Personalized Federated Learning with Parameter Propagation

Jun Wu
University of Illinois at Urbana-Champaign
junwu3@illinois.edu

Wenxuan Bao
University of Illinois at Urbana-Champaign
wbao4@illinois.edu

Elizabeth Ainsworth
USDA ARS Global Change and Photosynthesis Research Unit
University of Illinois at Urbana-Champaign
ainsworth@illinois.edu

Jingrui He
University of Illinois at Urbana-Champaign
jingrui@illinois.edu

ABSTRACT

With decentralized data collected from diverse clients, a personalized federated learning paradigm has been proposed for training machine learning models without exchanging raw data from local clients. We dive into personalized federated learning from the perspective of privacy-preserving transfer learning, and identify the limitations of previous personalized federated learning algorithms. First, previous works suffer from negative knowledge transferability for some clients, when focusing more on the overall performance of all clients. Second, high communication costs are required to explicitly learn statistical task relatedness among clients. Third, it is computationally expensive to generalize the learned knowledge from experienced clients to new clients.

To solve these problems, in this paper, we propose a novel federated parameter propagation (**FEDORA**) framework for personalized federated learning. Specifically, we reformulate the standard personalized federated learning as a privacy-preserving transfer learning problem, with the goal of improving the generalization performance for every client. The crucial idea behind **FEDORA** is to learn how to transfer and whether to transfer simultaneously, including (1) *adaptive parameter propagation:* one client is enforced to adaptively propagate its parameters to others based on their task relatedness (e.g., explicitly measured by distribution similarity), and (2) *selective regularization:* each client would regularize its local personalized model with received parameters, only when those parameters are positively correlated with the generalization performance of its local model. The experiments on a variety of federated learning benchmarks demonstrate the effectiveness of the proposed **FEDORA** framework over state-of-the-art personalized federated learning baselines.

CCS CONCEPTS

• **Information systems** → **Federated databases**; • **Computing methodologies** → *Transfer learning*.

KEYWORDS

federated learning, parameter propagation, negative transfer

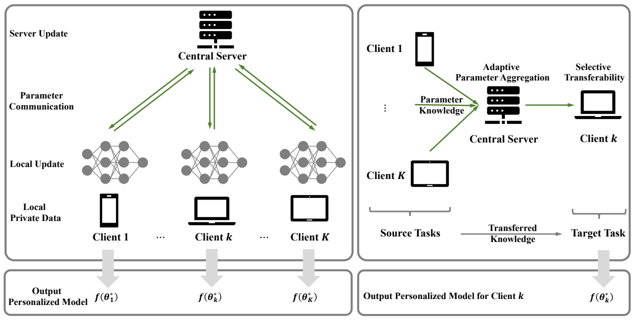
ACM Reference Format:
Jun Wu, Wenxuan Bao, Elizabeth Ainsworth, and Jingrui He. 2023. Personalized Federated Learning with Parameter Propagation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599464

1 INTRODUCTION

Federated learning [15, 25] is a learning paradigm where multiple clients collaborate in training machine learning models under the coordination of a central server. The crucial idea behind federated learning is to aggregate knowledge from diverse clients [24, 27], while protecting the privacy-sensitive data from private clients [41]. In recent years, federated learning techniques have been widely applied to a variety of high-impact domains, e.g., mobile keyboard prediction [12] and voice recognition [19] in smartphones, fMRI analysis [22] and drug discovery [3] in healthcare, etc. With decentralized data from different clients, traditional federated learning algorithms [21, 25, 38] are developed to build a global model by aggregating knowledge from all clients. But it is shown [48] that a single global model might not generalize well on the test data of each individual client when clients follow different data distributions. This motivates the paradigm of personalized federated learning [6, 24, 34], where a personalized model is learned for each client (shown in Figure 1(a)).

Most existing personalized federated learning algorithms [7, 14, 20, 23, 34, 36] consider the objective function of multi-task learning [32] by formulating the model training of each client as one task. Thus, the goal is to improve the overall performance of all the personalized models simultaneously. The intuition behind previous works is that the federated learning system focuses on improving the overall prediction performance. It cannot guarantee that all individual clients can benefit from the federated learning system. That is, some clients might have worse performance than their local training counterparts (i.e., each client trains the model over its own local data without communication across clients). This observation is verified in Figure 2, where four local clients collaborate in training models (see Figure 2(a)). It can be seen from Figure 2(b) that compared to local training (denoted as "LOCAL"), personalized federated learning approaches (e.g., LG-FedAvg [23], Ditto [20], FedAMP [13]) improve the overall prediction performance (e.g., test accuracy over all clients). However, we observe that not all

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599464



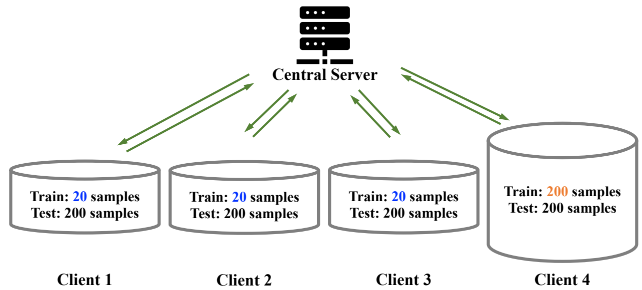
(a) Personalized federated learning (b) Privacy-preserving transfer learning

Figure 1: Illustration of personalized federated learning. (a) Personalized federated learning aims to find a personalized model for each client. (b) From the perspective of transfer learning, a client learns a personalized model by leveraging latent knowledge from other clients.

clients can benefit from federated training, e.g., client 4 has lower accuracy than LOCAL. Intuitively, this result indicates that client 4 is not incentivized to participate in federated training, because it introduces communication costs (by sharing model parameters) and achieves no performance improvement.

The observation above motivates us to re-think personalized federated learning with the following fundamental research questions. *Q1: Are all the clients incentivized to participate in federated collaboration*? *Q2: How do clients maximally benefit from federated collaboration under data heterogeneity across clients*? To answer these questions, in this paper, we study personalized federated learning by reformulating the model training of each client as a privacy-preserving transfer learning problem. As shown in Figure 1(b), for each client k, it considers itself as the target and other clients ($k' \in \{1, \cdots, k-1, k+1, \cdots, K\}$) as the sources. The goal is to improve the generalization performance of a learning algorithm on a client, by transferring the knowledge from other clients. From this point of view, we show that the aforementioned research questions are strongly correlated. That is because it is revealed [2, 40] that the target task can benefit from transfer learning when it has a limited number of training samples (*Q1*) and it shares similar data distribution with the source task (*Q1&Q2*). More specifically, on one hand, a target client is incentivized to participate in federated collaboration when it has limited training samples and there exist other clients sharing similar data distributions. This also explains that in Figure 2, with adequate training samples, Client 4 is more likely to suffer from the negative transfer (i.e., worse performance compared to LOCAL). On the other hand, a target client would collaborate with a source client sharing similar data distributions (indicating they have some common knowledge). The transferred knowledge from the source client would have a negative impact on the target learner if the distribution shift between clients is large.

Inspired by the connection between personalized federated learning and transfer learning, in this paper, we propose a novel federated parameter propagation (**FEDORA**) framework to learn personalized models in the federated learning system. The key idea of **FEDORA** is to identify whether the generalization performance of a client can benefit from the knowledge transferred from other clients, and how to maximally improve the generalization performance of



(a) Imbalanced training samples across clients

Model	Accuracy				Average Accuracy
	Client 1	Client 2	Client 3	Client 4	
LOCAL	0.5270	0.4840	0.4980	0.8110	0.5800
FedAvg	0.3755	0.4420	0.6455	0.7965	0.5649
LG-FedAvg	0.5440	0.5115	0.5430	0.8095	0.6020
Ditto	0.4095	0.4810	0.6465	0.8095	0.5866
FedAMP	0.5300	0.5210	0.5415	0.8105	0.6008
FEDORA (ours)	0.5565	0.5675	0.5850	0.8195	0.6321

(b) Results of personalized federated learning

Figure 2: Personalized federated learning on non-IID clients with imbalanced training samples. There are four clients with data drawn from Rotated MNIST [17]: Client 1 (0°), Client 2 (30°), Client 3 (60°), and Client 4 (90°), and several baselines, including FedAvg [25], LG-FedAvg [23], Ditto [20], and FedAMP [13]. The baselines suffer from the negative transfer in Client 4, i.e., lower accuracy than LOCAL.

a client by transferring the knowledge from other clients. To this end, we design two regularization terms for personalized model training. The first one is *selective regularization*, where client k updates its personalized model parameters θ_k with received knowledge (encoded by the auxiliary parameters $\hat{\theta}_k$) from other clients if $\hat{\theta}_k$ is positively correlated with the generalization performance on client k. The second regularization term is *adaptive parameter propagation*, where for client k, the transferred knowledge $\hat{\theta}_k$ is optimized based on the distribution similarity between client k and other client k' ($k' = 1, \cdots, k-1, k+1, \cdots, K$). The intuition behind adaptive parameter propagation is that two clients are more likely to have similar personalized model parameters when they are distributionally similar. Moreover, we provide theoretical generalization and convergence analysis of **FEDORA** for personalized federated learning. Extensive experiments on a variety of federated learning benchmarks demonstrate the effectiveness of our **FEDORA** framework over state-of-the-art baselines.

Compared to previous works, our proposed **FEDORA** framework has the following advantages. First, **FEDORA** can significantly alleviate the negative transfer of individual clients when participating in the federated collaboration. To the best of our knowledge, little effort (if any) has been devoted to studying negative transfer in previous works [7, 20, 33]. Second, **FEDORA** adaptively learns the transferred knowledge for each client based on the distribution similarity between this client and others. It is much more flexible than previous works [6, 20, 24] which encode the transferred knowledge with a single global model for all clients. Third, **FEDORA** has the same communication cost as vanilla FedAvg [25], which is much cheaper than previous adaptive federated learning approaches [34, 47]. Besides, we would like to point out that our work differs from existing federated transfer learning [4, 30]. In this paper, we focus on understanding standard personalized federated learning problems from a transfer learning perspective. This is in sharp contrast to the existing works which either transferred the knowledge from labeled clients to unlabeled ones [30] or fine-tuned a globally shared model [4] for personalization.

The major contributions of this paper are summarized as follows.

- We identify the negative transfer of personalized federated learning from the perspective of privacy-preserving transfer learning.
- A novel federated parameter propagation (**FEDORA**) framework is proposed for mitigating the negative transfer in personalized federated learning, followed by the theoretical convergence and generalization analysis.
- The effectiveness of **FEDORA** is confirmed in various personalized federated learning benchmarks. Besides, we show that **FEDORA** can be efficiently adapted to new clients associated with either labeled or unlabeled training samples.

The rest of this paper is organized as follows. Section 2 reviews the related work. We introduce the formal definition and major challenges of personalized federated learning in Section 3. In Section 4, we propose a novel federated parameter propagation (**FEDORA**) framework, followed by its theoretical convergence and generalization analysis. The effectiveness of **FEDORA** is empirically evaluated in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

2.1 Federated Learning

Federated learning (FL) [5, 15, 25, 49] is a learning paradigm where multiple clients collaborate in training a machine learning model under the coordination of a central server. A single model is globally trained for all clients when all the client data are independent and identically distributed (IID) [21]. But in real scenarios, it can not guarantee that all the clients collect the training samples from the same data distribution. It is found [48] that under statistical data heterogeneity among clients, a single global model might not generalize well to all the clients.

2.2 Personalized Federated Learning

Personalized federated learning [6, 20, 24] aims to learn the personalized model for every individual client. In recent years, various personalized federated learning frameworks have been proposed, including multi-task learning approaches [9, 31, 33, 34], meta-learning [8], customization regularization [36], partial parameter sharing [1, 23], etc. From the perspective of knowledge transferability, most existing algorithms consider two parameter-sharing mechanisms. One is to use a global model to encode common knowledge shared by all clients, and then regularize the personalized model of each client with this global model [6, 20, 24, 36]. The other one is to capture complex relations among individual clients, and the relation would guide the parameter sharing among clients [7, 13, 34, 47]. However, most existing algorithms focus on improving the overall performance of federated learning systems. Little effort has been devoted to studying the negative transfer in the context of personalized federated learning.

3 PRELIMINARIES

3.1 Notation

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input space and output label space respectively. In this paper, we consider the personalized federated learning setting [20, 34], where there is a central server and K local clients. Each client has access to a private training set $\{x_i^k, y_i^k\}_{i=1}^{n_k}$ drawn from a data distribution $\mathbb{P}_k$ in the k^{th} $(k = 1, \cdots, K)$ client. Here $x_i^k \in \mathcal{X}$ and $y_i^k \in \mathcal{Y}$ denote the input example and output label, respectively. The data set $\{x_i^k, y_i^k\}_{i=1}^{n_k}$ is exclusively owned by client k and will not be shared with the central server or other clients. We let $L(\cdot, \cdot)$ denote the loss function. Then the expected prediction error on client k is defined as $F_k(\theta_k) = \mathbb{E}_{(x^k, y^k)\sim\mathbb{P}_k}[L(f(x^k), y^k; \theta_k)]$ given a prediction function $f(\cdot)$, where θ_k denotes the model parameters. The empirical prediction error is then defined as $\hat{F}_k(\theta_k) = \frac{1}{n_k}\sum_{i=1}^{n_k} L(f(x^k), y^k; \theta_k)$.

3.2 Problem Definition

Following [1, 24, 34], the problem of personalized federated learning is formally defined as follows.

Definition 3.1. ***(Personalized Federated Learning)***
Input: (i) A central server; (ii) a set of local clients with private training sets; (iii) a learning algorithm $f(\cdot)$.
Output: Personalized prediction function for each client.

The goal of personalized federated learning is to learn a personalized model for each client. Most existing algorithms [7, 20, 34, 36] build the objective function based on multi-task learning, where each task is the personalized model training in one client. As a result, those works focus on improving the overall performance of all the clients. This objective cannot guarantee that all individual clients have improved performance when participating in the federated collaboration. This is also empirically confirmed in Figure 2, where some clients suffer from the negative transfer, though the overall prediction performance of federated learning is improved.

Therefore, in this paper, we would like to study personalized federated learning from the perspective of transfer learning. Regarding knowledge transferability [2, 44], it has been shown that the generalization performance of a learning algorithm on one client can be improved by leveraging latent knowledge from other relevant clients. As shown in Figure 1, personalized federated learning can be decomposed into a group of transfer learning problems [18]. For example, given a target client k, it learns the prediction function on this target client by transferring the knowledge from multiple source clients $k' \in \{1, \cdots, k-1, k+1, \cdots, K\}$ (shown in Figure 1(b)).

When evaluating the efficacy of a federated learning system, the commonly used metrics in previous works [7, 20, 34, 36] are average prediction results over all the clients, e.g., average classification accuracy for image classification [25]. In addition to average accuracy indicating the overall performance of the federated learning system, we also consider two additional evaluation metrics as follows.

Definition 3.2. ***(Relative Accuracy)*** Given a target client k, the relative accuracy of personalized federated learning is defined as

$$\text{R-Acc}\left(\theta_k^*\right) = \frac{\text{Acc}\left(\theta_k^*\right) - \text{Acc}\left(\theta_k^{\text{LOCAL}}\right)}{\text{Acc}\left(\theta_k^{\text{LOCAL}}\right)}$$

where $\text{Acc}(\cdot)$ denotes the test accuracy, θ_k^* denotes the parameters learned by personalized federated learning (i.e., with federated collaboration) on client k, and θ_k^{LOCAL} denotes the parameters learned by local training (i.e., without federated collaboration) on client k.

Definition 3.3. ***(Positive Transferability Ratio)*** Given a federated learning system with K clients, the positive transferability ratio of personalized federated learning is defined as

$$\text{PTR}\left(\theta_1^*, \cdots, \theta_K^*\right) = \frac{\sum_{k=1}^K \mathbb{I}\left[\text{Acc}\left(\theta_k^*\right) - \text{Acc}\left(\theta_k^{\text{LOCAL}}\right)\right]}{K}$$

where $\mathbb{I}[a] = 1$ if $a \geq 0$, and $\mathbb{I}[a] = 0$ otherwise.

Both relative accuracy and positive transferability ratio measure the knowledge transfer performance when client k participates in federated collaboration. Notably, the positive transferability ratio $\text{PTR}(\theta_1^*, \cdots, \theta_K^*)$ indicates whether negative transfer happens in the federated learning system. Higher $\text{PTR}(\theta_1^*, \cdots, \theta_K^*)$ implies that most clients benefit from federated collaboration. $\text{R-Acc}(\theta_k^*)$ indicates fine-grained performance improvement/degradation that client k achieves from federated collaboration.

3.3 Challenges

In addition to the data privacy and communication costs pointed out by previous works [15, 25], we identify additional challenges of personalized federated learning from the perspective of privacy-preserving transfer learning.

It is notable that each client can train a local model (termed LOCAL) over its own private training examples. By participating in federated training, a client is able to receive latent knowledge from other clients. However, the received knowledge might have a negative impact on the client learner. Following [2], this negative transfer phenomenon can be characterized by two critical factors in the context of personalized federated learning: the distribution difference between clients and the number of training samples in the client. To be specific, the distribution divergence between $\mathbb{P}_k(x, y)$ of client k and $\mathbb{P}_{k'}(x, y)$ of client k' over $\mathcal{X} \times \mathcal{Y}$ is the root of the negative transfer [40, 42]. Moreover, if there is abundant labeled data in a target client, the knowledge transferred from a slightly different source client could hurt the generalization performance. This motivates us to consider the challenge of negative knowledge transferability for each client in personalized federated learning.

4 FEDERATED PARAMETER PROPAGATION

In this section, we present a novel federated parameter propagation framework (**FEDORA**) for personalized federated learning.

4.1 Objective Function

The goal of personalized federated learning is to learn an optimal personalized model for each client, by transferring the knowledge from other relevant clients. As pointed out in previous work [2, 42, 44], when transferring the knowledge from a source task to a target task, the transfer performance is strongly correlated with both distribution differences between tasks and the number of training samples in the target task. This motivates us to propose a federated parameter propagation framework (**FEDORA**) for personalized federated learning. The overall objective function is formulated as:

$$\begin{aligned}\min_{\{\theta_k\}_{k=1}^K, \{\hat{\theta}_k\}_{k=1}^K} \mathcal{J} = \sum_{k=1}^K \frac{1}{\lambda_k n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \sum_{k=1}^K ||\theta_k - \hat{\theta}_k||_2^2 \\ + \frac{\alpha}{2} \sum_{k=1}^K \sum_{k'=1}^K \frac{w_{kk'}}{D_{kk}} \left\|\hat{\theta}_k - \hat{\theta}_{k'}\right\|_2^2\end{aligned} \tag{1}$$

where $w_{kk'}$ denotes the distribution similarity between client k and client k' and $D_{kk} = \sum_{k'=1}^K w_{kk'}$. θ_k is the personalized parameters and $\hat{\theta}_k$ is the auxiliary personalized parameters in client k. Here $\alpha > 0$ and $\lambda_k > 0$ balance different terms in our objective function.

Intuitively, the first term represents empirical prediction error for learning personalized parameters θ_k ($k = 1, \cdots, K$). The second term enforces θ_k to approximate auxiliary personalized parameters $\hat{\theta}_k$. In this case, $\hat{\theta}_k$ would encode the knowledge transferred from other clients. λ_k indicates whether client k would benefit from the received knowledge. $\lambda_k \to 0$ implies that client k would focus more on local training over its own training samples, and the received knowledge $\hat{\theta}_k$ might hurt the generalization performance of client k. The third term of our objective function is to regularize the auxiliary personalized parameters based on distribution similarity between clients. That is, when two clients are distributionally similar, they would share similar model parameters.

Remark. It can be seen that in the special case where $\hat{\theta}_k = \theta_k$ for all $k \in \{1, \cdots, K\}$ and $\lambda_1 = \cdots = \lambda_K = \lambda$, our objective function is equivalent to existing federated multi-task learning algorithms, e.g., MOCHA [34], FedU [7], with the objective function.

$$\min_{\theta_1, \cdots, \theta_K} \sum_{k=1}^K \frac{1}{n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \frac{\alpha\lambda}{2} \sum_{k=1}^K \sum_{k'=1}^K \frac{w_{kk'}}{D_{kk}} ||\theta_k - \theta_{k'}||_2^2$$

Compared to previous works [7, 34], **FEDORA** is more flexible because client k can choose whether to collaborate by dynamically adjusting λ_k during model training (see Subsection 4.2.3).

4.2 Model Training

By minimizing the objective function of Eq. (1), we iteratively update the parameters θ_k and $\hat{\theta}_k$, including (a) fixing $\hat{\theta}_k$ and updating θ_k for $k = 1, \cdots, K$ in parallel, i.e.,

$$\min_{\theta_k} \frac{1}{n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \lambda_k ||\theta_k - \hat{\theta}_k||_2^2 \tag{2}$$

and (b) fixing θ_k and updating $\hat{\theta}_k$, i.e.,

$$\min_{\hat{\theta}_1, \cdots, \hat{\theta}_K} \sum_{k=1}^K ||\theta_k - \hat{\theta}_k||_2^2 + \frac{\alpha}{2} \sum_{k=1}^K \sum_{k'=1}^K \frac{w_{kk'}}{D_{kk}} \left\|\hat{\theta}_k - \hat{\theta}_{k'}\right\|_2^2 \tag{3}$$

In this case, Eq. (2) optimizes the personalized parameters θ_k via empirical risk minimization with respect to local training data in client k. It implies that θ_k ($k = 1, \cdots, K$) can be updated locally in parallel. In contrast, Eq. (3) updates $\hat{\theta}_k$ globally, because it requires the auxiliary parameters $\hat{\theta}_{k'}$ ($k' \neq k$) from other clients. Therefore, following [15, 25], we can roughly summarize the federated training procedures of **FEDORA** as follows.

(i) *Forward Communication:* The server broadcasts the auxiliary personalized parameters to clients, e.g., send $\hat{\theta}_k$ to client k;
(ii) *Client Update:* Each client updates its own personalized parameters θ_k via Eq. (2);

(iii) *Backward Communication:* Each client uploads the personalized parameters θ_k back to the server;
(iv) *Server Update:* The server updates the auxiliary personalized parameters $\hat{\theta}_k$ ($k = 1, \cdots, K$) via Eq. (3).

We see that similar to FedAvg [25], the communication cost of **FEDORA** is determined by $\hat{\theta}_k$ and θ_k during federated training, i.e., a client shares θ_k to the central server and receives $\hat{\theta}_k$ from the server. Thus, it is much cheaper than existing federated multi-task learning algorithms, e.g., MOCHA [34], FedFOMO [47]. This is because each client in these algorithms receives the model parameters from multiple clients. Next, we present the detailed training procedures of our **FEDORA** framework.

4.2.1 Preprocessing. The intuition behind **FEDORA** framework is that two clients share similar personalized parameters if they are distributionally similar. Before introducing the iterative optimization solution of **FEDORA**, we first estimate the data distribution similarity between clients in the context of federated learning.

Inspired by [37], we measure the client similarity (i.e., $w_{kk'}$ between client k and client k') by exploring the subspaces induced by training samples across clients. As shown in [40, 43], the distribution shift across clients can be induced by both input features and output labels. As a result, we focus on estimating the client similarity over the joint data distribution $\mathcal{X} \times \mathcal{Y}$. Given a training set $X_k = \{x_1^k, \cdots, x_{n_k}^k\}$ with associated labels $Y_k = \{y_1^k, \cdots, y_{n_k}^k\}$ in client k, we perform truncated SVD on $X_k \circ Y_k$ and obtain the subspace representation $U_k = [u_1^k, \cdots, u_p^k]$ ($p \ll \text{rank}(X_k)$), i.e., $X_k \circ Y_k = U_k \Sigma_k V_k^T$. Here $\circ$ denotes the vector concatenation. Then, the similarity of two orthogonal subspaces can be measured by the principal angles. To be specific, given two subspaces $\mathcal{U}_k = \text{span}\{u_1^k, \cdots, u_p^k\}$ and $\mathcal{U}_{k'} = \text{span}\{u_1^{k'}, \cdots, u_p^{k'}\}$, the principal angles [11] are formally defined as:

$$\zeta_1^{kk'} = \min_{a_1^k \in \mathcal{U}_k, b_1^k \in \mathcal{U}_{k'}} \arccos\left(\frac{\left\langle a_1^k, b_1^k \right\rangle}{||a_1^k||||b_1^k||}\right)$$

$$\vdots$$

$$\zeta_p^{kk'} = \min_{\substack{a_p^k \in \mathcal{U}_k b_p^k \in \mathcal{U}_{k'} \\ a_p^k \perp a_1^k, \cdots, a_{p-1}^k \\ b_p^k \perp b_1^k, \cdots, b_{p-1}^k}} \arccos\left(\frac{\left\langle a_p^k, b_p^k \right\rangle}{||a_p^k||||b_p^k||}\right)$$

The orthogonal subspaces $\mathcal{U}_k$ and $\mathcal{U}_{k'}$ are identical when $\zeta_1 = \cdots = \zeta_p = 0$. Then based on the principal angles, we define the similarity between client k and client k' as follows.

$$w_{kk'} = \sum_{i=1}^{p} \cos \zeta_i^{kk'} \tag{4}$$

Previous work [11] shows that the principal angles can be efficiently calculated by the following matrix decomposition.

$$U_k^T U_{k'} = P \left(\text{diag}\left(\cos \zeta_1^{kk'}, \cdots, \cos \zeta_p^{kk'}\right)\right) \tilde{P}^T \tag{5}$$

where P and $\tilde{P}$ are orthogonal matrices by performing SVD on $U_k^T U_{k'}$ and $\cos \zeta_1^{kk'}, \cdots, \cos \zeta_p^{kk'}$ are the corresponding eigenvalues. The estimation of client similarity in federated learning is summarized in Algorithm 1 (Lines 1-5). Each client extracts the orthogonal subspace representation $U_k = [u_1^k, \cdots, u_p^k]$ over its own training samples, and then uploads U_k to the central server. The central server would estimate the pair-wise client similarity using Eq. (4) and Eq. (5).

Remark. Compared to previous works [13, 31, 46, 47], the client similarity in Eq. (4) has the following advantages. First, our client similarity measure Eq. (4) is directly correlated with the data distributions of clients. But previous works implicitly estimate the client similarity using local model parameters. Second, the estimation of Eq. (4) is computationally efficient, because it only requires one-time calculation (see Line 5 in Algorithm 1). In contrast, previous works would have to calculate the client similarity in every training round of federated learning.

Besides, Lemma 4.1 shows the connections between our framework and previous works [20, 36] by using constant client similarity.

LEMMA 4.1. *With different measures of client similarity, our objective function Eq. (1) has the following special cases.*

- *If $w_{kk'} = 1$ for $k = k'$ and $w_{kk'} = 0$ for $k \neq k'$, then we have $\hat{\theta}_k = \theta_k$. Moreover, the optimization problem of Eq. (1) becomes standard local training with the objective function*

$$\min_{\{\theta_k\}_{k=1}^K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right)$$

- *If $w_{kk'} = n_{k'}$ for $k, k' \in \{1, \cdots, K\}$ and $\alpha \to \infty$, then we have $\hat{\theta}_k = \sum_{k'=1}^{K} \frac{n_{k'}}{\sum_{j=1}^{K} n_j} \theta_{k'}$. Moreover, the optimization problem of Eq. (1) becomes customized personalized federated learning (e.g., Ditto [20]) with the objective function*

$$\min_{\{\theta_k\}_{k=1}^K} \sum_{k=1}^{K} \left(\frac{1}{n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \lambda_k ||\theta_k - w||_2^2\right)$$

where $w = \sum_{k'=1}^{K} \frac{n_{k'}}{\sum_{j=1}^{K} n_j} \theta_{k'}$ is the weighted personalized parameters.

4.2.2 Parameter Propagation for Server Update. The central server updates the auxiliary personalized parameters $\hat{\theta}_k$ ($k = 1, \cdots, K$) by minimizing the sub-problem Eq. (3) of our objective function. The following lemma shows that Eq. (3) has a closed-form solution.

LEMMA 4.2. *Let $\Theta = [\theta_1, \theta_2, \cdots, \theta_K]^T \in \mathbb{R}^{K \times d_\theta}$ and $\hat{\Theta} = [\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_K]^T \in \mathbb{R}^{K \times d_\theta}$ be the personalized model parameters and auxiliary personalized model parameters respectively, where d_θ denotes the dimensionality of model parameters. The optimal solution to the objective of Eq. (3) satisfies the equation*

$$\hat{\Theta}^* = (1 - \kappa)\left(I - \kappa D^{-1} W\right)^{-1} \Theta \tag{6}$$

where $\kappa = \frac{\alpha}{1+\alpha}$. It has an equivalent iterative solution:

$$\hat{\Theta}^{(m)} = \left(\kappa D^{-1} W\right) \hat{\Theta}^{(m-1)} + (1 - \kappa)\,\Theta \tag{7}$$

where $\hat{\Theta}_k^{(0)} = \Theta$. Moreover, $\hat{\Theta}^{(m)}$ converges, i.e., $\lim_{m \to \infty} \hat{\Theta}^{(m)} = \hat{\Theta}^*$, when m goes to infinity.

Lemma 4.2 illustrates the intuition behind server updates of **FEDORA**. To be specific, Eq. (7) can be rewritten as

$$\hat{\theta}_k^{(m)} = \frac{\alpha}{(1+\alpha) D_{kk}} \sum_{k'=1}^{K} w_{kk'} \hat{\theta}_{k'}^{(m-1)} + \frac{1}{1+\alpha} \theta_k$$

We see that client k would be more likely to iteratively aggregate the knowledge from client k', when they have higher distribution similarity $w_{kk'}$. Moreover, similar to personalized PageRank [29], $\hat{\theta}_k$ has probability $\frac{1}{1+\alpha}$ of being updated via its counterpart θ_k, and probability $\frac{\alpha}{1+\alpha}$ of being updated using other clients.

4.2.3 Personalized Training for Client Update. When receiving the auxiliary personalized parameters $\hat{\theta}_k$ from the server, client k would update its personalized model parameters θ_k by minimizing the sub-problem Eq. (2) of our objective function. In this case, λ_k can be considered as an indicator to show whether the transferred knowledge $\hat{\theta}_k$ can benefit the personalized model training of client k. From the perspective of transfer learning [2, 44], the goal of personalized learning on client k is to improve the generalization performance by leveraging the knowledge from other clients (encoded by $\hat{\theta}_k$). Therefore, we define the selection parameter λ_k by empirically evaluating the generalization performance of auxiliary personalized parameters $\hat{\theta}_k$.

$$\lambda_k = \max\left(\epsilon, \tilde{L}_k\left(\theta_k\right) - \tilde{L}_k(\hat{\theta}_k)\right) \tag{8}$$

where $\tilde{L}_k(\cdot)$ denotes the prediction error on the validation set of client k and $\epsilon > 0$ is a constant ($\epsilon = 1e-8$ used in the experiments). The intuition behind Eq. (8) is that the auxiliary personalized parameters $\hat{\theta}_k$ will guide the personalized model training of client k, only when $\hat{\theta}_k$ can generalize better than θ_k. In the experiments, we show that this simple selection strategy over λ_k can largely mitigate the negative transfer of local clients.

When λ_k is learned, client k would update its personalized parameters θ_k using standard gradient descent as follows.

$$\theta_k \leftarrow \theta_k - \eta\nabla_{\theta_k}\left(\frac{1}{n_k}\sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \lambda_k||\theta_k - \hat{\theta}_k||_2^2\right) \tag{9}$$

where $\eta > 0$ is the learning rate.

The overall training procedures of **FEDORA** are summarized in Algorithm 1. It first estimates the client similarity based on the data distributions of local clients. Then **FEDORA** iteratively updates the personalized parameters θ_k and auxiliary parameters $\hat{\theta}_k$ by minimizing the overall objective function in Eq. (1).

4.3 Generalization to New Clients

In addition to standard model training, we show that our **FEDORA** framework can easily generalize to new clients. In this paper, we consider two federated learning scenarios: (1) new clients have labeled training samples, and (2) new clients only have unlabeled training samples. In the first case, the new client extracts the orthogonal subspace representation U_{K+1} and then uploads U_{K+1} to the central server. The server would estimate the distribution similarity between this new client and other clients. Based on the distribution similarity, the server calculates the auxiliary parameters $\hat{\theta}_{K+1}$ via Eq. (6) and sends it back to the new client. Finally, the new client can optimize its personalized parameters θ_{K+1} (see Eq. (9)) by regularizing θ_{K+1} with the received auxiliary parameters $\hat{\theta}_{K+1}$. We summarize the training procedures in Algorithm 2.

In the second scenario where new clients only have unlabeled training samples, these clients do not support building pure locally trained models or fine-tuning the received parameters from the federated learning system. We show that our **FEDORA** framework can generate the auxiliary parameters $\hat{\theta}_{K+1}$ for the new client under mild assumptions. If all the clients follow the covariate shift

Algorithm 1 Federated Parameter Propagation (**FEDORA**)

Input: K private clients with data $\{x_i^k, y_i^k\}_{i=1}^{n_k}$ $(k = 1, \cdots, K)$, a learning algorithm $f(\cdot)$.
Output: Personalized model parameters $\{\theta_k\}_{k=1}^K$

```
for client k = 1, ..., K in parallel do
    Compute base vectors U_k of subspace in client k
    Upload U_k to the central server
end for
Estimate client similarity w_kk' via Eq. (4) on the server
Initialize personalized parameters θ_k on local client
Initialize auxiliary parameters θ_k' on central server
for each round r = 0, 1, ..., do
    for client k = 1, ..., K in parallel do
        Estimate λ_k using Eq. (8)
        for local epoch i = 1, ..., E do
            Update personalized parameters θ_k using Eq. (9)
        end for
        Upload θ_k to the central server
    end for
    Update auxiliary parameters using Eq. (6) or Eq. (7)
    Send updated auxiliary parameters back to local clients
end for
```

Algorithm 2 Generalization to a New Client

Input: A new client with data $\{x_i^{K+1}, y_i^{K+1}\}_{i=1}^{n_{K+1}}$
Output: Personalized model parameter θ_{new}

```
Compute base vectors U_{K+1} of subspace in the new client
Upload U_{K+1} to the central server
Estimate client similarity between this new client and old ones
Calculate the auxiliary parameters θ̂_{K+1}
Send the auxiliary parameters θ̂_{K+1} to the new client
Update personalized parameters θ_{K+1} using Eq. (9)
```

assumption [2], i.e., they have the same labeling function $\mathbb{P}_k(y|x) = \mathbb{P}_{k'}(y|x)$ but different marginal distributions $\mathbb{P}_k(y|x) \neq \mathbb{P}_{k'}(y|x)$, we can estimate the client similarity based on the subspace representation U_k induced by features X_k (i.e., $X_k = U_k\Sigma_k V_k^T$). In this case, when a new client with unlabeled training samples appears, **FEDORA** can estimate the client similarity between this new client and other ones based on the feature-guided subspace representation. Then the parameter propagation mechanism in Eq. (6) would generate the auxiliary parameters $\hat{\theta}_{K+1}$ for the new client. The objective function of Eq. (2) shows that when no labeled training samples are available on the new client, it achieves the optimal solution at $\theta_{K+1} = \hat{\theta}_{K+1}$ (see more empirical analysis in Subsection 5.3.1).

4.4 Discussion

In this subsection, we analyze the convergence and generalization performance of **FEDORA** for personalized federated learning.

We first study the convergence of **FEDORA**. The objective function of **FEDORA** can be rewritten as follows.

$$\sum_{k=1}^{K}\left(\frac{1}{\lambda_k}F_k(\theta_k) + \left\|\theta_k - \hat{\theta}_k\right\|_2^2\right) + \frac{\alpha}{2}G\left(\{\hat{\theta}_k\}_{k=1}^K\right) \tag{10}$$

where $F_k(\theta_k) = \mathbb{E}_{(x^k,y^k)\sim\mathbb{P}_k}[L(x^k,y^k;\theta_k)]$ and $G\left(\{\hat{\theta}_k\}_{k=1}^K\right) = \sum_{k=1}^K\sum_{k'=1}^K \frac{w_{kk'}}{D_{kk}}\left\|\hat{\theta}_k-\hat{\theta}_{k'}\right\|_2^2$.

Before analyzing the model convergence, we first introduce some assumptions commonly used in federated learning [7, 36].

Assumption 1 (Strong convexity of F_k). *$F_k(\theta_k)$ is μ-strongly convex w.r.t. θ_k, $\forall k$. That is, for any θ,θ',*

$$F_k(\theta') \geq F_k(\theta) + \nabla F(\theta)^\top(\theta'-\theta) + \frac{\mu}{2}\|\theta-\theta'\|_2^2$$

Assumption 2 (δ-approximate solution). *In each round r, after local updates, the learned parameters $\theta_k(r)$ approximate the optimal parameters $\theta_k^*(r) = \arg\min_{\theta_k} F_k(\theta_k) + \lambda_k\|\theta_k - \hat{\theta}_k(r)\|_2^2$ as follows.*

$$\|\theta_k(r) - \theta_k^*(r)\|_2 \leq \delta$$

Theorem 4.3. *Let $\Theta = [\theta_1,\theta_2,\cdots,\theta_K]^T$ and $\hat{\Theta} = [\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_K]^T$ be the personalized model parameters and auxiliary personalized model parameters respectively. With Assumptions 1 and 2 above, after R training rounds, we have*

$$\|\Theta(R)-\Theta^*\|_F \leq \left(\frac{2\lambda}{\mu+2\lambda}\right)^R \|\Theta(0)-\Theta^*\|_F + \frac{\mu+2\lambda}{\mu}\sqrt{K}\delta$$

where $\lambda = \max\{\lambda_1,\cdots,\lambda_K\}$, $\Theta(0)$ is the initialized personalized model parameters, and Θ^* is the global minimizer of the objective in Eq. (10).

It can be seen from Theorem 4.3 that the estimation error linearly converges with bounded error.

Then we derive the generalization error of **FEDORA** for personalized federated learning.

Assumption 3 (Smoothness). *For each $k \in \{1,\cdots,K\}$, F_k is ν-smooth, i.e., for any parameters θ,θ',*

$$\left\|\nabla F_k(\theta)-\nabla F_k(\theta')\right\|_2 \leq \nu\left\|\theta-\theta'\right\|_2$$

Theorem 4.4. *With Assumption 3 and bounded loss function $L(\cdot,\cdot)$, i.e., $L(x,y)\leq M$ for any example (x,y) within all the clients, if the expected local minimizer $\bar{\theta}_k$ of client k ($k=1,\cdots,K$) is given by $\bar{\theta}_k = \arg\min_{\theta_k}\mathbb{E}_{(x^k,y^k)\sim\mathbb{P}_k}\left[L\left(x^k,y^k;\theta_k\right)\right]$, and the empirical local minimizer θ_k^* of client k is given by the objective function in Eq (1), then for any $\delta\in(0,1)$, with probability at least $1-\delta'$, the following holds*

$$\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right] \leq \mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right] + \Omega + \frac{1}{2}\sum_{k'=1}^K \frac{w_{kk'}}{D_{kk'}}\left(\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2 + 3d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)$$

where $\Omega = 2\sqrt{\left(\sum_{k'=1}^K \frac{\alpha_{kk'}^2\left(2d\log 2(n+1)+\log\frac{4}{\delta'}\right)}{n_{k'}}\right)} + \frac{5}{2}M\sqrt{\frac{\log 4/\delta'}{2n_k}}$ is the sample complexity term and $d_{\mathcal{Y}}(\cdot,\cdot)$ is $\mathcal{Y}$-discrepancy [26] indicating the distribution difference between clients, i.e., $d_{\mathcal{Y}}(\mathbb{P}_k,\mathbb{P}_{k'}) = \sup_{f\in\mathcal{F}}|\mathbb{E}_{(x,y)\sim\mathbb{P}_k}[L(f(x),y)] - \mathbb{E}_{(x,y)\sim\mathbb{P}_{k'}}[L(f(x),y)]|$.

Theorem 4.4 shows that the expected prediction error of client k is bounded in terms of the distribution distance $d_{\mathcal{Y}}(\mathbb{P}_k,\mathbb{P}_{k'})$ between clients. It indicates that the error bound on client k can be empirically minimized by assigning large weight $w_{kk'}$ to the client k' when they have a small distribution distance. This is consistent with Subsection 4.2.1, where we define the weight $w_{kk'}$ explicitly based on the distribution similarity of client k and client k'.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Data Sets. In the experiments, we use the following data sets: MNIST [17], Fashion-MNIST [45], CIFAR-10 [16], Yearbook [10], GTSRB [35], and agriculture data [28, 39]. Following [9, 25], we consider two methods to partition the non-IID data over clients. For MNIST, Fashion-MNIST, and GTSRB, we partition the training images into K clients, where the images in each client are rotated with a certain angle. For the rotated MNIST, Fashion-MNIST, and GTSRB, the data heterogeneity among clients is induced by the feature shift. There are 36, 72, and 10 clients in rotated MNIST, Fashion-MNIST, and GTSRB respectively. For CIFAR-10, we follow the pathological non-IID setting where each client has data with at most two classes. In addition, Yearbook consists of 37921 frontal-facing American high school yearbook photos from 1930 to 2013 for gender classification. Agriculture data sets contain maize the soybean data collected from Illinois and Nebraska over the years. The task in the agriculture data sets is to predict diverse traits (e.g., Nitrogen) of plants related to the plants' growth using leaf hyperspectral reflectance. Therefore, Yearbook and agriculture data sets can be naturally partitioned into different clients based on the data collection time. Then Yearbook has 84 clients and Agriculture has 11 clients. Data heterogeneity exists in Yearbook and agriculture data sets because the underlying sampling distribution might be changing over time.

5.1.2 Baselines. The baselines used in the experiments include (global) federated learning approaches: FedAvg [25], FedProx [21] and their variants with fine-tuning (FedAvg+FT and FedProx+FT), and the following personalized federated learning approaches.

- LOCAL: Each client trains its own personalized model without knowledge communication.
- Parameter Decoupling: LG-FedAvg [23], FedPer [1] and pFedHN [33] partially share the model parameters indicating the shared common knowledge among clients.
- Model Interpolation: APFL [6] and Ditto [20] learn personalized models using a mixture of global and local models.
- Clustering: IFCA [9] and FeSEM [46] alternately estimate the cluster identities and optimize model parameters for each cluster.
- Multi-Task Learning: FedFOMO [47], FedAMP [13] and FedU [7] explicitly capture relationships among the clients with different data distributions.

5.1.3 Model Configuration. In the experiments, we use a 3-layer MLP for MNIST, Fashion-MNIST, Yearbook and agriculture data sets. A 5-layer CNN is adopted for CIFAR-10 and GTSRB. We use cross-entropy loss for image classification (MNIST, Fashion-MNIST, Yearbook, CIFAR-10 and GTSRB) and mean square error as the loss function for agriculture analysis. In addition, we set $p = 1$ and $\alpha = 1$ in the experiments. All the experiments are performed on a Windows machine with four 3.80GHz Intel Cores, 64GB RAM, and two NVIDIA Quadro RTX 5000 GPUs.

5.2 Results

Table 1 and Table 2 provide the results of personalized federated learning on image and agriculture data sets (the best results are indicated in bold) where each client has the same number of training samples. As illustrated in Subsection 3.2, we report the average

Model	Rotated MNIST			Rotated Fashion-MNIST			CIFAR-10			Yearbook			Rotated GTSRB		
	Acc ↑	R-Acc ↑	PTR ↑	Acc ↑	R-Acc ↑	PTR ↑	Acc ↑	R-Acc ↑	PTR ↑	Acc ↑	R-Acc ↑	PTR ↑	Acc ↑	R-Acc ↑	PTR ↑
LOCAL	0.7642	-	-	0.7057	-	-	0.7617	-	-	0.8068	-	-	0.5531	-	-
FedAvg [25]	0.6889	-0.0976	0	0.6441	-0.0847	0.1250	0.6531	-0.1382	0.3000	0.8165	0.0191	0.5119	0.6375	0.2006	0.8000
FedAvg+FT	0.7411	-0.0293	0.3056	0.6848	-0.0283	0.3472	0.7992	0.0513	0.9000	0.8180	0.0195	0.5595	0.6375	0.2185	0.8000
FedProx [21]	0.5375	-0.2962	0	0.5968	-0.1521	0	0.6984	-0.0799	0.2000	0.7995	-0.0027	0.4405	0.7031	0.3384	0.9000
FedProx+FT	0.6893	-0.0973	0.0278	0.6788	-0.0358	0.3056	0.7953	0.0460	0.9000	0.8227	0.0258	0.5595	0.7312	**0.3984**	0.9000
LG-FedAvg [23]	0.7804	0.0214	0.9444	0.7137	0.0115	0.7361	0.7656	0.0054	0.8000	0.8072	0.0007	0.8095	0.5938	0.1044	0.8000
FedPer [1]	0.7741	0.0135	0.6389	0.6725	-0.0457	0.1389	0.8352	0.0990	**1.0000**	0.7974	-0.0096	0.4167	0.6687	0.2607	0.8000
pFedHN [33]	0.8004	0.0486	0.8611	0.7215	0.0249	0.6944	0.7766	0.0221	0.6000	0.8263	0.0313	0.6310	0.4500	-0.1778	0.2000
APFL [6]	0.7871	0.0303	0.8889	0.7134	0.0112	0.7639	0.8258	0.0866	0.9000	0.8128	0.0081	0.7619	0.6469	0.1995	0.9000
Ditto [20]	0.7806	0.0220	0.7222	0.7212	0.0232	0.7361	0.8078	0.0630	0.9000	0.8148	0.0112	0.6429	0.7063	0.3345	0.8000
IFCA [9]	0.7915	0.0365	0.6944	0.7305	0.0370	0.7639	0.8227	0.0828	0.9000	0.8122	0.0076	0.5238	0.7344	0.3852	**1.0000**
FeSEM [46]	0.7720	0.0110	0.6111	0.7074	0.0051	0.5278	0.8547	0.1255	**1.0000**	0.7821	-0.0258	0.3810	0.6562	0.2274	0.8000
FedFOMO [47]	0.7749	0.0140	0.9167	0.7110	0.0076	0.7639	0.8242	0.0797	**1.0000**	0.8111	0.0059	0.7619	0.6156	0.1321	**1.0000**
FedU [7]	0.7837	0.0260	0.8889	0.7208	0.0225	0.8056	0.7836	0.0295	0.9000	0.8092	0.0047	0.5357	0.5625	0.0549	0.6000
FedAMP [13]	0.7869	0.0298	**1.0000**	0.7203	0.0213	0.8056	0.7953	0.0457	0.8000	0.8111	0.0059	0.6905	0.5625	0.0233	0.6000
FEDORA	**0.8251**	**0.0806**	**1.0000**	**0.7433**	**0.0548**	**0.9028**	**0.8570**	**0.1288**	**1.0000**	**0.8341**	**0.0386**	**0.9167**	**0.7375**	0.3773	**1.0000**

Table 1: Results on image data sets (Acc: average accuracy, R-Acc: average relative accuracy, PTR: positive transferability ratio)

Figure 3: Relative performance improvement of each client on Rotated MNIST

Model	MAE ↓	R-MAE ↑	PTR ↑
LOCAL	0.5576	-	-
FedPer [1]	0.4399	0.1224	0.6364
pFedHN [33]	0.4262	0.1289	0.6364
APFL [6]	0.4263	0.1398	0.8182
Ditto [20]	0.4331	0.1281	0.8182
FedFOMO [47]	0.4488	0.0973	0.8182
FedU [7]	0.5421	0.0289	0.8182
FedAMP [13]	0.4433	0.1103	0.8182
FEDORA	**0.4185**	**0.1499**	**0.9091**

Table 2: Results on agriculture data set

classification accuracy, average relative accuracy, and positive transferability ratio for image classification in Table 1. For the regression task in the agriculture data set, we use the Mean Absolute Error (MAE) between the predicted outputs and the ground-truth outputs (i.e., Nitrogen content). Similarly, we report the average MAE, average relative MAE, and positive transferability ratio in Table 2. We have the following observations. (1) Higher accuracy does not imply that all the clients benefit from federated collaboration (e.g., IFCA [9] obtains much higher accuracy on Rotated MNIST than LF-FedAvg [23], but it suffers from negative transfer on local clients). Relative accuracy (relative MAE) and positive transferability ratio can better characterize whether the negative transfer happens in local clients. (2) The proposed **FEDORA** framework achieves comparable average accuracy (MAE) but a much better positive transferability ratio than state-of-the-art baselines.

Model	Rotated MNIST			Rotated Fashion-MNIST		
	Acc ↑	R-Acc ↑	PTR ↑	Acc ↑	R-Acc ↑	PTR ↑
LOCAL	0.7736	-	-	0.7079	-	-
FedAvg [25]	0.5961	-0.2310	0.1944	0.5156	-0.2697	0.0694
FedAvg+FT	0.7631	-0.0131	0.3333	0.6671	-0.0559	0.2083
FedProx [21]	0.5564	-0.2826	0.1389	0.4529	-0.3590	0.0417
FedProx+FT	0.7111	-0.0805	0.1944	0.6387	-0.0955	0.1111
LG-FedAvg [23]	0.7916	0.0240	0.8611	0.7187	0.0154	0.7500
FedPer [1]	0.7676	-0.0071	0.4722	0.6599	-0.0654	0.1806
pFedHN [33]	0.7927	0.0254	0.6667	0.7254	0.0275	0.7083
APFL [6]	0.7934	0.0262	0.8889	0.7219	0.0204	0.8056
Ditto [20]	0.7908	0.0231	0.5000	0.6738	-0.0464	0.2083
IFCA [9]	0.8263	0.0693	0.8889	0.7356	0.0414	0.7500
FeSEM [46]	0.7876	0.0183	0.5833	0.7171	0.0154	0.6389
FedFOMO [47]	0.7959	0.0293	0.8611	0.7225	0.0213	0.7639
FedU [7]	0.7928	0.0255	0.8333	0.7229	0.0226	0.7778
FedAMP [13]	0.7906	0.0228	0.9167	0.7228	0.0217	0.8194
FEDORA	**0.8366**	**0.0828**	**1.0000**	**0.7466**	**0.0562**	**0.9444**

Table 3: Impact of imbalanced samples among clients

As discussed in Subsection 3.3, the number of training samples in one client might affect whether it can benefit from federated

Model	Acc ↑	R-Acc ↑	PTR ↑
FEDORA with random similarity	0.7657	0.0020	0.7222
FEDORA with parameter similarity	0.8003	0.0477	0.9444
FEDORA with gradient similarity	0.8125	0.0637	0.9722
FEDORA	0.8251	0.0806	1.0000

Table 4: Impact of client similarity measure on FEDORA

Model	Acc ↑	R-Acc ↑	PTR ↑
FEDORA with $\lambda_k = 0.01$	0.7752	0.0021	0.8611
FEDORA with $\lambda_k = 0.1$	0.8116	0.0498	0.9722
FEDORA with $\lambda_k = 1$	0.8256	0.0688	0.9722
FEDORA with $\lambda_k = 10$	0.8038	0.0408	0.8333
FEDORA	0.8366	0.0828	1.0000

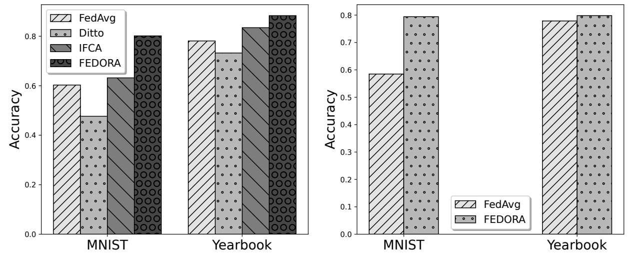
Table 5: Impact of λ_k on FEDORA

collaborations. As a result, we empirically investigate the impact of the number of training samples in personalized federated learning. Table 3 reports the image classification results on Rotated MNIST and Fashion-MNIST when one client (e.g., client 18 in Rotated MNIST) has much more training samples than others. It shows that when using the average model parameters (e.g., learned by FedAvg [25]) to regularize the local model, personalized federated learning approaches (e.g., Ditto [20], FedPer [1]) has lower positive transferability ratio. This is because the shared average model parameters would significantly bias toward the client with a large number of training samples. We visualize the relative performance improvement of local clients in Figure 3. It confirms that Ditto and FedPer can only achieve satisfactory performance on clients when they have similar distributions as client 18. In contrast, **FEDORA** can mitigate the negative transfer for all the clients.

5.3 Analysis

5.3.1 Generalization to New Clients. We show in Subsection 4.3 that **FEDORA** can be adapted to new clients associated with either labeled or unlabeled training samples. Figure 4 provides the results on Rotated MNIST by adapting the federated learning models to new clients, where the classification accuracy on new clients is reported. We observe from the results that **FEDORA** achieves better prediction performance when the new client has labeled training samples for fine-tuning (Figure 4(a)). When the new client only has unlabeled training samples, most existing approaches [9, 33] fail to adapt the trained federated learning system to the new client. FedAvg [25] can simply share the global model with the new client, but it does not consider the data heterogeneity between the new client and the old ones. In contrast, **FEDORA** learns the personalized auxiliary parameters based on the client similarity. Figure 4(b) confirms the superior performance of **FEDORA** over FedAvg.

5.3.2 Impact of Client Similarity Measure. We study the impact of client similarity measurements on the proposed **FEDORA** framework. More specifically, we consider several methods to estimate client similarity, including random client similarity [7], parameter similarity [13], and gradient similarity [31]. Table 4 shows the personalized federated learning results. It can be seen that when estimating the client similarity based on data distribution, **FEDORA** achieves better prediction performance and positive transferability ratio. The coordinate-wise parameter/gradient similarity cannot accurately measure the client similarity due to the permutation invariance of neural network parameters [38].



(a) New labeled client (b) New unlabeled client

Figure 4: Generalization to the new client where the new client has (a) labeled training samples; (b) unlabeled training samples

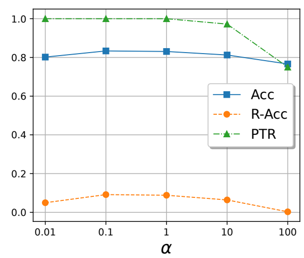


Figure 5: Hyper-parameter sensitivity

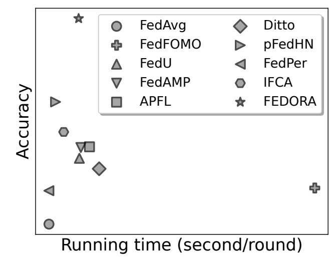


Figure 6: Computational efficiency

5.3.3 Hyper-parameter Sensitivity. Figure 5 shows the impact of hyper-parameter α on the proposed **FEDORA** framework. **FEDORA** can achieve better performance when $\alpha \in [0.1, 1]$. Moreover, we investigate the impact of λ_k in Eq. (8). Table 5 shows the results by instantiating all λ_k with a constant value on Roated MNIST. It indicates that a single constant value cannot identify whether all the clients would benefit from federated collaboration, especially when some clients have a large number of training samples.

5.3.4 Computational Efficiency. We compare the computational efficiency of **FEDORA** with baselines. Figure 6 shows that **FEDORA** is computationally efficient compared to other multi-task learning baselines, e.g., FedFOMO [47], FedAMP [13], which estimate the client relationship in every training round.

6 CONCLUSION

In this paper, we study personalized federated learning from the perspective of transfer learning. To mitigate the negative transfer issue for each client, we propose a novel federated parameter propagation (**FEDORA**) framework with an adaptive parameter propagation mechanism and a simple selective regularization. The efficacy of **FEDORA** is analyzed theoretically and empirically.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, IIS-2137468, and Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

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

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79 (2010), 151–175.

[3] Shaoqi Chen, Dongyu Xue, Guohui Chuai, Qiang Yang, and Qi Liu. 2021. FL-QSAR: a federated learning-based QSAR prototype for collaborative drug discovery. *Bioinformatics* 36, 22-23 (2021), 5492–5498.

[4] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems* 35, 4 (2020), 83–93.

[5] Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. 2022. To Federate or Not To Federate: Incentivizing Client Participation in Federated Learning. *arXiv preprint arXiv:2205.14840* (2022).

[6] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461* (2020).

[7] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. 2022. A New Look and Convergence Rate of Federated Multitask Learning With Laplacian Regularization. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[8] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems* (2020).

[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[10] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. 2015. A Century of portraits: A visual historical record of American high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*.

[11] Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

[12] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).

[13] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data.. In *AAAI*. 7865–7873.

[14] Wonyong Jeong and Sung Ju Hwang. 2022. Factorized-FL: Personalized Federated Learning with Parameter Factorization & Similarity Matching. In *Advances in Neural Information Processing Systems*.

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998).

[18] Joshua Lee, Prasanna Sattigeri, and Gregory Wornell. 2019. Learning new tricks from old dogs: Multi-source transfer learning from pre-trained networks. *Advances in neural information processing systems* 32 (2019).

[19] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. 2019. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6341–6345.

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

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[22] Xiaoxiao Li, Yufeng Gu, Nicha Dvornek, Lawrence H Staib, Pamela Ventola, and James S Duncan. 2020. Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. *Medical Image Analysis* 65 (2020), 101765.

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

[24] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* (2020).

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*.

[26] Mehryar Mohri and Andres Muñoz Medina. 2012. New analysis and algorithm for learning with drifting distributions. In *Algorithmic Learning Theory*.

[27] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic federated learning. In *International Conference on Machine Learning*.

[28] Christopher M Montes, Carolyn Fox, Álvaro Sanz-Sáez, Shawn P Serbin, Etsushi Kumagai, Matheus D Krause, Alencar Xavier, James E Specht, William D Beavis, Carl J Bernacchi, et al. 2022. High-throughput characterization, correlation, and mapping of leaf photosynthetic and functional traits in the soybean (Glycine max) nested association mapping population. *Genetics* 221, 2 (2022).

[29] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report.

[30] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. 2020. Federated Adversarial Domain Adaptation. In *ICLR*.

[31] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on neural networks and learning systems* (2020).

[32] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 31 (2018).

[33] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. 2021. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*. PMLR, 9489–9502.

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

[35] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2011. The German traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*.

[36] Canh T Dinh, Nguyen Tran, and Josh Nguyen. 2020. Personalized federated learning with Moreau envelopes. *Advances in Neural Information Processing Systems* 33 (2020), 21394–21405.

[37] Saeed Vahidian, Mahdi Morafah, Chen Chen, Mubarak Shah, and Bill Lin. [n. d.]. Rethinking Data Heterogeneity in Federated Learning: Introducing a New Notion and Standard Benchmarks. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

[38] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *ICLR*.

[39] Sheng Wang, Kaiyu Guan, Zhihui Wang, Elizabeth A Ainsworth, Ting Zheng, Philip A Townsend, Kaiyuan Li, Christopher Moller, Genghong Wu, and Chongya Jiang. 2021. Unique contributions of chlorophyll and nitrogen to predict crop photosynthetic capacity from leaf spectroscopy. *Journal of experimental botany* 72, 2 (2021), 341–354.

[40] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. Characterizing and avoiding negative transfer. In *CVPR*. 11293–11302.

[41] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[42] Jun Wu and Jingrui He. 2021. Indirect invisible poisoning attacks on domain adaptation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1852–1862.

[43] Jun Wu and Jingrui He. 2022. Domain adaptation with dynamic open-set targets. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2039–2049.

[44] Jun Wu, Jingrui He, Sheng Wang, Kaiyu Guan, and Elizabeth Ainsworth. 2022. Distribution-informed neural networks for domain adaptation regression. In *Advances in Neural Information Processing Systems*.

[45] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[46] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, Jing Jiang, and Chengqi Zhang. 2020. Multi-center federated learning. *arXiv preprint arXiv:2005.01026* (2020).

[47] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. 2021. Personalized Federated Learning with First Order Model Optimization. In *ICLR*.

[48] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

[49] Yao Zhou, Jun Wu, Haixun Wang, and Jingrui He. 2022. Adversarial robustness through bias variance decomposition: A new perspective for federated learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2753–2762.

A APPENDIX

A.1 Proof of Lemma 4.1

Proof. In the first case, if $w_{kk'} = 1$ for $k = k'$ and $w_{kk'} = 0$ for $k \neq k'$, our objective function Eq. (1) becomes

$$\min_{\{\theta_k\}_{k=1}^K, \{\hat{\theta}_k\}_{k=1}^K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} L\left(x_i^k, y_i^k; \theta_k\right) + \lambda_k \left(\sum_{k=1}^{K} ||\theta_k - \hat{\theta}_k||_2^2\right)$$

We see that the optimal solution is achieved at $\hat{\theta}_k = \theta_k$ for all client k. Each client updates its own personalized parameters θ_k locally without communication with others. In the second case, if $w_{kk'} = n_{k'}$ for $k, k' \in \{1, \cdots, K\}$ and $\alpha \to \infty$, using Lemma 4.2,

$$\hat{\theta}_k^{(m)} = \frac{\alpha}{(1+\alpha)D_{kk}} \sum_{k'=1}^{K} w_{kk'} \hat{\theta}_{k'}^{(m-1)} + \frac{1}{1+\alpha}\theta_k \to \sum_{k'=1}^{K} \frac{n_{k'}}{\sum_{j=1}^K n_j} \hat{\theta}_{k'}^{(m-1)}$$

Since $\hat{\theta}_{k'}^0 = \theta_{k'}$, we have $\hat{\theta}_k^{(m)} \to \sum_{k'=1}^K \frac{n_{k'}}{\sum_{j=1}^K n_j} \theta_{k'}$ when $\alpha \to \infty$. That is, all clients share the same auxiliary parameters $w = \sum_{k'=1}^K \frac{n_{k'}}{\sum_{j=1}^K n_j} \theta_{k'}$. □

A.2 Proof of Lemma 4.2

Proof. The problem Eq. (3) can be re-written as:

$$\min_{\hat{\Theta}} \mathrm{Tr}\left((\Theta - \hat{\Theta})^T (\Theta - \hat{\Theta})\right) + \alpha \cdot \mathrm{Tr}\left(\hat{\Theta}^T \left(I - D^{-1}W\right)\hat{\Theta}\right)$$

Setting the derivative of Eq. (1) with respect to $\hat{\theta}_k$ ($k = 1, \cdots, K$) to zero gives the result $2\left(\hat{\Theta} - \Theta\right) + 2\alpha \cdot \left(I - D^{-1}W\right)\hat{\Theta} = 0$. Thus, the following holds $\hat{\Theta}^* = (1-\kappa)\left(I - \kappa D^{-1}W\right)^{-1}\Theta$ where $\kappa = \frac{\alpha}{1+\alpha}$.

For the iterative solution, we have

$$\hat{\Theta}^{(m)} = \left(\kappa D^{-1}W\right)^m \hat{\Theta}^{(0)} + \sum_{i=0}^{m-1}\left(\kappa D^{-1}W\right)^i (1-\kappa)\Theta$$

It is easy to show that for the spectral radius of $\kappa D^{-1}W$, we have:

$$\rho(\kappa D^{-1}W) \leq \alpha||\kappa D^{-1}W||_\infty = \kappa = \frac{\alpha}{1+\alpha} < 1$$

Therefore, when $m \to \infty$, $\left(\kappa D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right)^m$ converges, and.

$$\lim_{m\to\infty}\left(\kappa D^{-1}W\right)^m = \mathbf{0} \qquad \lim_{m\to\infty}\sum_{i=0}^{m-1}\left(\kappa D^{-1}W\right)^i = \left(I - \kappa D^{-1}W\right)^{-1}$$

which completes the proof. □

A.3 Proof of Theorem 4.3

The objective function of **FEDORA** can be rewritten as follows.

$$\begin{aligned}\mathcal{J}(\Theta, \hat{\Theta}) &= \sum_{k=1}^{K}\left(\frac{1}{\lambda_k}F_k(\theta_k) + \left\|\theta_k - \hat{\theta}_k\right\|_2^2\right) + \frac{\alpha}{2}G\left(\{\hat{\theta}_k\}_{k=1}^K\right) \\ &= \left(\sum_{k=1}^{K}\frac{1}{\lambda_k}F_k(\theta_k)\right) + ||\Theta - \hat{\Theta}||^2 + \frac{\alpha}{2}G(\hat{\Theta})\end{aligned}$$

where $F_k(\theta_k) = \mathbb{E}_{(x^k, y^k)\sim\mathbb{P}_k}[L(x^k, y^k; \theta_k)]$ and $G(\hat{\Theta}) = G\left(\{\hat{\theta}_k\}_{k=1}^K\right) = \sum_{k=1}^K \sum_{k'=1}^K \frac{w_{kk'}}{D_{kk}}\left\|\hat{\theta}_k - \hat{\theta}_{k'}\right\|_2^2$.

We aim to prove that **FEDORA** guarantees the convergence of the above objective. With $\Theta^*, \hat{\Theta}^* = \arg\min_{\Theta^*, \hat{\Theta}^*} \mathcal{J}(\Theta, \hat{\Theta})$, we show that $||\Theta(r) - \Theta^*||$ and $||\hat{\Theta}(r) - \hat{\Theta}^*||$ linearly converge with bounded error. We first explore some properties of the objective function.

Lemma A.1 (Convexity of G). *$G(\{\hat{\theta}_k\}_{k=1}^K)$ is convex w.r.t. $\{\hat{\theta}_k\}_{k=1}^K$.*

Proof. $||x||_2^2 = x^\top x$ is a convex function w.r.t. x. Therefore, $||\hat{\theta}_k - \hat{\theta}_{k'}||_2^2$ is convex w.r.t. $(\hat{\theta}_k - \hat{\theta}_{k'})$. Composition with an affine mapping preserves convexity. Since $(\hat{\theta}_k - \hat{\theta}_{k'})$ is an affine composition of $\{\hat{\theta}_k, \hat{\theta}_{k'}\}$, $||\hat{\theta}_k - \hat{\theta}_{k'}||_2^2$ is convex w.r.t. $\{\hat{\theta}_k, \hat{\theta}_{k'}\}$. A non-negative weighted sum preserves convexity. Since each $\frac{w_{kk'}}{D_{kk}}$ is non-negative, $G(\{\hat{\theta}_k\}_{k=1}^K)$ is convex w.r.t. $\{\hat{\theta}_k\}_{k=1}^K$. □

We study how the estimations of Θ and $\hat{\Theta}$ improve in client update and server update in each round. Denote $\lambda = \max_{k\in\{1,\cdots,K\}}$.

Lemma A.2 (Contraction). *Given a μ-strongly convex function $F(y)$, we define $y_1 = \arg\min_y[F(y) + \lambda||x_1 - y||_2^2]$, $y_2 = \arg\min_y[F(y) + \lambda||x_2 - y||_2^2]$, we have $||y_1 - y_2|| \leq \frac{2\lambda}{\mu+2\lambda}||x_1 - x_2||$.*

Proof. W.l.o.g, $y_1 \neq y_2$. By definition of y_1, y_2, we have

$$\nabla F(y_1) + 2\lambda(y_1 - x_1) = 0 \quad \text{and} \quad \nabla F(y_2) + 2\lambda(y_2 - x_2) = 0$$

$$\Rightarrow \quad \nabla F(y_1) - \nabla F(y_2) = 2\lambda(x_1 - x_2) - 2\lambda(y_1 - y_2)$$

Since F is μ-strongly convex, we have

$$F(y_1) \geq F(y_2) + \nabla F(y_2)^\top (y_1 - y_2) + \frac{\mu}{2}||y_1 - y_2||_2^2$$

$$F(y_2) \geq F(y_1) + \nabla F(y_1)^\top (y_2 - y_1) + \frac{\mu}{2}||y_2 - y_1||_2^2$$

Add them together

$$0 \geq -[\nabla F(y_1) - \nabla F(y_2)]^\top (y_1 - y_2) + \mu||y_1 - y_2||_2^2$$

$$2\lambda(x_1 - x_2)^\top (y_1 - y_2) \geq (2\lambda + \mu)||y_1 - y_2||_2^2$$

$$\frac{2\lambda}{2\lambda + \mu}||x_1 - x_2|| \geq ||y_1 - y_2||$$

which completes the proof. □

Proposition A.3 (Improvement of client update). *In each round of client update, it holds that $||\Theta(r) - \Theta^*||_F \leq \frac{2\lambda}{\mu+2\lambda}||\hat{\Theta}(r) - \hat{\Theta}^*||_F + \sqrt{K}\delta$.*

Proof. Notice that $\theta_k^* = \arg\min_{\theta_k} F_k(\theta_k) + \lambda_k\left\|\theta_k - \hat{\theta}_k^*\right\|_2^2$, $\theta_k^*(r) = \arg\min_{\theta_k} F_k(\theta_k) + \lambda_k\left\|\theta_k - \hat{\theta}_k(r)\right\|_2^2$. By Lemma A.2, for all $k \in \{1, \cdots, K\}$ we have

$$||\theta_k^*(r) - \theta_k^*||_2 \leq \frac{2\lambda_k}{\mu + 2\lambda_k}||\hat{\theta}_k(r) - \hat{\theta}_k^*||_2 = \frac{2\lambda}{\mu + 2\lambda}||\hat{\theta}_k(r) - \hat{\theta}_k^*||_2$$

In matrix form, it holds that $||\Theta^*(r) - \Theta^*||_F \leq \frac{2\lambda}{\mu+2\lambda}||\hat{\Theta}(r) - \hat{\Theta}^*||_F$. Then, we have

$$||\Theta(r) - \Theta^*(r)||_F = \sqrt{\sum_{k=1}^{K}||\theta_k(r) - \theta_k^*(r)||_2^2} \leq \sqrt{K\delta^2} = \sqrt{K}\delta$$

which completes the proof. □

Proposition A.4 (Improvement of server update). *In each round of server update, it holds that $||\hat{\Theta}(r+1) - \hat{\Theta}^*||_F \leq ||\Theta(r) - \Theta^*||_F$.*

Proof. Notice that $\hat{\Theta}^* = \arg\min_{\hat{\Theta}} ||\Theta^* - \hat{\Theta}||_F^2 + \frac{\alpha}{2}G(\hat{\Theta})$, $\hat{\Theta}(r+1) = \arg\min_{\hat{\Theta}} ||\Theta(r) - \hat{\Theta}||_F^2 + \frac{\alpha}{2}G(\hat{\Theta})$. Since G is convex, by Lemma A.2 with $\mu = 0$, we have $||\hat{\Theta}(r+1) - \hat{\Theta}^*||_F \leq ||\Theta(r) - \Theta^*||_F$. □

Finally, the convergence analysis of **FEDORA** is given below.

Proof.

$$\begin{aligned}\|\Theta(R)-\Theta^*\|_F &\le \frac{2\lambda}{\mu+2\lambda}\|\hat{\Theta}(R)-\hat{\Theta}^*\|_F+\sqrt{K}\delta\\ &\le \left(\frac{2\lambda}{\mu+2\lambda}\right)^R\|\Theta(0)-\Theta^*\|_F+\sum_{r=0}^{R-1}\left(\frac{2\lambda}{\mu+2\lambda}\right)^r\sqrt{K}\delta\\ &\le \left(\frac{2\lambda}{\mu+2\lambda}\right)^R\|\Theta(0)-\Theta^*\|_F+\frac{1}{1-\frac{2\lambda}{\mu+2\lambda}}\sqrt{K}\delta\\ &= \left(\frac{2\lambda}{\mu+2\lambda}\right)^R\|\Theta(0)-\Theta^*\|_F+\frac{\mu+2\lambda}{\mu}\sqrt{K}\delta\end{aligned}$$

which completes the proof. □

A.4 Proof of Theorem 4.4

Proof. Using assumption 3, the following holds

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]-\mathbb{E}_{\mathbb{P}_{k'}}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]\\ &\le \mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]-\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]+\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]-\mathbb{E}_{\mathbb{P}_{k'}}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]\\ &\le \nu\left\|\theta_k^*-\bar{\theta}_{k'}\right\|_2^2+d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\end{aligned}$$

We let $\alpha_{kk'}=w_{kk'}/D_{kk}$, then

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\le\sum_{k'=1}^K\alpha_{kk'}\left(\mathbb{E}_{\mathbb{P}_{k'}}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]+\nu\left\|\theta_k^*-\bar{\theta}_{k'}\right\|_2^2+d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\sum_{k'=1}^K\alpha_{kk'}\left(\mathbb{E}_{\mathbb{P}_{k'}}\left[L\left(x,y;\bar{\theta}_{k'}\right)\right]+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\sum_{k'=1}^K\alpha_{kk'}\left(\mathbb{E}_{\mathbb{P}_{k'}}\left[L\left(x,y;\tilde{\theta}\right)\right]+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\sum_{k'=1}^K\alpha_{kk'}\left(\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\tilde{\theta}\right)\right]+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\sum_{k'=1}^K\alpha_{kk'}\left(\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\quad+M\sqrt{\frac{\log 4/\delta}{2n_k}}\\ &=\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+M\sqrt{\frac{\log 4/\delta}{2n_k}}-\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\theta_k^*\right)\\ &\quad+\sum_{k'=1}^K\alpha_{kk'}\left(\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\theta_k^*\right)+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+2M\sqrt{\frac{\log 4/\delta}{2n_k}}-\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\\ &\quad+\sum_{k'=1}^K\alpha_{kk'}\left(\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\theta_k^*\right)+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\le\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+\sum_{k'=1}^K\alpha_{kk'}\left(\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\bar{\theta}_k\right)+\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\quad+2M\sqrt{\frac{\log 4/\delta}{2n_k}}-\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\\ &\le\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]\\ &\quad+\sum_{k'=1}^K\alpha_{kk'}\left(\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)+3M\sqrt{\frac{\log 4/\delta}{2n_k}}-\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\\ &\le\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)+\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]\\ &\quad+\sum_{k'=1}^K\alpha_{kk'}\left(\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)+4M\sqrt{\frac{\log 4/\delta}{2n_k}}-\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\end{aligned}$$

where

$$\begin{aligned}&\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\theta_k^*\right)+\nu\left\|\theta_k^*-\bar{\theta}_k\right\|_2^2+\nu\sum_{k'=1}^K\alpha_{kk'}\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2\\ &\le\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\bar{\theta}_k\right)+\nu\sum_{k'=1}^K\alpha_{kk'}\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2\\ &\le\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]+\nu\sum_{k'=1}^K\alpha_{kk'}\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2\end{aligned}$$

and following [2], we have

$$\begin{aligned}\frac{1}{n_k}\sum_{i=1}^{n_k}L\left(x_i^k,y_i^k;\tilde{\theta}\right)&\le\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]+4\sqrt{\left(\sum_{k'=1}^K\frac{\alpha_{kk'}^2\left(2d\log 2(n+1)+\log\frac{4}{\delta}\right)}{n_{k'}}\right)}\\ &\quad+\sum_{k'=1}^K\alpha_{kk'}d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)+M\sqrt{\frac{\log 4/\delta}{2n_k}}\end{aligned}$$

As a result, the following holds

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\theta_k^*\right)\right]\\ &\le\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]+\frac{1}{2}\sum_{k'=1}^K\alpha_{kk'}\left(\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+2d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)+2M\sqrt{\frac{\log 4/\delta}{2n_k}}\\ &\quad+2\sqrt{\left(\sum_{k'=1}^K\frac{\alpha_{kk'}^2\left(2d\log 2(n+1)+\log\frac{4}{\delta}\right)}{n_{k'}}\right)}+\frac{1}{2}\sum_{k'=1}^K\alpha_{kk'}d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)+\frac{1}{2}M\sqrt{\frac{\log 4/\delta}{2n_k}}\\ &=\mathbb{E}_{\mathbb{P}_k}\left[L\left(x,y;\bar{\theta}_k\right)\right]+\frac{1}{2}\sum_{k'=1}^K\alpha_{kk'}\left(\nu\left\|\bar{\theta}_k-\bar{\theta}_{k'}\right\|_2^2+3d_{\mathcal{Y}}\left(\mathbb{P}_k,\mathbb{P}_{k'}\right)\right)\\ &\quad+2\sqrt{\left(\sum_{k'=1}^K\frac{\alpha_{kk'}^2\left(2d\log 2(n+1)+\log\frac{4}{\delta}\right)}{n_{k'}}\right)}+\frac{5}{2}M\sqrt{\frac{\log 4/\delta}{2n_k}}\end{aligned}$$

which completes the proof. □

B ADDITIONAL EXPERIMENTAL SETUP

In this paper, we consider the following client generation and train/validation/test split methods. Take Rotated MNIST as an example, we generate balanced and imbalanced clients respectively.

- For the balanced setting, we randomly choose 128 samples from MNIST without replacement to formulate the training set for each client, and 64 samples as the validation set. The test data of the original MNIST are partitioned into K clients each receiving $10000/K$ samples (K is the number of clients), in order to formulate the test set of each client.
- In contrast, for the imbalanced setting, each of the $K-1$ clients also has 128 training samples and 64 validation samples, while the remaining client has $(60000-(K-1)\times 192)*2/3$ training samples and $(60000-(K-1)\times 192)/3$ validation samples. In this case, the remaining client will have a significantly large number of training and validation samples. This helps us evaluate the impact of the number of training/validation samples on the personalized federated learning approaches.

After partitioning MNIST to local clients, we then rotate the training/validation/test images in each client according to a specific angle $a=360*i/K$ ($i\in\{0,1,\cdots,K-1\}$ is the client index). For other real-world data sets (e.g., Yearbook and agriculture), the clients are naturally partitioned based on the data collection time (as described in Subsection 5.1.1). Then we simply set the train/validation/test ratio as 0.4/0.2/0.4.