \( \nabla f_i(v_i^*) = 0 \) and the global solution \( x^* \) for \( \nabla f(x^*) = 0 \), which results in slow FL convergence [4–6]. Moreover, for an FL network with massive participators and limited communication resources, only a small fraction of clients can be selected to join a training round, i.e., \( m \ll n \). Such client sampling (or called partial communication) aggravates the inconsistency of local models, thus further leading to unstable training and poor performance.

III. FEDERATED LEARNING WITH DEPERSONALIZATION

To alleviate the negative impact of non-iid data and partial communication on FL, we propose a new Depersonalized FL (FedDeper) algorithm. As depicted in Fig. 1, we aim
to ascertain local approximations of the global model on each client with personalized models and then upload and aggregate them in place of the original local models for training stabilization and acceleration. In the subsequent, the FedDeper is introduced in detail.

Decoupling Global and Local Updating: Recall that performing (2) aims to minimize the local objective $f_i(\cdot)$ that usually disagrees with the global objective (1), which results in slow FL convergence. To deal with this issue, we propose a new depersonalization mechanism to decouple the two objectives. In specific, to optimize the objective $f(\cdot)$, we induce a more globalized local model in place of the original local model to mitigate the local variance accumulation in aggregation rounds. Different from (2), the SGD is carried out on the following surrogate loss in each selected client $i$,

$$ f^i_i(y_i) := f_i(y_i) + \frac{\rho}{2\eta} \| v_i + y_i - 2x \|^2, \quad (5) $$

where $\frac{\rho}{2\eta}$ is a constant for balancing the two terms, and $v_i$ denotes the personalized local model (the analogue of the original local model). We aim to obtain two models in this phase. The personalized model $v_i$ used to search the local solution $v_i^*$ remains locally while the globalized model $y_i$ used to estimate the global model locally (i.e., $y_i^* \approx x^*$) is uploaded to the aggregator to accelerate FL convergence.

Using Local Information Reversely with Regularizer: As shown in Fig. 2, the globalized model $y_i$ is updated by using the personalized one $v_i$ reversely. To minimize (5), the value of $y_i$ is restricted to a place slightly away from the local optimum with the regularizer $\| v_i + y_i - 2x \|^2$. More specifically, we regard $(v_i - x)$ and $(y_i - x)$ as two directions in the update. Since $v_i$ is a personalized solution for the client, we note that $(v_i - x)$ contains abundant information about local deviations. To avoid introducing overmuch variance, we give a penalty to term $(y_i - x)$ that reflects $y_i$ to the opposite direction of $(v_i - x)$. Nevertheless, in suppressing bias with the regularizer $\| v_i + y_i - 2x \|^2$, we also eliminate the global update direction implied in $(v_i - x)$, which further interprets the necessity of carefully tuning on $\rho$ and $\eta$ (i.e., trading off variance reduction and convergence acceleration).

Retaining Historical Information for Personalized Model: In the current local update stage, model $y_i$ is initialized as the received global model $x$ while $v_i$ is initialized as the trained $y_i$ in the previous stage. And then they are updated alternately by first-order optimizers, i.e., each selected client $i$ performs SGD on $f_i(\cdot)$ according to (2) and (5) to obtain a personalized local model and an approximated globalized model, respectively. However, this initialization policy results in the new $v_i$ forgetting all accumulated local information contained in the previous $v_i$. To fully use of the historical models, we let $v_i$ partially inherit the preceding value, i.e.,

$$ v_{i,0}^{k+1} \leftarrow (1-\lambda)v_{i,\tau}^{k} + \lambda y_{i,\tau}^{k}, \quad (6) $$

where $v_{i,\tau}^{k}$ and $y_{i,\tau}^{k}$ are the trained models in the $k$-th round, $v_{i,0}^{k+1}$ is the initial model in the $(k+1)$-th round, and $\lambda$ is the mixing rate controlling the stock of local deviation information. Intuitively, $\lambda$ limits the distance between the initial $v_i$ and $y_i$ within a certain range to avoid destructively large correction caused by $\| v_i + y_i - 2x \|^2$ since the monotonically increasing difference between $v_i$ and $x$ (e.g., $\| v_i - x \|$) in updating. Additionally, if hyper-parameter $\lambda$ is set within interval $[0, 1]$, there always exists suitable $\eta$ and $\rho$ enabling the global model $x$ to converge to a stationary point (See Section IV for details).

Procedure of FedDeper: The proposed method is summarized as Algorithm 1. In Lines 7-8, the globalized local model $y_i$ and the personalized model $v_i$ are updated with alternating SGD. Line 7 shows the $j$-th step of local SGD where $\nu$ is involved in the stochastic (mini-batch) gradient $g_i^{j}$ of $f_i^{j}$. Line 8 shows a step of SGD for approaching the optimum of the local objective. In Line 10, we initialize the personalized model $v_i$ for the next round of local update according to (6). In Line 13, client $i \in \mathbb{U}^k$ skips the current round and only updates

---

**Algorithm 1 FedDeper: Depersonalized Federated Learning**

**Input:** learning rate $\eta$, penalty $\rho$, mixing rate $\lambda$, local step $\tau$, total round $K$, initialized models $x^0 = y_{0,0}^0 = v_{0,0}^0$

1: for each round $k = 0, 1, \ldots, K - 1$
2: sample clients $\mathbb{U}^k \subseteq \mathbb{N}$ uniformly
3: send $x^k$ to selected clients $i \in \mathbb{U}^k$
4: for each client $i \in \mathbb{U}^k$ in parallel
5: initialize $y_{i,0}^k \leftarrow x^k$
6: for $j = 0, 1, \ldots, \tau - 1$ do
7: $y_{i,j+1}^k \leftarrow y_{i,j}^k - \eta g_i^{j} (y_{i,j}^k)$
8: $v_{i,j+1}^k \leftarrow v_{i,j}^k - \eta g_i^{j} (v_{i,j}^k)$
9: end for
10: $v_{i,0}^{k+1} \leftarrow (1-\lambda)v_{i,\tau}^{k} + \lambda y_{i,\tau}^{k}$
11: send $y_{i,\tau}^{k} - x^k$ to server
12: end for
13: each client $i \in \mathbb{U}^k \cup \mathbb{U}^k$ updates $y_{i,0}^{k+1} \leftarrow y_{i,0}^{k}$
14: $x^{k+1} \leftarrow x^k + \frac{\rho}{2\eta} \sum_{i \in \mathbb{U}^k} (y_{i,\tau}^{k} - x^k)$
15: end for
superscripts of variables. In Line 14, the server receives and aggregates globalized local models from selected clients.

Further Discussion: (i) Regularizer: The proposal of the regularizer is inspired by FedProx. More concretely, the proximal term \( \| y - x \|^2 \) is applied to local solvers to impose restrictions on the deviation between global and local solutions in [8]. Nevertheless, the measure is conservative that only finds an inexact solution near the previous global model \( x \). In this paper, we modify the term to move the restriction near a local prediction of the current global model so as to accelerate FL convergence. (ii) More about Personalized FL: The FedDeper method can be classified as the global model personalization strategy in [14]. However, unlike the personalized FL works using the strategy to modify the training process for customization (as [10–13], global optimizing then personalizing), the proposed scheme is aimed to simultaneously accelerate the global convergence and generate the personalized model (early personalizing for global optimizing, i.e., tackling statistical heterogeneity at both global and local levels). Besides, the defined personalized model differs from the original local model defined in (2) because of their different initial policies.

IV. CONVERGENCE ANALYSIS

In this section, we analyze the convergence performance of FedDeper. To derive the pertinent result, we start by applying some common assumptions.

Assumption 1 (\( \beta \)-smooth). For any \( x, x' \in \mathbb{R}^d \), there holds \( f_i(x) \leq f_i(x') + \langle \nabla f_i(x'), x - x' \rangle + \frac{\beta}{2} \| x - x' \|^2 \).

Assumption 2 (Unbiased gradient & bounded variance). \( g_i \) is unbiased stochastic gradient, i.e., \( E[g_i] = \nabla f_i \), and its variance is uniformly bounded, i.e., \( \| g_i - \nabla f_i \|^2 \leq \varsigma^2 \).

Assumption 3 (Bounded dissimilarity). For any \( x \in \mathbb{R}^d \), there exists proper constants \( B^2 \geq 1, G^2 \geq 0 \) such that \( \frac{1}{n} \sum_{i \in N} \| \nabla f_i(x) \|^2 \leq B^2 \| \nabla f(x) \|^2 + G^2 \).

All Assumptions 1-3 are wildly used in existing literatures [7, 9, 12]. The convergence performance of the proposed FedDeper is summarized as follows.

Theorem 1. Under Assumptions 1-3, by choosing \( \lambda \in \left[ \frac{1}{2}, 1 \right] \), and \( \rho \tau \leq \eta \tau \beta \leq \min \left\{ \frac{1}{144B^2}, \frac{1}{84\sqrt{\tau} B^2 + \eta^2 B^2} \right\} \), we have

\[
\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \| \nabla f(x^k) \|^2 \leq \frac{24\bar{G} + \varsigma^2}{\eta \tau K} + 12\eta \tau \beta \left( \frac{4\bar{G}^2 + \varsigma^2}{\tau m} + \frac{24\eta^2 \rho^2 \beta^2 (35 + 7\rho)G^2 + (330 + 897 + \frac{897}{6})}{\tau m} \right) + \frac{75}{2} \left( \frac{(1 - \rho^2)G^2}{p^2} + \frac{40}{m} + \frac{280}{m} + \frac{73}{\tau} \right) + 192\eta^3 \beta^3 \left( \frac{3G^2 + \varsigma^2}{\tau} \right) + 96\rho^4 \beta^4 \left( \frac{(3\rho + 20)\varsigma^2}{p^2} \right) + 12G^2 + 576\eta^4 \beta^5 \left( \frac{4G^2 + \varsigma^2}{\tau^2} \right) + \frac{5760\eta^6 \beta^6 \varsigma^2}{p^2} \]

where \( l_p := \frac{15(1-p)^2}{49p} + \frac{75(1-p)^2}{3136p} + \frac{75(1-p)^2}{3136p} + \frac{75(1-p)^2}{3136p} + \frac{75(1-p)^2}{3136p} \).

Besides, \( p := \frac{n}{m} \) and \( q := \frac{5 + 175p}{5 + 175p} \). \( \Gamma := f(x^0) - f(x^*) \), \( B^2 := 2B^2 \left( \frac{1}{m} - \frac{1}{n} \right) + 1 \), \( G^2 := 2B^2 \left( \frac{1}{m} - \frac{1}{n} \right) \).

Corollary 1. By choosing \( \eta \leq \left( \frac{m}{n} \right)^{\frac{1}{2}} \) in Theorem 1, we have

\[
\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \| \nabla f(x^k) \|^2 \leq \mathcal{O} \left( \frac{\Gamma + \eta \tau \beta \left( \frac{4\bar{G}^2 + \varsigma^2}{\tau m} + \frac{24\eta^2 \rho^2 \beta^2 (35 + 7\rho)G^2 + (330 + 897 + \frac{897}{6})}{\tau m} \right) + \frac{75}{2} \left( \frac{(1 - \rho^2)G^2}{p^2} + \frac{40}{m} + \frac{280}{m} + \frac{73}{\tau} \right) + 192\eta^3 \beta^3 \left( \frac{3G^2 + \varsigma^2}{\tau} \right) + 96\rho^4 \beta^4 \left( \frac{(3\rho + 20)\varsigma^2}{p^2} \right) + 12G^2 + 576\eta^4 \beta^5 \left( \frac{4G^2 + \varsigma^2}{\tau^2} \right) + \frac{5760\eta^6 \beta^6 \varsigma^2}{p^2} \right) \]

where \( \mathcal{O} \) hides constants including \( \beta \), and \( G^2 := 2 + \varsigma^2 \).

Remark 1. Combining with Theorem 1 and Corollary 1, we find the smoothness parameter \( \beta \), stochastic variance \( \varsigma^2 \), the gradient dissimilarity \( G^2 \) and the sampling ratio \( p = \frac{m}{n} \) are the dominant factors affecting the convergence rate. Note that sampling ratio \( p \) contains in the crucial low-order term \( 12\eta \tau \beta (4G^2 + \frac{\varsigma^2}{m}) \) with \( G^2 |_{p=1} = 0 \), and mainly decides the impact degree of dissimilarity \( G^2 \) on the training in the dominant convergence rate \( \mathcal{O}(\frac{\Gamma}{\eta \tau K}) \). Furthermore, the penalty constant \( \rho \) also implicitly influences the convergence in choosing learning rate \( \eta \) due to the precondition \( \rho \leq \eta \beta \). Moreover, the corollary shows appropriately choosing \( \eta \) for Theorem 1 and ignoring high-order terms, the convergence bound can be scaled as \( \mathcal{O}(\frac{1}{\sqrt{\eta \tau K}}) \), which meets the sublinear rate similar to works on FedAvg and its variants [8, 9, 12]. Besides, constants in Theorem 1 are generated by technical inequality scaling. They are inexact and can be safely ignored.

Theorem 2. Let \( \frac{1}{n\tau K} \sum_{i,j,k} \| v_{i,j}^k - x^* \|^2 \) average over all the indexes \( i, j, k \), in terms of Theorem 1, (i) for any \( \lambda \in \left[ \frac{1}{2}, 1 \right] \), we have

\[
\frac{1}{n\tau K} \sum_{i,j,k} \| v_{i,j}^k - x^* \|^2 \leq \mathcal{O}(\epsilon^0) + \mathcal{O}(\epsilon),
\]

and (ii) for \( \lambda = 1 \), we have

\[
\frac{1}{n\tau K} \sum_{i,j,k} \| v_{i,j}^k - x^* \|^2 \leq \mathcal{O}(\epsilon),
\]

where \( \mathcal{O} \) hides all constants, \( \epsilon^0 := \frac{1}{\eta \tau K} \sum_{i,j,k} \mathbb{E} \| v_{i,j}^k - x^0 \|^2 \), and \( \epsilon := \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \| \nabla f(x^k) \|^2 \).

Remark 2. We here bound the gap between all personalized solutions \( \{v_{i,j}^k\}_{i,j,k} \) and the global optimum \( x^* \). The result in case (i) shows the personalized model converges around \( x^* \) in average with the radius \( \mathcal{O}(\epsilon^0) \) in terms of the initial distance. While in case (ii), we indicate that the behavior of the personalized model degenerates into the original local model, which can converge to the global optimum by choosing an infinitesimal learning rate as shown in Corollary 1.

V. EXPERIMENT RESULTS

In this section, experiment results are presented and discussed. Firstly, the experiment setup is introduced. The effects of crucial hyper-parameters (i.e., the penalty \( \rho \), the mixing rate \( \lambda \), the local step \( \tau \), and the total communication round \( K \)) on
Fig. 3. Effect of Hyper-parameters: (i) using MNIST + MLP and the sampling rate \( p = 0.5 \) with \( n = 10 \). (ii) the value of local step \( \tau = 10 \) in (a)(b). (iii) the total communication round \( K = 500 \) in (a)(b)(c) while the total iteration \( K\tau = 1500 \) in (d). (iv) the penalty \( \rho = 0.03 \) in (b)(c)(d). (v) the mixing rate \( \lambda = 0.5 \) in (a)(c)(d).

Fig. 4. Convergence Rate Comparison using MNIST in Massive-Device Scenario with \( n = 100 \): (i) the value of local step \( \tau = 10 \). (ii) the total communication round \( K = 500 \). (iii) the sampling rate \( p = (a)(b) 0.05 \), (c)(d) 0.1.

FL performance is analyzed. Besides, the aggregated model \( \mathbf{x} \) performance is analyzed in both cross-silo \( (n = 10) \) and cross-device \( (n = 100) \) scenarios. Finally, the performance and generalization property of personalized model \( \{v_i\}_{i \in \mathcal{N}} \) are shown in local testing.

**Experiment Setup:** (i) Basic Settings: Each client holds completely heterogeneous raw data generated by non-iid splits as [3] (sorted data). Due to limited bandwidth, the server can only communicate to a subset of clients per round. Besides, the learning rate \( \eta \) is always set to 0.01. (ii) Machine Learning Model and Dataset: Models: Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) (with high non-convexity) are used as the primary ML model. Datasets: MNIST and CIFAR-10 as public datasets are used to train ML models with the FL framework. Model Architectures: MLP contains 2 hidden layers with 512 and 256 neurons. While for MNIST, CNN contains 2 convolutional layers with 32 and 64 3×3 filters followed by 2 fully connected layers with 1024 and 512 neurons. For CIFAR-10, CNN contains 2 convolutional layers with 64 and 128 5×5 filters and 3 fully connected layers with 1024, 512 and 256 neurons. (iii) Baselines: We compare FedDeper with the following baselines to evaluate the convergence performance.

- **FedAvg** [3] is a canonical FL method, which is the prototype of FedDeper.
- **FedProx** [8] adds a proximal term as the regularizer to FedAvg for dealing with heterogeneity, which can be regarded as the analogue of our approach.
- **SCAFFOLD** [9], the state-of-art method that provably improves the FL performance on non-iid data via cross-client variance reduction but at the expense of double communication overhead, which similarly globalizes local gradients directly with control variables instead of personalized models.

Besides, we also provide FedDeper* defined as a version of FedDeper with half the local update steps to align the computation costs with baselines.

**Effect of Hyper-parameters:** The effects of \( \rho, \lambda, \tau \) and \( K \) on FL performance are shown in Fig. 3. In Fig. 3(a), it is observed that \( \rho \) is limited to the same order of magnitude as the learning rate \( \eta \), and a suitable setting exists to reach the performance upper bound for a particular training environment, e.g., when \( \rho = 0.03 \), the final performance is the best among the five. In Fig. 3(b), the parameter \( \lambda \) yields a similar result as in (a). Besides, the performance with the setting \( \lambda = 0.45 \) is provided to show that the range of \( \lambda \) limited in Remark 1 is sufficient but unnecessary. In Fig. 3(c), the result illustrates the effectiveness...
of additional local update steps, namely the convergence speed and final performance (w.r.t. aggregation rounds) improve as $\tau$ increases. In Fig. 3(d), the reduction of aggregation rounds leads to performance degradation, and the depersonalization mechanism alleviates the objective drift resulting in a better performance of FedDeper than the original FedAvg.

**Global Performance on Common Dataset:** Fig. 4 depicts the global training losses varying with communication rounds, which is used to compare the convergence rate of our proposed method with baselines in various settings. In specific, the proposed FedDeper is carefully tuned to reach its theoretical convergence performance, the proximal constant in FedProx is fixed to 1, and the FedAvg and SCAFFOLD methods are unmodified and have no extra hyper-parameters. As shown in Fig. 4, in the massive-device scenario with the low sampling rate, FedDeper has the lowest training loss in all cases except (a), which outperforms almost all baselines and verifies the effectiveness of FedDeper in convergence acceleration. Especially, FedDeper performs equally well as the state-of-the-art SCAFFOLD in cases (a)(c) and much better in (b)(d), with only half the communication overhead. Besides, FedDeper is stabler than SCAFFOLD and FedAvg because of the additional regularizer. Additionally, Table I summarizes the performance of all methods under fixed communication rounds in different settings (including Fig. 4, and $K = 2000$ on CIFAR-10). The results show that the proposed FedDeper has significant advantages over the baselines in most cases.

**Local Performance on Individual Dataset:** The individual testing dataset on each participantator is generated by the non-iid splits on the whole testing dataset, which holds only one label class of the data samples. Fig. 5(a) depicts the averaged local testing accuracy over all the individual datasets, where FedDeper notably has two models at each client, i.e., global model (GM) and personalized model (PM). It is illustrated that both FedDeper GM and PM improve the FL performance compared to baselines and that PM converges much faster than GM and baselines experimentally in local testing. This fact validates the statement in Theorem 2 that personalized models converge around the global model. Moreover, we randomly add extra label classes to each individual testing dataset to investigate the PM generalization performance via averaged testing accuracy. As shown in Fig. 5(b), PM performs well on the original individual testing dataset with one label class. Besides, as the number of label classes increases, the performance deteriorates since PM learns more from private data on the client than the information (about other classes) through model aggregation. Furthermore, as the communication round increases, overall performance improves in all cases, which shows that model aggregation generalizes PM and enables it to classify other label types missing from the original individual training dataset.

**VI. CONCLUSION**

In this paper, a new FL method called FedDeper was proposed to improve performance on non-iid data by reversing the personalization techniques. The corresponding convergence properties were analyzed and discussed. Also, experiment results verified the FedDeper performed better than other representative FL methods in convergence speed and model accuracy.

**REFERENCES**

[1] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, “In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning,” IEEE Network, vol. 33, no. 5, pp. 156–165, 2019.
[2] Z. Zhou, X. Chen, E. Li et al., “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, 2019.
[3] B. McMahan, E. Moore, D. Ramage et al., “Communication-efficient learning of deep networks from decentralized data,” in Artificial intelligence and statistics. PMLR, 2017, pp. 1273–1282.
[4] S. Wang, T. Turo et al., “Adaptive federated learning in resource constrained edge computing systems,” IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1205–1221, 2019.
[5] H. Yang, M. Fang, and J. Liu, “Achieving linear speedup with partial worker participation in non-iid federated learning,” in International Conference on Learning Representations, 2021.
[6] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of fedavg on non-iid data,” in International Conference on Learning Representations, 2019.
[7] A. Koloskova, N. Loizou, S. Boreiri et al., “A unified theory of decentralized sgd with changing topology and local updates,” in International Conference on Machine Learning. PMLR, 2020, pp. 5381–5393.
[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” Proceedings of Machine Learning and Systems, vol. 2, pp. 429–450, 2020.
[9] S. P. Karimireddy, S. Kale et al., “Scaffold: Stochastic controlled averaging for federated learning,” in International Conference on Machine Learning. PMLR, 2020, pp. 5132–5143.
[10] T. Li, S. Hu, A. Beirami, and V. Smith, “Ditto: Fair and robust federated learning through personalization,” in International Conference on Machine Learning. PMLR, 2021, pp. 6357–6368.
[11] A. Fallah et al., “Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,” in Advances in Neural Information Processing Systems, vol. 33, pp. 3557–3568, 2020.
[12] C. T Dinh, N. Tran, and J. Nguyen, “Personalized federated learning with moreau envelopes,” Advances in Neural Information Processing Systems, vol. 33, pp. 21394–21405, 2020.
[13] J. Mills, J. Hu, and G. Min, “Multi-task federated learning for personalised deep neural networks in edge computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 3, pp. 630–641, 2022.
[14] A. Z. Tan, H. Yu et al., “Towards personalized federated learning,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–17, 2022.
[15] L. Cai et al., “Dynamic sample selection for federated learning with heterogeneous data in fog computing,” in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6.
[16] B. Luo, W. Xiao et al., “Tackling system and statistical heterogeneity for federated learning with adaptive client sampling,” in IEEE INFOCOM, 2022, pp. 1739–1748.