ADAPT TO ADAPTATION: LEARNING TO PERSONALIZE FOR CROSS-SILO FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The goal of conventional federated learning (FL) is to train a global model for a federation of clients with decentralized data, reducing the systemic privacy risk of centralized training. The distribution shift across non-IID datasets, also known as the data heterogeneity, often poses a challenge for this one-global-model-fits-all solution. In this work, we propose `APPLE`, a personalized cross-silo FL framework that adaptively learns how much each client can benefit from other clients' models. We also introduce a method to flexibly control the focus of training `APPLE` between global and local objectives. We empirically evaluate our method's convergence and generalization behavior and performed extensive experiments on two benchmark datasets and two medical imaging datasets under two non-IID settings. The results show that the proposed personalized FL framework, `APPLE`, achieves state-of-the-art performance compared to several other personalized FL approaches in the literature.

1 INTRODUCTION

In recent years, federated learning (FL) (McMahan et al., 2017; Kairouz et al., 2019) has shown great potential in training a shared global model for decentralized data. In contrast with previous large-scale machine learning approaches, training in FL resides on the sites of the data owners without the need to migrate the data, which reduces systemic privacy risks and expenses on massive datacenters (Kairouz et al., 2019). Compared to separate individual training, the leading FL algorithm, `FedAvg` (McMahan et al., 2017), as a representative of global FL algorithms, attempts to train a consensus global model by iteratively averaging the local updates of the global model. However, such an approach often suffers from the convergence challenges (Zhao et al., 2018; Hsieh et al., 2020) brought by the statistical data heterogeneity (Smith et al., 2017; Li et al., 2018), where data are not identically distributed (non-IID) across all clients due to the inherent diversity (Li et al., 2019; Sahu et al., 2018).

Data heterogeneity lies almost everywhere in real-world FL applications. For cross-device training of a mobile keyboard next-word prediction model, non-IIDness is generated by different typing preferences of users (Hard et al., 2018; Yang et al., 2018); medical datasets across different silos are heterogeneous by nature, due to factors such as different data acquisition protocols and various local demographics (Rieke et al., 2020). Data heterogeneity may lead to inferior performance of federated models in certain silos (medical institutes) and that may lose their incentives to participate in the federation.

Attempts to supplement FL algorithms with the ability to better handle data heterogeneity fall into two general schemes, based on the number of the trained model(s). The first scheme tries to enhance the global consensus model for higher robustness to non-IID datasets (Li et al., 2018; Karimireddy et al., 2020; Acar et al., 2020). The other scheme looks at FL from a client-centric perspective, aiming to train multiple models for different clients (Kairouz et al., 2019), and is often referred to as *personalized* FL.

Personalized FL tries to systematically mitigate the influence of data heterogeneity, since a different model could be trained for a different target data distribution(Kulkarni et al., 2020; Kairouz et al., 2019). Efforts in this direction include approaches that fine-tune the global model (Wang et al., 2019a), and more sophisticated approaches that leverage meta-learning (Finn et al., 2017; Nichol

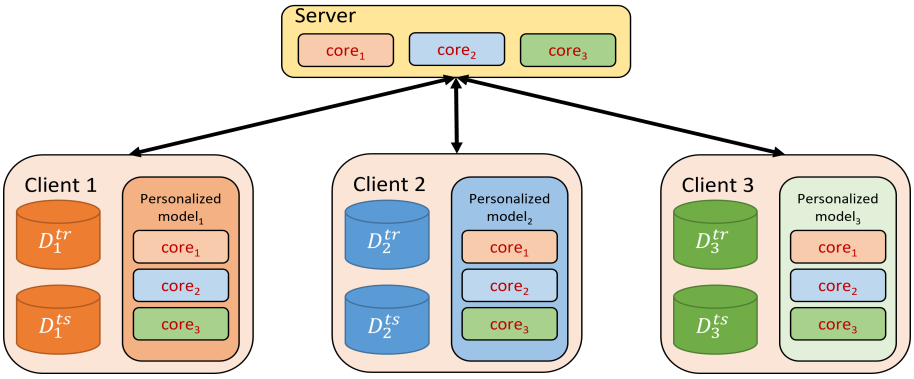

Figure 1: The workflow of the proposed framework, APPLE. Each client trains a local personalized model, uploads its updated core model to the server, and downloads others' core models from the server as needed at the beginning of each round.

et al., 2018) or multi-task learning (Smith et al., 2017; Sattler et al., 2020) to learn the relationships between source and target domains/tasks, which corresponds to different distributions of the datasets. Other efforts pay more attention to interpolations between local models and the global model (Deng et al., 2020; Huang et al., 2021; Zhang et al., 2020).

In this work, we focus on the personalization aspect of cross-silo FL for non-IID data. We propose ***A**daptive **P**ersonalized Cross-Sil**o** **Fe**derated Learning* (APPLE), a novel personalized FL framework for cross-silo settings that adaptively learns to personalize each client's model by learning how much the client can benefit from other clients' models according to the local objective. In this process, the clients do not need to acquire information regarding other clients' data distributions. We illustrate the workflow of APPLE in Figure 1.

There are three major distinctions between APPLE and other existing personalized FL algorithms: in APPLE, (1) after local training, a client does not upload the local personalized model, but a constructing component of the personalized model, here called a *core model*; (2) the central server only maintains the core models uploaded from the clients, for further downloading purposes; (3) a unique set of local weights on each client, here called a *directed relationship (DR)* vector, is adaptively learned to weight the downloaded core models from the central server. This enables the personalized models to take more advantage of the beneficial core models, while suppressing the less beneficial or potentially harmful core models. We also introduce a method to flexibly control the focus of training APPLE between global and local objectives, by dynamically penalizing the DR vectors.

We summarize our contribution as follows:

- We propose APPLE, a novel personalized cross-silo FL framework that adaptively learns to personalize the client models.
- Within APPLE, we introduce a method to flexibly control the focus of training between global and local objectives via a dynamic penalty.
- We evaluate APPLE on two benchmark datasets and two medical imaging datasets under two types of non-IID settings. Our results show that APPLE achieves state-of-the-art performance in both settings compared to other personalized FL approaches in the literature.

2 Related Work

2.1 Federated Learning on Non-IID Data

Federated learning (FL) (McMahan et al., 2017; Kairouz et al., 2019; Wang et al., 2021) enables participating clients to collaboratively train a model without migrating the clients data, which mitigates the systemic privacy risks. The most notable FL algorithm, FedAvg (McMahan et al., 2017),

achieves this by aggregating the updated copies of the global model using an averaging approach. While concerns for the behavior of FedAvg on non-IID data began to accumulate (Li et al., 2019; Sahu et al., 2018; Zhao et al., 2018), numerous work has been focusing on the robustness of FL on non-IID data. Li et al. (2018) proposed FedProx that penalizes the local update when it is far from the prox-center. Karimireddy et al. (2020) proposed SCAFFOLD that corrects the local gradient under client-drift with control variates. FedDyn by Acar et al. (2020) dynamically updates the regularizer in the empirical risk to reduce the impact of data heterogeneity.

2.2 Personalized Federated Learning

To systematically mitigate the impact of data heterogeneity, a new branch of FL, the *personalized* FL, has emerged in recent years (Kairouz et al., 2019; Kulkarni et al., 2020). Instead of being restricted by the global consensus model, personalized FL allows different models for different clients, especially when the data are drawn from distinct distributions.

In this line of work, a natural way is to fine-tune the global model at each client (Wang et al., 2019a). However, Jiang et al. (2019) claims that fine-tuning the global model may result in poor generalization to unseen data. They also demonstrate the similarity between personalizing a FedAvg-trained model and a type of model-agnostic meta-learning (MAML) (Finn et al., 2017) algorithm called Reptile (Nichol et al., 2018). And many works have been focusing on the overlap between meta-learning and FL Fallah et al. (2020); Khodak et al. (2019); Chen et al. (2018).

Apart from approaches that need further fine-tuning the trained models, Smith et al. (2017) proposed MOCHA that leverages multi-task learning (Zhang & Yang, 2017) to learn the relationship between different clients. Hanzely & Richtárik (2020) seek a balance between the trade-off of global and local models. Sattler et al. (2020) and Ghosh et al. (2020) focus on a new setting where clients are adaptively partitioned into clusters, and a personalized model is trained for clients in the same cluster.

Other personalized FL algorithms include carefully interpolating a model for each client. APFL (Deng et al., 2020) weights the global and the local model at each client. FedFomo (Zhang et al., 2020) computes estimates of the optimal weights for each client's personalized model using a local validation set. FedAMP (Huang et al., 2021) uses an attention-inducing function to compute an interpolated model as the prox-center for the personalized model.

3 Adaptive Personalized Cross-Silo Federated Learning

In this section, we look at personalized FL with more details, and present APPLE, a framework for personalized cross-silo FL that adaptively learns to personalize the client models. Similar to most FL methods, in APPLE, the training progresses in round. Each client iteratively downloads and uploads model parameters in each round. However, in APPLE, each client uploads a constructing component of the **personalized model**, here called a **core model**. And the central server maintains the core models uploaded from the clients. Before we go into further details, we formulate the problem and define the notations which will be used throughout the paper.

3.1 Problem Formulation

In general, federated learning aims to improve model performance of individually trained models, by collaboratively training a model over a number of participating clients, without migrating the data due to privacy concerns. Specifically, the goal is to minimize:

$$\min_{\boldsymbol{w}} f_G(\boldsymbol{w}) = \min_{\boldsymbol{w}} \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}), \tag{1}$$

where $f_G(\cdot)$ denotes the global objective. It is computed as the weighted sum of the N local objectives, with N being the number of clients and $p_i \geq 0$ being the weights. The local objective $F_i(\cdot)$ of client i is often defined as the expected error over all data under local distribution $\mathcal{D}_i$, i.e.

$$F_i(\cdot) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[\mathcal{L}(\cdot; \xi)] \approx \frac{1}{n_i} \sum_{\xi \in D_i^{tr}} \mathcal{L}(\cdot; \xi), \tag{2}$$

where ξ represents the data under local distribution $\mathcal{D}_i$. As shown in Equation 2, $F_i(\cdot)$ is often approximated by the local empirical risk on client i using its training set D_i^{tr} ($n_i = |D_i^{tr}|$). The notable `FedAvg` tries to solve the empirical risk minimization (EMR) by iteratively averaging the local updates of the copy of global model, i.e. $\boldsymbol{w_{t+1}} = \sum_{i=1}^{K} p_i \boldsymbol{w_{t+1}^i}$, where K is the number of selected clients in each round, and the p_i's are defined as the ratio between the number of data samples on client i, n_i, and the number of total data samples from all clients n, with constraints: $\forall i \in [N],\ p_i \geq 0$, and $\sum_i p_i = 1$.

In personalized FL, the global objective slightly changes to a more flexible form:

$$\min_{\boldsymbol{W}} f_P(\boldsymbol{W}) = \min_{\boldsymbol{w_i}, i\in[N]} f_P(\boldsymbol{w_1}, ..., \boldsymbol{w_N}) = \min_{\boldsymbol{w_i}, i\in[N]} \sum_{i=1}^{N} p_i F_i(\boldsymbol{w_i}) \tag{3}$$

where $f_P(\cdot)$ is the global objective for the personalized algorithms, and $\boldsymbol{W} = [\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_N}]$ is the matrix with all personalized models. In this work, we aim to obtain the optimal $\boldsymbol{W^*} = \arg\min_{\boldsymbol{W}} f_P(\boldsymbol{W})$, which equivalently represents the optimal set of personalized models $\boldsymbol{w_i^*}, i \in [N]$. In addition, we focus on the cross-silo setting of FL, which is differentiated with the cross-device setting by much smaller number of participating (stateful) clients, and no selection of clients are strictly needed at the beginning of each round.

3.2 Adaptively Learning to Personalize

As mentioned above, in `APPLE`, each client uploads to the central server a **core model**, and downloads other clients' core models maintained on the server at the end and beginning of each round, respectively. In an ideal scenario where communication cost is not taken into account, each core model maintained on the server is downloaded by every client. In practice, communication is costly, and the limitations on the communication bandwidth always exists. We will discuss how `APPLE` handles this in Section 3.4.

In `APPLE`, after each client has downloaded the needed core models from the server, the personalized model for client i is subsequently computed as

$$w_i^{(p)} = \sum_{j=1}^{N} p_{i,j} w_j^{(c)}, \tag{4}$$

where $\boldsymbol{w_i^{(p)}}$ represents the personalized model of client i, and $\boldsymbol{w_j^{(c)}}$ is the downloaded client j's core model. Similar to some personalized FL algorithms that focus on interpolating a model for each client (Acar et al., 2020; Zhang et al., 2020; Huang et al., 2021), the personalized model here is also a convex combination of models. The difference is that in `APPLE`, there is a unique set of learnable weights for each client. We use $p_{i,j}$ to denote the learnable weight on client i for the downloaded core model $\boldsymbol{w_j^{(c)}}$, and use $\boldsymbol{p_i} = [p_{i,1}, ..., p_{i,N}]^T$ to denote the set of learnable weights on client i, calling it the **directed relationship (DR)** vector.

During local training on client i, after the personalized model $\boldsymbol{w_i^{(p)}}$ is computed, we freeze the downloaded core models ($\boldsymbol{w_j^{(c)}},\ j \neq i$), and only update its local core model $\boldsymbol{w_i^{(c)}}$ using a gradient-based method, such as local Stochastic Gradient Descent (SGD). Meanwhile, we adaptively update the DR vector, $\boldsymbol{p_i}$, according to the local objective, i.e.

$$\boldsymbol{w_i^{(c)}} \leftarrow \boldsymbol{w_i^{(c)}} - \eta_1 \frac{\partial}{\partial \boldsymbol{w_i^{(c)}}} F_i(\boldsymbol{w_i^{(p)}}) \tag{5}$$

$$\boldsymbol{p_i} \leftarrow \boldsymbol{p_i} - \eta_2 \frac{\partial}{\partial \boldsymbol{p_i}} F_i(\boldsymbol{w_i^{(p)}}) \tag{6}$$

Note that after a round of local training is finished, each client only uploads the local core model ($\boldsymbol{w_i^{(c)}}$ for client i, $i \in [N]$) to the server. The DR vector $\boldsymbol{p_i}$ is always maintained at client i without any migration, which makes it impossible for others to infer the personalized model, and further protects the data privacy.

Algorithm 1 APPLE

Input: N clients, learning rates η_1, η_2, number of total rounds R, proximal term coefficients $\lambda(r), \mu$, prox-center $\boldsymbol{p_0}$

$\forall i \in [N]$, initialize core models $\boldsymbol{w_i^{(c)}}$ on server
$\forall i \in [N]$ **in parallel**, initialize local DR vector $\boldsymbol{p_i}$
for $r \leftarrow 1, 2, ..., R$ **do**
  for $i \leftarrow 1, 2, ..., N$ **in parallel do**
    Download core models from server as needed.
    Iteratively optimize the local core model $\boldsymbol{w_i^{(c)}}$ and local DR vector $\boldsymbol{p_i}$ by the following:
      Compute personalized model $\boldsymbol{w_i^{(p)}}$ with $\boldsymbol{w_j^{(c)}}, j \in [N]$ and $\boldsymbol{p_i}$ by Eq. (4)
      Compute local empirical risk $F_i(\boldsymbol{w_i^{(p)}})$ by Eq. (7)
      Update $\boldsymbol{w_i^{(c)}}$ and DR vector $\boldsymbol{p_i}$ by Eq. (5, 6)
    When optimization is finished, upload local core model $\boldsymbol{w_i^{(c)}}$ to the server
  end for
end for
return Personalized models $\boldsymbol{w_1^{(p)}}, \boldsymbol{w_2^{(p)}}, ..., \boldsymbol{w_N^{(p)}}$ on site of corresponding client.

3.3 PROXIMAL DIRECTED RELATIONSHIPS

In APPLE, for each client, the learned global information is blended in the downloaded core models, whose contributions to the local personalized model are measured by the learnable weights in the DR vector. Ideally, the entry $p_{i,i}$, or "self-relationship", should be larger than the other entries in $\boldsymbol{p_i}$, since the local core model $\boldsymbol{w_i^{(c)}}$ is the only network trained with local distribution $\mathcal{D}_i$. On the other hand, for all $j \neq i$, $p_{i,j}$ should be somewhere in between 0 and $p_{i,i}$, if the local personalized model $\boldsymbol{w_i^{(p)}}$ can benefit more from $\boldsymbol{w_j^{(c)}}$ (may happen if the distributions $\mathcal{D}_i$ and $\mathcal{D}_j$ are similar), while $p_{i,j}$ should be closer to 0 or even negative, if $\boldsymbol{w_j^{(c)}}$ results in potential negative transfer to $\boldsymbol{w_i^{(p)}}$.

However, in a real-world situation, due to the data heterogeneity in FL, chances for similar distributions among clients are slim. Most off-diagonal entries in the DR matrix should be small. Without any constraint, this may result in a natural pitfall that the learned DR matrix is too quickly drawn to somewhere near the identity matrix (in terms of the Frobenius norm). This can lead the personalized models to hardly benefit from FL, and the training process undesirably resembles individual learning.

To address this issue and facilitate collaboration between clients, we penalize the directed relationship by adding a proximal term (Rockafellar, 1976; Li et al., 2018) to the local empirical risk. We summarize the final empirical risk in APPLE in Equation 7.

$$F_i(\boldsymbol{w_i^{(p)}}) = \frac{1}{n_i} \sum_{\xi \in D_i^{tr}} \mathcal{L}(\boldsymbol{w_i^{(p)}}; \xi) + \lambda(r) \frac{\mu}{2} ||\boldsymbol{p_i} - \boldsymbol{p_0}||_2^2 \tag{7}$$

In Equation 7, λ and μ are two coefficients for the proximal term with the prox-center at $\boldsymbol{p_0} = [n_1/n, ..., n_N/n]$. It is obvious that FedAvg is a special case of APPLE by setting μ to ∞, which infers that a larger coefficient of the proximal term can push the personalized model to a global model, facilitating collaboration between clients. While this may benefit the personalized model to learn high-level features, it is not always desired throughout the training. Ultimately, with the learned high-level features, the personalized models should still focus on how to be personalized.

To this end, we design λ with a certain type of decay in terms of the current training round r, inspired by Wang et al. (2019b), and call such $\lambda(r)$ a *loss scheduler*. More details regarding the loss scheduler are presented in Appendix A.1. We summarize the steps of APPLE in Algorithm 1.

3.4 APPLE UNDER LIMITED COMMUNICATION BUDGET

The collaboration between clients in APPLE relies on the downloaded core models at the beginning of each round. Without considering communication limitations, each client can download all other clients' core models from the server, which effectively enhances the communication across all clients.

However, although the number of participating clients in cross-silo settings will not be as large compared to the cross-device settings, the communication cost of downloading all core models to each client is still considerable. While this issue can be mitigated by techniques including quantization (Xu et al., 2018; Reisizadeh et al., 2020; Dai et al., 2019) and knowledge distillation (Chen et al., 2017; Hinton et al., 2015; Li & Wang, 2019), in the worst-case scenario, the communication per round still cost N times more overhead than algorithms that only download one model for each client per round (e.g. FedAvg).

To address this issue, we restrict the number of models a client can download per round, denoted by M. Under limited communication budget ($M < N - 1$), briefly, APPLE decides to select which M core models to download by the following rules: on client i, the core model of client j will be downloaded if it has never been downloaded on client i (breaks tie randomly); if all other clients' core models have all been downloaded at least once, with high probability, priority goes to client j's core model who has a large $p_{i,j}$. We elaborate this selection process in Appendix A.2.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness of APPLE with experiments under two different non-IID federated settings. We show the empirical convergence behaviors and the generalization performances with respect to each client on different image datasets. In addition, we study the transition of the pairwise directed relationships between different clients throughout the training process. Last but not least, we also investigate the performance of APPLE under different levels of limited communication budget.

To evaluate the convergence behavior, we plot the training losses and test accuracies against the number of trained rounds. For the personalized methods, the training loss and test accuracy are computed in a way such that if a data sample resides on client i, then we use the personalized model of client i to conduct inference with it. In addition, we quantify the performance of the methods by computing the test accuracies with respect to each client, and the *best mean client test accuracy (BMCTA)* (best over all rounds, mean over all clients), a metric also used in Huang et al. (2021).

4.1 EXPERIMENTAL SETUP

Datasets We use four public datasets including two benchmark image datasets: MNIST (LeCun & Cortes, 2010) and CIFAR10 (Krizhevsky et al., 2009), and two medical imaging datasets from the MedMNIST datasets collection (Yang et al., 2021), namely the OrganMNIST(axial) dataset: an 11-class of liver tumor image dataset (Bilic et al., 2019), and the PathMNIST dataset: a 9-class colorectal cancer image dataset (Kather et al., 2019). We partition each of the four datasets into a training set and test set with the same distribution (if such split does not pre-exist). Then we transform the datasets according to a non-IID distribution, and ensure the same distribution of training and test set on the same client.

Pathological and Practical Non-IID Settings We design two non-IID distributions for empirical evaluation, namely the *pathological non-IID* and the *practical non-IID*. For the pathological non-IID, we follow precedent work and select two random classes for each client. A random percentage of images from each of the two selected classes is assigned to the client. To simulate a cross-silo setting, the number of clients is set to be 12. For the practical non-IID, our endeavor aims to simulate a more realistic cross-silo federation of medical institutes. To this end, we partition each class of the dataset into 12 shards (corresponding to the 12 clients): 10 shards of 1%, 1 shard of 10% and 1 shard of 80% images within this class. A randomly picked shard from each class is assigned to each client, so that every client will possess data from every class. The practical non-IID setting is more similar to the real-world FL in medical applications.This is because the datasets at medical institutes

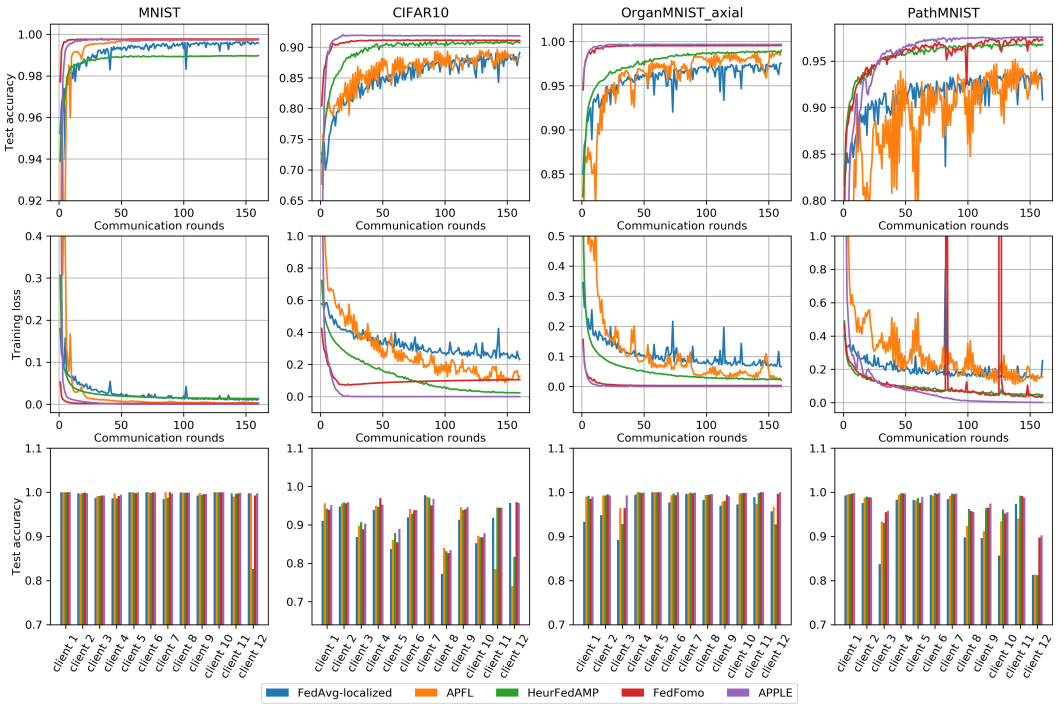

Figure 2: The training loss, test accuracy and client test accuracies of personalized methods under the pathological non-IID setting.

most likely contain a variety of categories of data. And due to different demographic distributions of patients, medical institutes located in different regions may have more frequent occurrences of different categories of data. As a result, these datasets are often imbalanced with different majority classes, and have a wide range in size. Appendix B.1 shows the data distributions in further details.

Compared Baselines We compare `APPLE`, with and without proximal DR penalty, against the following approaches: (1) `Separate` training, meaning the clients' models are trained purely locally without FL; (2) the `FedAvg` (McMahan et al., 2017) which takes the average of the locally trained copies of the global model; (3) `FedAvg-local`, a naïve personalized approach of using the locally trained copy of `FedAvg`'s global model; (4) a fine-tuning approach (Wang et al., 2019a) on the `FedAvg` and on the `FedProx` (Li et al., 2018), here denoted as `FedAvg-FT` and `FedProx-FT`, respectively; (5) `APFL` (Deng et al., 2020), a personalized method using a mixture of the global and the local model; (6) `HeurFedAMP` (Huang et al., 2021), a personalized method on the cross-silo setting with federated attentive message passing; and (7) `FedFomo` (Zhang et al., 2020), a personalized method that computes first-order approximations for the personalized models. We train each method 160 rounds with 5 local epochs and summarize the results as follows.

4.2 Experimental Results

We summarize the empirical convergence behavior and performance under pathological and practical non-IID settings in Figure 2 and Table 1. Convergence performance under the practical non-IID setting is shown in Figure 7 in Appendix B.2. Across all datasets and different non-IID settings, our proposed method has a fast convergence, and achieves highest BMCTAs. Specifically, for the pathological non-IID setting, the separate training reaches comparable performance with all other methods, due to little similarity in data distribution shared by different clients, and the small number of classes in each client. With a direct averaging of the local updates as in `FedAvg` and no fine-tuning as in `FedAvg-FT` and `FedProx-FT`, the global model is hardly able to boost the performance of separate training. The personalized FL methods bring further improvement to the naïve personalization, and `APPLE` outperforms the other compared personalized FL methods. For the practical non-IID setting, since different client can have different majority classes in the local

Table 1: Best mean client test accuracy (BMCTA) of the four datasets under the pathological and practical non-IID settings. Highest performance is represented in bold.

	Pathological non-IID				Practical non-IID			
	MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST	MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST
Separate	97.34	74.96	93.14	87.09	78.20	63.06	65.21	61.36
`FedAvg`	95.71	51.44	59.43	56.61	94.00	34.32	86.56	53.83
`FedAvg-local`	99.52	90.10	96.76	93.21	97.47	71.99	93.75	78.70
`FedAvg-FT`	99.43	90.49	97.03	92.31	97.66	72.08	94.13	78.69
`FedProx-FT`	99.43	90.49	97.03	92.38	97.66	72.08	94.13	78.69
`APFL`	99.75	89.30	98.72	98.98	98.80	71.19	93.75	78.70
`HeurFedAMP`	98.13	91.10	98.39	96.55	97.45	69.54	86.82	79.33
`FedFomo`	99.71	91.96	99.31	97.24	98.05	70.15	82.86	79.39
`APPLE`, $\mu = 0$	99.73	92.22	**99.66**	96.78	**99.00**	75.62	**95.70**	84.22
`APPLE`, $\mu \neq 0$	**99.77**	**92.68**	99.61	**97.51**	98.97	**77.41**	95.62	**86.39**

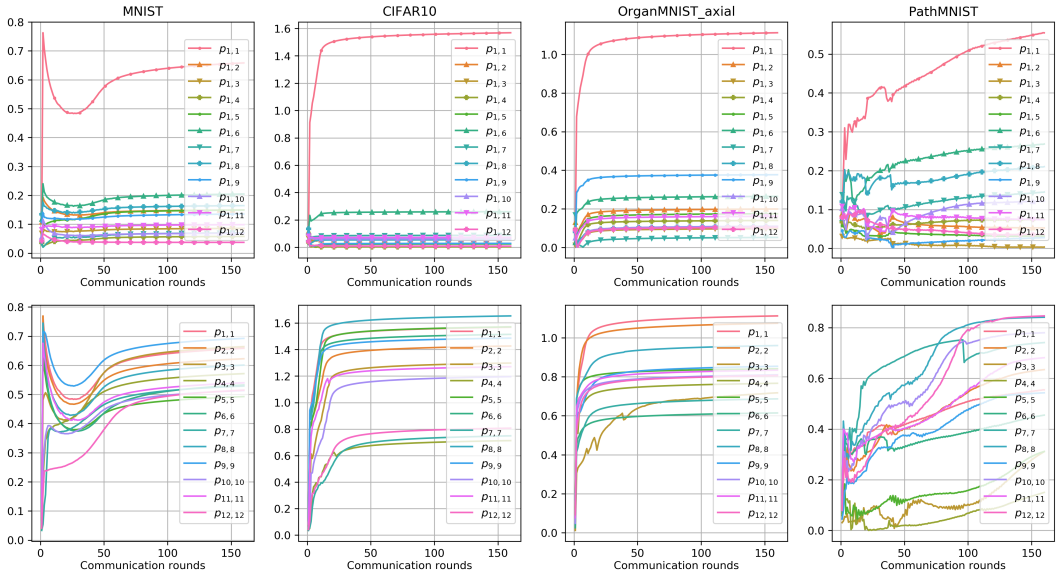

Figure 3: Directed Relationships of different datasets under the pathological non-IID setting. The first row shows the DRs on client 1. The second row shows the "self-relationships", $p_{i,i}$, for each client.

imbalanced dataset, high performance requires careful integration of the global information. As a result, the fine-tuning method outperforms some personalized methods, and `APPLE` also reaches state-of-the-art performance in all settings.

Next, we visualize the trajectories of the directed relationships throughout the training process. Specifically, we study the 12 local DRs on client 1, and the $p_{i,i}$'s for all clients. Figure 3 shows these trajectories of DRs under the pathological non-IID, and Figure 8 in Appendix B.2 shows them under the practical non-IID setting. As mentioned in Section 3.3, the self-relationship, $p_{i,i}$, should be larger than the other local DRs since on client i, the only updated core model is $\boldsymbol{w}_{\boldsymbol{i}}^{(c)}$. And depending on the similarity between the distribution on client i and on client j, larger similarity will push $p_{i,j}$ towards $|p_{i,i}|$ and lower similarity will push $|p_{i,j}|$ towards 0. These properties can be observed in Figure 3. For instance, in CIFAR10, even in the same class, images can have high variance, different client shares little similarity in distribution, so it has high $p_{i,i}$ and low $p_{i,j}$. In PathMNIST, $p_{1,6}$ and $p_{1,8}$ are closer to $p_{1,1}$, which makes sense since client 1 and 6 both have a

Table 2: BMCTA of APPLE ($\mu = 0$) and FedFomo with a maximum of M models to download for each client of the $N = 12$ clients.

		Pathological non-IID				Practical non-IID			
		MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST	MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST
$M = 11$	FedFomo	99.71	91.96	99.31	97.24	98.05	70.15	82.86	79.39
	APPLE	99.73	92.22	99.66	96.78	99.00	75.62	95.70	84.22
$M = 7$	FedFomo	99.71	91.95	99.31	97.33	97.65	70.24	80.88	80.19
	APPLE	99.73	92.17	99.53	97.15	98.70	76.14	94.21	84.07
$M = 5$	FedFomo	99.71	91.94	99.31	97.40	97.47	70.44	82.83	79.62
	APPLE	99.72	92.28	99.48	97.17	98.45	75.63	94.49	85.46
$M = 2$	FedFomo	99.71	91.98	99.31	97.25	96.51	69.87	79.53	79.26
	APPLE	99.70	92.41	99.47	97.11	98.29	74.84	92.29	84.64
$M = 1$	FedFomo	99.71	91.95	99.31	97.15	91.54	69.93	78.37	75.17
	APPLE	99.66	92.31	99.59	96.29	98.52	73.03	93.55	83.35

large portion of images from class 1, and client 1 and 8 both have a large portion of images from class 7 (refer to Figure 5 in Appendix B.1 for the data distribution).

Furthermore, we report the performance of APPLE under limited communication budget. With the same levels of communication restriction, we compare APPLE against FedFomo since FedFomo also needs to download $N - 1$ models to each client by default. We restrict the maximum number of downloaded models for each client per round, M, to be $11, 7, 5, 2, 1$ ($M \geq 11$ is equivalent to no communication limitation since the number of clients is 12). Table 2 shows the results under limited communication budget. For the pathological non-IID setting, results are mixed across different datasets. APPLE outperforms FedFomo on the CIFAR10 and OrganMNIST (axial) dataset, while FedFomo reaches higher performance on the PathMNIST dataset. For the practical non-IID setting, APPLE outperforms FedFomo across all datasets and different levels of limitations. Note that in APPLE, less downloaded models (smaller M) does not necessarily lead to an inferior performance. This is because the proposed rule of picking which models to download tends to download to a client the top M core models that has the highest chance to benefit the client.

5 Conclusions

In this work, we proposed APPLE, a novel personalized cross-silo federated learning framework for non-IID data, that adaptively learns the quantification of how much each client can benefit from other clients' models, named as directed relationships. We introduced a proximal directed relationship penalty on the local objective to control the training between global and local. In addition, we evaluate our method's empirical convergence behavior and performance on four image datasets over two non-IID settings. Our experimental results show the overall superior effects of APPLE several related personalized FL alternatives. Through the visualization of directed relationships, our study empirically shows that: a client can adaptively take more advantage from other clients with similar distribution, while mitigate the potential non-beneficial influence or negative transfer from clients with drastically different distributions. We also investigated the behavior of APPLE under limited communication budget and showed that APPLE can still reach state-of-the-art performance with little drop in performance. As a future work, we plan to further explore personalized FL algorithm that is robust to non-IID data.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, et al. The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056*, 2019.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

Xinyan Dai, Xiao Yan, Kaiwen Zhou, Han Yang, Kelvin KW Ng, James Cheng, and Yu Fan. Hyper-sphere quantization: Communication-efficient sgd for federated learning. *arXiv preprint arXiv:1911.04655*, 2019.

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

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088*, 2020.

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7865–7873, 2021.

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

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

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

Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

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

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

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020.

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

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3:3, 2018.

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

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019a.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5017–5026, 2019b.

Yuhui Xu, Yongzhuang Wang, Aojun Zhou, Weiyao Lin, and Hongkai Xiong. Deep neural network compression with single and multiple level quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195. IEEE, 2021.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

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

A Details of the Algorithm Design

A.1 Loss Scheduler for Proximal Directed Relationships

To push the model to learn more global information at the beginning of the training while gradually transitioning to focusing more on local training, we design the loss scheduler, $\lambda(r)$, to be a monotonically decreasing function between 1 and 0 in terms of the current training round r. Theoretically, $\lambda(r)$ can be designed in different forms, as long as it has the above property. Here, we explore the following two types of loss scheduler (shown in Figure 4) and treat the choice of loss scheduler type as an additional hyperparameter. In both of the following two loss scheduler expressions, L is the round number after which the loss scheduler's value is always 0:

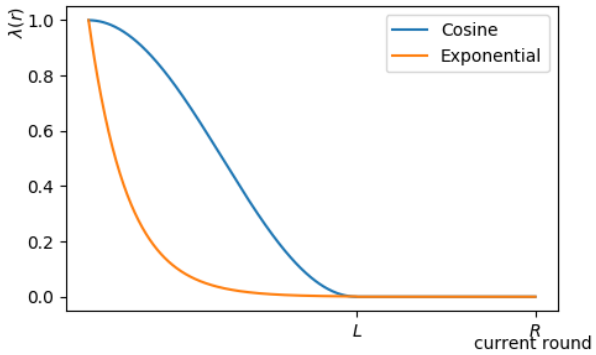

Figure 4: loss-scheduler

- a cosine-shaped scheduler: $\lambda(r) = (\cos(r\pi/L) + 1)/2$ indicating the learning focus transitions gradually from global to local.
- an exponentially decreasing scheduler: $\lambda(r) = \epsilon^{r/L},\ \epsilon = 10^{-3}$, indicating a rapid transition from global to local.

A.2 Core Model Selection under Limited Communication Budget

Under the constraint that a maximum of M core models can be downloaded for each of the N clients per round, we first compute a normalized set of powers as the probabilities for the core models to be selected. The base of the powers is shared among all clients, and is computed by

$$b(r) = \max(1.5, rM/N), \tag{8}$$

where $b(r)$ is the base with respect to the current training round r, and rM/N computes the mean downloaded times per core model for the first r rounds. For client i, the exponent of the powers are $|p_{i,j}|$'s. In other words, the core model $\boldsymbol{w}_j^{(c)}$ will be downloaded to client i with probability:

$$P(\boldsymbol{w}_j^{(c)} \text{downloaded to client } i) = \frac{b(r)^{|p_{i,j}|}}{\sum_{j=1, j\neq i}^{N} b(r)^{|p_{i,j}|}} \tag{9}$$

The reasoning behind this exponential design is that, as training progresses and r increases, $|p_{i,j}|$ gradually represents the contribution of core model $\boldsymbol{w}_j^{(c)}$ on client i with more *confidence*. However, for the first several rounds (with small r) where the core model $\boldsymbol{w}_j^{(c)}$ still has a large potential to update, the confidence of $|p_{i,j}|$ to represent the contribution is still small. With an exponential design, where the base is correlated to the mean downloaded times per core model, this growth in confidence can be better represented.

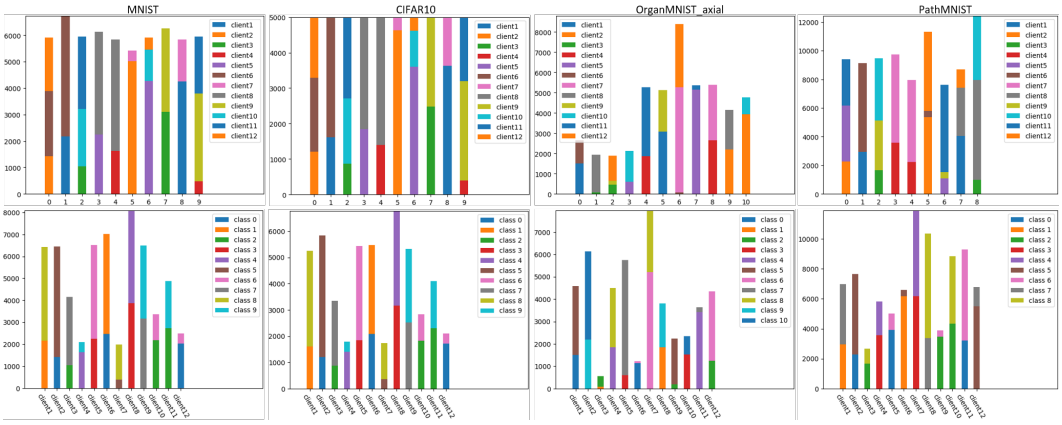

Figure 5: Data distribution of the pathological non-IID setting. The first row represents which clients the data is assigned to. The second row represents the local label distribution of each client.

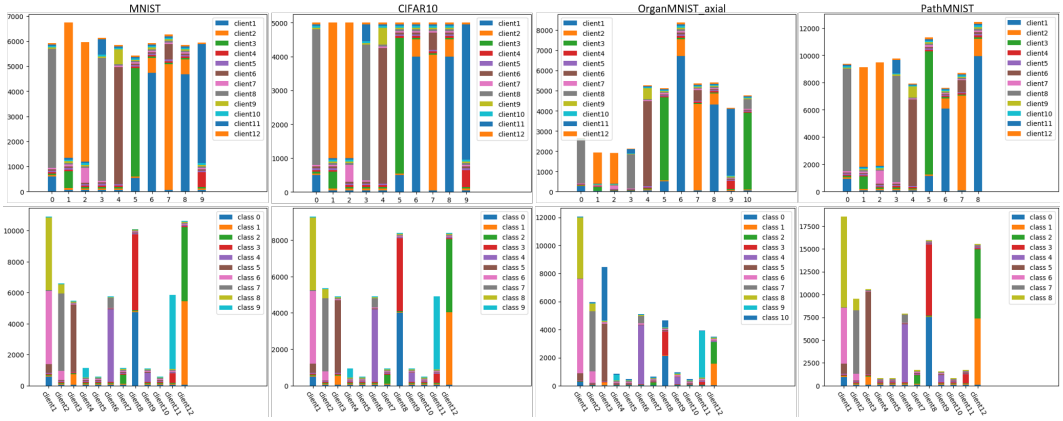

Figure 6: Data distribution of the practical non-IID setting. The first row represents which clients the data is assigned to. The second row represents the local label distribution of each client.

B EXPERIMENTS

B.1 DATASETS

We partition each dataset into pathological and practical non-IID distributions. Figure 5 and Figure 6 shows the partition of the training set with respect to "where do the images of each class go" and "what is the label distribution on each client". For example Figure 5 bottom right plot (PathMNIST dataset) shows that for the PathMNIST dataset under the pathological non-IID setting, client 1 and client 6 both contain a large portion of data from class 1, which explains why the visualization in Figure 3 for the PathMNIST dataset demonstrates that $p_{1,6}$ and $p_{1,8}$ are closer to $p_{1,1}$ than other DRs.

B.2 EMPIRICAL CONVERGENCE BEHAVIOR AND DIRECTED RELATIONSHIPS UNDER THE PRACTICAL NON-IID SETTING

Under the practical non-IID setting, the importance of learning global information will be increased due to the following major aspects of the data: the datasets on the clients (1) contain a variety of categories, increasing the variance of the data; (2) are likely to be imbalanced with a different majority class on different clients; (3) are different in size, and local training on a small dataset might quickly overfit. Consequently, personalized methods that address the global update of the model, such as `APPLE` and `FedFomo`, have natural advantages in this regard. Experimental results in

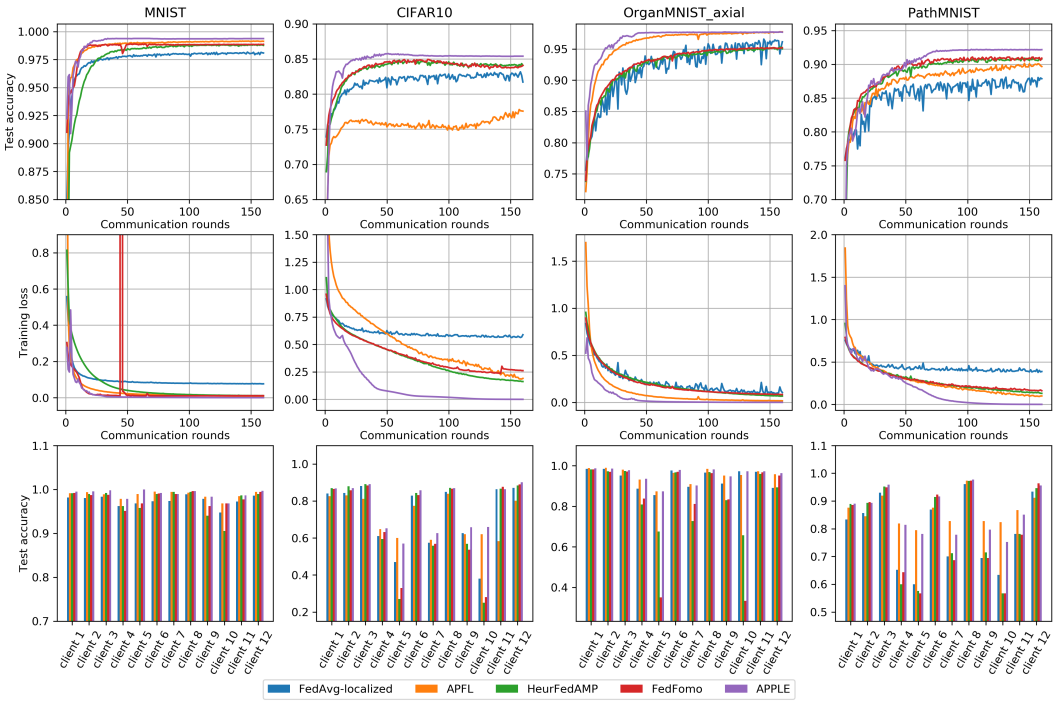

Figure 7: The training loss, test accuracy and client test accuracies of personalized methods under the practical non-IID setting.

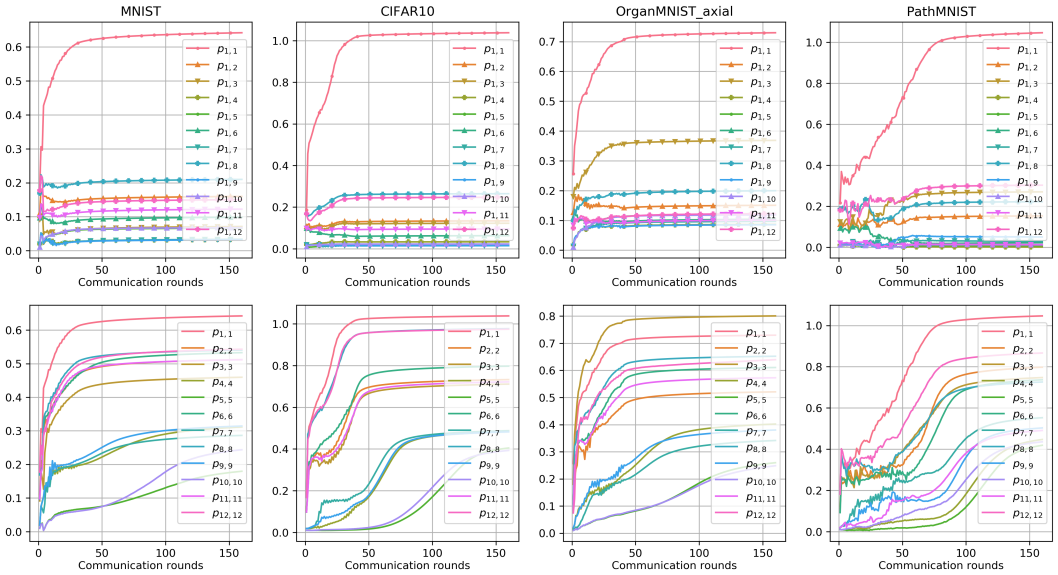

Figure 8: Directed Relationships of different datasets under the practical non-IID setting. The first row shows the DRs on client 1. The second row shows the "self-relationships", $p_{i,i}$, for each client.

Figure 7 show the advantages of `APPLE` and `FedFomo` over other compared personalized methods, and `APPLE` and `FedFomo` achieve similarly fast convergence.

Figure 8 shows the visualization of the directed relationships. Under the practical non-IID setting, since a large portion of samples in each class are assigned to only one client (recall the $80\% \times 1$, $10\% \times 1$, $1\% \times 10$ split described in Section 4.1), less can be inferred about the DRs given each

client's data distribution. We elaborate this through an example of FedFomo. FedFomo is a personalized FL method that focuses on weighting the personalized models with local validation sets. As the majority class on a different client is different, the personalized model on client j can hardly perform well on client i. This results in less weight for client j's personalized model on client i, and it fails to maximize the global information that can be learned. Our proposed method, APPLE, takes a different scheme from FedFomo. Rather than deciding the weights of other clients' personalized models purely based on the validation performance, APPLE's learnable DRs prevent the waste of other clients' core models. The learnable DRs enable to adaptively optimize the joint contribution from each downloaded core model. This is empirically demonstrated in Figure 8 (in the OrganMNIST (axial) dataset). Although client 1 and client 8 do not share any majority class (refer to Figure 6), $p_{1,8}$ can still be large, as long as the personalized model learns a beneficial assignment of each downloaded core model's contribution.

B.3 Additional Implementation Details

We used the pre-existing training set and test set of MNIST and CIFAR10. For the two datasets from MedMNIST, since the training and test datasets are different in terms of the distribution, we combined them and split it into a new training set of 80% of the entire dataset, and a new test set of the remaining 20%.

We adopted the classic four-layer CNN model. The model has two 5×5 convolutional layers followed by a fully connected layer with 500 units and another fully connected layer with the number of units equals to the number of classes.

For each method, we trained the model for 160 rounds of 5 local epochs using a batch size of 256. We used SGD as the optimizer with 0.9 momentum, and chose the best performing learning rate in $\{10^{-2}, 10^{-3}, 10^{-4}\}$, and learning rate decay in $\{1.0, 0.9964\ (= \sqrt[100]{0.7}), 0.9\}$. For APPLE, we selected the loss scheduler type from cosine and exponential, μ from $\{1.0,\ 0.1,\ 0.01\}$, and L from $\{10\%, 20\%, 30\%\}$ of the number of total training rounds. The detailed hyperparameter values are summarized in Table 3

Table 3: The hyperparameter values used in APPLE

	Pathological non-IID				Practical non-IID			
	MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST	MNIST	CIFAR10	Organ-MNIST (axial)	Path-MNIST
Net's learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
DRs' learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-4}	10^{-4}
Loss scheduler type	cos	exp.	exp.	cos	cos	cos	cos	cos
μ	0.1	0.001	0.001	1.0	0.01	0.001	0.1	1.0
L	30%	20%	20%	10%	30%	20%	20%	10%