Abstract

Personalized federated learning (FL) facilitates collaborations between multiple clients to learn personalized models without sharing private data. The mechanism mitigates the statistical heterogeneity commonly encountered in the system, i.e., non-IID data over different clients. Existing personalized algorithms generally assume all clients volunteer for personalization. However, potential participants might still be reluctant to personalize models since they might not work well. In this case, clients choose to use the global model instead. To avoid making unrealistic assumptions, we introduce the personalization rate, measured as the fraction of clients willing to train personalized models, into federated settings and propose DyPFL. This dynamically personalized FL technique incentivizes clients to participate in personalizing local models while allowing the adoption of the global model when it performs better. We show that the algorithmic pipeline in DyPFL guarantees good convergence performance, allowing it to outperform alternative personalized methods in a broad range of conditions, including variation in heterogeneity, number of clients, local epochs, and batch sizes.

1 Introduction

Personalized FL is tasked with training machine learning models for multiple clients, each with its data distribution (Fallah, Mokhtari, and Ozdaglar 2020b). The goal is to train personalized models collaboratively while accounting for client data disparities and reducing communication costs. Existing work that attempts to fulfill this goal follows three steps iteratively until convergence: (i) the server sends the latest global model to clients; (ii) the clients update the global and personalized models using their data samples; (iii) the server collects the updated global model from a subset of sampled clients to aggregate a new global model (Ma et al. 2021).

Performance Disparities in Personalized FL. While matching data distributions on clients by training personalized models among different parties, personalized FL still incurs difficulty with performance disparities among clients. Due to the heterogeneity of data, not all participants stand to benefit from the personalization. The personalized models’ performances on some clients may be worse than that using the global model, thereby hindering its adoption with more complex modalities. We demonstrate this difficulty in Figure 1 showing that personalized FL methods’ performances vary among clients. In some cases, the personalized model is worse than the global model.

Unrealistic Assumptions in Conventional Personalized FL. Existing personalized FL algorithms generally assume all clients are willing to participate in the personalization. However, training personalized models needs extra computational and communication resources, i.e., clients trade the resources’ costs for personalized models. Suppose the global model performs better than the personalized one on a client. In that case, the personalization can be hard to arrange in reality, and the client may not assent to continue training it, causing personalized FL to malfunction.

The Need for Dynamically Personalized FL. One possible solution to this problem without any assumptions about clients is using incentive mechanisms alongside dynamically personalized FL. Some prior work has tried this route in FL by directly evaluating clients’ contributions, reputation, and resource allocation (Zhan et al. 2021). However, using stateful metrics (Kang et al. 2019; Zhan et al. 2020) requires extra computations (Yu et al. 2020) or shared representations across clients (Sun et al. 2021b), limiting them only to fit in the cross-silo setting or undermining clients’ privacy by enabling inversion of representations (Geiping et al. 2020).
It is unlikely that their direct extensions will function well in the high heterogeneity and resource-constrained environment of cross-device personalized FL. Thus, a new challenge is raised to personalized FL, i.e., can we actively incentivize users to contribute to personalized training rather than training the global model?

**Contributions of this Paper**: To resolve the limitations noted above and tackle the incentive design in personalized FL, we argue that a dynamically personalized framework needs to be developed that is mindful of the challenges seen in personalized federated training. We thus propose DyPFL, a novel personalized FL method that dynamically improves the fraction of incentivized clients compared to standard personalized FL algorithms. Our primary contributions follow.

- **Notion of Incentives**: We introduce the notion of incentives into personalized FL, where the metric (personalization rate) is measured as the fraction of clients who are inspired to train personalized models. We show that the sigmoid relaxation of personalization rate is differentiable, approximates the original metric, and can be optimized using gradient-based algorithms.

- **Dynamically Personalized FL**: We propose DyPFL, a dynamically personalized method that maximizes incentivized client personalization. The incentive-based perspective arises from an analysis of clients capable of training personalized models if they perform better than the global model. DyPFL applies to general non-convex objectives with convergence guarantees.

- **Extensive Empirical Analysis**: We extensively compare DyPFL with several leading personalized FL techniques. Unlike DyPFL, personalized methods are often more sensitive to several important FL parameters (e.g., local epochs, batch size, dropped clients).

## 2 Related Work

**Incentives in FL** Incentive mechanisms for FL can be broadly divided into two categories: (i) monetary incentive mechanisms and (ii) non-monetary incentives. Monetary-based methods assume participants and the users of FL models are two separate groups, and incentives focus on providing rewards to clients to motivate them to contribute more to training. Song et al. (2019) propose RRAFL to select and pay participants based on reputation and reverse auction. Sarikaya et al. (2020) model the interaction between the server and participants as a Stackelberg game to improve the performance and optimize the commitment of local resources and the allocation of the incentive budget. Ye et al. (2020) and Kang et al. (2019) apply contract theory to design efficient incentive mechanisms to attract more participation. Zhan et al. (2020) use a deep reinforcement learning-based incentive mechanism to determine the optimal pricing strategy for the server and the participants. When the participants are also users of FL models, competition might exist among clients in heterogeneous settings. Clients that share the same global model without regard to their contributions have been shown to break down the system since the monetary incentive is no longer a strong motivator. Using this insight, Yu et al. (2020) assign each participant a different model with performance reflecting its contribution.

To our best knowledge, the proposed incentive formulation is the first personalized framework that explicitly considers the fraction of clients willing to participate in personalization. Compared with existing methods that consider the incentives in FL, we aim to maximize the personalization rate directly. Furthermore, no stateful metrics are evaluated on the server and clients, i.e., no need for extra computations on the server or sharing losses across clients.

**Personalized FL** Given the data variability in FL, personalization is an approach used to improve accuracy, and numerous work has been proposed along this line. Smith et al. (2017) explore personalized FL via a primal–dual multi-task learning framework. As summarized in Tan et al. (2022), Ma et al. (2021), the subsequent work has explored personalized FL through local customization (Fallah, Mokhtari, and Ozdaglar 2020b) Wang et al. (2019), where models are built by customizing a well-trained global model. There are several ways to achieve personalization: (i) the mixture of the global model and local models combines the global model with the clients’ latent local models (Shamsian et al. 2021 Sun et al. 2021a); (ii) meta-learning approaches build an initial meta-model that can be updated effectively using Hessian or approximations of it, and the personalized models are learned for local data samples (Fallah, Mokhtari, and Ozdaglar 2020b); (iii) local fine-tuning methods customize the global model using local datasets to learn personalized models on each client (Collins et al. 2021 Li et al. 2021).

Existing personalized FL algorithms implicitly assume all clients consent to train personalized models. In contrast, DyPFL does not impose such an unrealistic assumption on the system and explicitly maximizes the personalization rate to incentivize clients’ participation.

## 3 Dynamically Personalized Federated Learning (DyPFL)

### 3.1 Preliminaries and Problem Formulation

**Notations** Let us consider a personalized FL setup with $M$ clients and a server. For each client $k \in \{1, 2, ..., M\}$, its (personalized) loss is given by $f_k(\beta_k) = \mathbb{E}_{\xi \sim D_k}[l_1(\beta_k, \xi)]$.

**Client Incentives Settings.** In personalized FL, the $k$-th client communicates with the server to find a personalized model that minimizes its true loss function, denoting $\beta_k^* := \arg\min_{\beta_k} f_k(\beta_k)$. It also updates the global model using local dataset $B_k$ by running a few SGD steps on the empirical loss $\hat{F}_k(\omega)$. The global model may not generalize well to $k$-th client’s local distribution $D_k$ since data among
Clients are highly heterogeneous. We define that the $k$-th client is incentivized to participate in personalization (i.e., uses the personalized model) if the personalized model performs better than the global model.

**Definition 1** (Client Incentives in Personalized FL) Let the $k$-th client’s personalized model be $\beta_k$, and the global model be $\omega$, the $k$-th client is said to be incentivized to participate in personalization if $f_k(\beta_k) < f_k(\omega)$, i.e., the personalized model performs better than the global model.

Clients have a separate validation dataset on which they can compare losses of the personalized and global models to decide if they are incentivized to participate. In general, $\hat{f}_k(\omega)$ in Definition 1 acts as a performance benchmark for the personalized model and can also be replaced by a different value depending on a client’s specific need.

Conventional personalized FL does not consider client incentives in Definition 1 and implicitly assumes that all clients will participate in personalization and use the personalized model. However, this assumption may not hold in general due to the performance disparities originating from data heterogeneity (e.g., see Figure 1).

**The Measurement of Incentives.** Based on Definition 1 we formulate a personalization rate metric to explicitly measure the fraction of clients incentivized for personalization, where

$$\text{Personalization rate} = \frac{1}{M} \sum_{k=1}^{M} \mathbb{1}(f_k(\beta_k) < \hat{f}_k(\omega)).$$

(1)

Here, $\mathbb{1}$ is the indicator function. Since joining in the personalization or not is a binary decision, the personalization rate in Equation (1) only accounts for whether or not a client is incentivized but not how much a client is incentivized. Without loss of generality, one can measure the incentive margin of clients’ personalization, i.e., $\sum_k \max\{f_k(\beta_k) - \hat{f}_k(\omega), 0\}$, but this does not capture the motivation of the paper that aims to improve the number of incentivized clients in personalized FL.

### 3.2 DyPFL: Objective

**Intuitions for the Objective.** Directly maximizing the personalization rate in Equation (1) equals

$$\min_{\beta_k} \frac{1}{M} \sum_{k=1}^{M} \text{sign}(f_k(\beta_k) - \hat{f}_k(\omega)),$$

(2)

where $\text{sign}(x) = 1$ if $x \geq 0$ and 0 otherwise. However, the sign function is not differentiable, limiting the use of gradient-based methods. Additionally, clients may not know their true data distribution $D_k$ to compute $f_k(\beta_k) - \hat{f}_k(\omega)$. We propose an equivalent objective in DyPFL using an empirical loss with a differentiable sigmoid function.

Optimization methods (Nguyen and Sanner 2013; Masnadi-Shirazi and Vasconcelos 2008) commonly use a smooth differentiable function to replace a binary loss. DyPFL incorporates sigmoid function $\sigma(\cdot)$ to approximate the true objective in Equation (2) and explores its theoretical implications in Section 3.3. Moreover, an empirical estimation $\sigma(F_k(\beta_k) - \hat{F}_k(\omega))$ of the true loss is adopted since clients cannot access their true distribution $D_k$. As $\beta_k$ is updated using local training data, it is likely that $f_k(\beta_k) < \hat{f}_k(\beta_k)$. On the other hand, the global model $\omega$ is aggregated from all participants and unlikely to overfit to the local data, leading to $f_k(\omega) \sim \hat{F}_k(\omega)$. Thus, in most cases, $f_k(\beta_k) - \hat{f}_k(\omega) < F_k(\beta_k) - \hat{F}_k(\omega)$ exists. Since sigmoid is an increasing function, we have $\sigma(f_k(\beta_k) - \hat{f}_k(\omega)) < \sigma(F_k(\beta_k) - \hat{F}_k(\omega))$, indicating that minimizing the empirical loss equals minimizing an upper bound of the true objective.

**The Equivalent Objective for Maximizing the Personalization Rate.** With insights into using an empirical loss with sigmoid function, DyPFL’s objective can be formulated as

$$\text{DyPFL} : \min_B \{ \bar{F}(B) := \frac{1}{M} \sum_{k=1}^{M} \bar{F}_k(\beta_k) \},$$

(3)

where $\bar{F}_k(\beta_k) = \sigma(F_k(\beta_k) - \hat{F}_k(\omega))$.

In DyPFL, $B = [\beta_1, ..., \beta_M]$ is a $d$-by-$M$ dimensional matrix that collects $\beta_1, ..., \beta_M$ as its columns. $\omega$ is found by exploiting the model aggregation from multiple participants at the server, and $\beta_k$ is optimized to $k$-th client’s data distribution. Minimizing the proposed objective leads to a higher personalization rate and encourages more clients to train personalized models.

### 3.3 DyPFL: A Case Study

We use a simple mean estimation problem to illustrate the behaviors of the objective. Suppose $M = 2$ clients aim to find the mean of their data distribution by minimizing the truly personalized loss $f_k(\beta_k) = \mathbb{E}_{\xi_k}[(\beta_k - \xi_k)^2], \xi_k \sim \mathcal{N}(\mu_k, \sigma_k^2), k = [1, 2]$. $n_k$ samples are drawn with variance $\sigma_k^2$ around the true mean of the distribution on the $k$-th client, denoted by $B_k = \{x_{k,l}\}_{l=1}^{n_k}$ and the empirical loss is given by $F_k(\beta_k) = \frac{1}{n_k} \sum_{l=1}^{n_k} (\beta_k - x_{k,l})^2$. The global model will be $\omega = \mu_k + \frac{\mu_2}{2}$, where $\mu_k = \frac{1}{|B_k|} \sum_{l=1}^{n_k} x_{k,l}$. Assuming
Algorithm 1: DyPFL. M clients are indexed by $k$; $\eta$ and $\delta$ denote the learning rates; $T$ is the maximal number of communication rounds; $E_1$ and $E_2$ represent the number of local epochs for the global updates and personalization.

**Input:** $M, T, E_1, E_2, \eta, \delta, \omega, \{\beta_k\}_{k \in [M]}$

1. for $\tau = 0, \ldots, T - 1$
   2. The server randomly selects a subset of clients $S_\tau$ and sends the current global model $\omega^\tau$ to them
   3. for Client $k \in S_\tau$, in parallel do
      4. Solve the local sub-problem starting from $\omega^\tau$ to obtain $\omega^\tau_k$ for $E_1$ epochs
         $\omega^\tau_{k+1} = \omega^\tau_k - \eta \nabla \hat{F}_k(\omega^\tau_k)$
         */ Solve $\hat{F}_k(\beta_k)$ */
      5. Update $\beta_k$ for $E_2$ epochs
         $\beta^\tau_{k+1} = \beta^\tau_k - \delta(1 - \hat{F}_k(\beta^\tau_k))\nabla \hat{F}_k(\beta^\tau_k)$
      6. Send $\Delta^\tau_k := \omega^\tau_{k+1} - \omega^\tau$ back
      7. Server aggregates $\{\Delta^\tau_k\}_{k \in [S_\tau]}$
         $\omega^{\tau+1} \leftarrow \text{AGGREGATE}(\omega^\tau; \{\Delta^\tau_k\}_{k \in [S_\tau]})$
   8. return $\{\beta_k\}_{k \in [M]}$ (personalized model), $\omega^T$ (global model)

$n_1 = n_2 = n$, i.e., two clients have the same number of samples. Denote by $\kappa^2 = \frac{(n_1 - n_2)^2}{n_1 n_2}$ the differences between true means. The personalized model, in this case, is given by $\beta_k = a_k \hat{\mu}_k + (1 - a_k) \omega$, where $a_k \in [0, 1]$.

**Maximizing the Personalization Rate with DyPFL’s Objective.** Given the DyPFL’s objective, we first show that minimizing empirical loss with a differentiable sigmoid function leads to an increased personalization rate. Figure 2 shows that the DyPFL’s objective (Equation 3) adapts to the differences between true means $\kappa^2$. When $\kappa^2$ is small (the left sub-figure), i.e., the heterogeneity among clients is small, DyPFL encourages the global model’s training by setting the global model to be the average of local models. On the other hand, when $\kappa^2$ is large (the right sub-figure), i.e., significant client heterogeneity, the objective incentivizes clients to train personalized models.

**Theorem 1** Given the personalized minima of DyPFL’s objective $B = [\beta_1, \ldots, \beta_M]$, the expected personalization rate over datasets $B_1$ and $B_2$ is lower bounded by $\frac{1}{2} \exp(-\frac{\kappa^2}{2})$.

Since the expected personalization rate in Theorem 1 is independent of the differences between the true mean of the distributions, for $\kappa^2 \rightarrow 0$, DyPFL still incentivizes clients to personalization. The discussions on the $M$-client-mean-estimation case and detailed proof are deferred to the supplementary material.

**3.4 DyPFL: Algorithm**

To solve DyPFL’s objective, we propose jointly solving for the global model $\omega$ and personalized model $\{\beta_k\}_{k \in [M]}$ alternatingly, as summarized in Algorithm 1. Optimization proceeds in two phases: (i) updates to the global model $\omega$ are computed across the network, and then (ii) the personalized model $\beta_k$ is fit on each local client. The process of optimizing $\omega$ is the same as conventional FL algorithms: If we use iterative solvers, then at each communication round, each selected client can solve the local subproblem of approximately (Line 4). For personalization, the $k$-th client solves $\min \hat{F}_k(\beta_k)$ inexactly at each round (Line 5). Due to the alternating scheme, our solver scales well to large models. It does not introduce additional communication or privacy overheads compared with the existing solver for traditional FL algorithms.

**Insights on the Personalization Updates.** Observing that the gradient of local objective $\nabla \hat{F}_k(\beta_k) = (1 - \hat{F}_k(\beta_k))\nabla \hat{F}_k(\beta_k)$ (Line 5 in Algorithm 1) is incentive-dependent since the empirical incentive gap, $F_k(\beta_k) - \hat{F}_k(\omega)$, is dynamically updated based on the current personalized model $\beta_k$ and global model $\omega$. Let $z_k(\beta_k) = (1 - \hat{F}_k(\beta_k))\hat{F}_k(\beta_k)$. We explore the behaviors of incentive gap versus $z_k(\beta_k)$ in Figure 3. If $F_k(\beta_k) \ll \hat{F}_k(\omega)$, i.e., the $k$-th client’s personalized model performs much better than the global model, $z_k(\beta_k) \approx 0$ and DyPFL will focus on the updates of other clients. Moreover, when $F_k(\beta_k) \gg \hat{F}_k(\omega)$, i.e., the $k$-th client’s personalized model is incompatible with the client, $z_k(\beta_k) \approx 0$ and the $k$-th client will choose to adopt the global model.

$z_k(\beta_k) > 0$ when the $k$-th client’s personalized model performs similar to the global model, i.e., $F_k(\beta_k) \approx \hat{F}_k(\omega)$. DyPFL incentivizes the $k$-th client, in this case, to train the personalized model and increases the personalization rate without degrading the system’s performance.

**Adaptive Learning Rate on Clients.** Given that $F_k(\beta_k)$ is $L_k$ continuous and $L_k$ smooth, where $\forall k \in [M]$, the objective $\hat{F}(B)$ is $L_s$ smooth, where $\hat{L}_s = \frac{L}{M} \sum_{k=1}^{M} z_k(\beta_k) + \frac{\delta}{2}$. Thus, the optimal learning rate $\delta$ for DyPFL is $\delta = \frac{1}{L_s} \sqrt{\frac{M \delta}{\sum_{k=1}^{M} z_k(\beta_k) + \nu}}$, where $\delta = \frac{1}{L_s}$ and $\nu = \frac{M \delta L_s}{4L_s} > 0$. The

![Figure 3: $z_k(\beta_k)$ versus the incentive gap $F_k(\beta_k) \ll \hat{F}_k(\omega)$ is presented. $z_k(\beta_k)$ is small if (i) the $k$-th client has a more significant incentive, i.e., the personalized model is much better than the global model; or (ii) the $k$-th client cannot be incentivized at all, i.e., the global model performs much better than the personalized model. $z_k(\beta_k)$ is large when the $k$-th client is moderately incentivized, i.e., the personalized model performs similarly to the global model.](image-url)
Theorem 2
Under Assumptions 1-2, if there exists \( h(\tau) \) such that \( \lim_{\tau \to \infty} h(\tau) = 0 \), \( E[||\omega^* - \omega^\tau||^2] \leq h(\tau) \), and \( h(\tau+1) \leq h(\tau) \), then there exists \( C < \infty \) such that for any client \( k \in [M] \), \( E[||\beta_k^\tau - \beta_k^\tau||^2] \leq C h(\tau) \) with a local learning rate \( \delta = \frac{\delta_k}{\sum_{s \in S_{z_k}} z_s(\beta^\tau_k + v)} \).

Given Theorem 2, the analysis can adapt to different versions of aggregation mechanisms in the server. For instance, when the server aggregates updates using the weighted average method (as FedAvg), the algorithm converges to its local optimum with the rate \( O(\frac{1}{n}) \). We defer the proof of convergence and a case using weighted average aggregation to the supplementary material.

### 4 Experiments
We evaluate DyPFL to answer the following key questions: (i) How much performance improvement (in terms of testing accuracy and personalization rate) does DyPFL achieve over existing personalized FL baselines? (ii) How sensitive is DyPFL with different choices of hyperparameters? (iii) How does DyPFL perform when it jointly operates on the real-world dataset?

#### 4.1 Experimental Setup

**Datasets** To simulate FL tasks in DyPFL’s evaluation, we use four benchmark datasets and a real-world dataset, which can be categorized into three different domains as follows:

- **Image Classification**: We evaluate DyPFL on two image recognition datasets: FEMNIST (Caldas et al. 2018) and Celeba (Liu et al. 2015). FEMNIST dataset contains images of hand-written digits and characters from 712 clients with total 157k samples. Celeba dataset contains face attributes of 915 clients with 19k samples. We use ResNet50 for both datasets (He et al. 2016).
- **Natural Language Processing (NLP)**: We evaluate DyPFL on two NLP tasks with different datasets: next-word prediction on the Reddit (Caldas et al. 2018) dataset and next-character prediction on the Shakespeare (McMahan et al. 2017) dataset. The Reddit dataset contains posts from 813 clients with 32k samples, and the Shakespeare dataset contains 845k samples separated into 171 clients. We use LSTM models for both datasets.
- **Real-world Dataset**: We evaluate DyPFL on RSNA Intracranial Hemorrhage Detection dataset (Flanders et al. 2020), where 106k magnetic resonance image (MRI) samples are classified into six classes. We adopt the ResNet18 model to classify brain diseases.

**Baselines** We compare DyPFL with state-of-the-art personalized FL methods: Per-FedAvg (Fallah, Mokhtari, and Ozdaglar 2020), pFedMe (T. Dinh, Tran, and Nguyen 2020), and FedAMP (Huang et al. 2021a). These personalized FL algorithms implicitly assume that all clients are willing to join in personalization. We relax this unrealistic assumption and modify methods with a flexible decision, i.e., clients adopt the personalized model if its performance is better than the global model and vice versa. Results on conventional FL algorithms such as FedAvg are also provided in experiments.

**Metrics** We mainly evaluate the average testing accuracy on preferred models and personalization rate (Equation (1)). The preferred models’ testing accuracy indicates the average testing accuracy on clients’ personalized or global models. All results are presented after 50 runs with different random seeds.

**Implementation** All experiments are implemented using PyTorch (Paszke et al. 2019) and run on a cluster where

| Task | Image Classification | NLP |
|------|----------------------|-----|
| Dataset(Non-IID) | FEMNIST | Celeba | Reddit | Shakespeare |
| Methods | P. Rate | ACC. | P. Rate | ACC. | P. Rate | ACC. | P. Rate | ACC. |
| FedAvg | .00(.00) | .591(.207) | .00(.00) | .813(.325) | .00(.00) | .291(.001) | .00(.00) | .394(.318) |
| Per-FedAvg | .75(.25) | .771(.128) | .86(.19) | .852(.006) | .70(.24) | .415(.005) | .66(.28) | .552(.310) |
| pFedMe | .78(.16) | .793(.064) | .85(.18) | .849(.004) | .81(.21) | .448(.001) | .79(.13) | .570(.143) |
| FedAMP | .83(.14) | .804(.127) | .89(.17) | .861(.005) | .78(.34) | .427(.001) | .76(.11) | .565(.151) |
| DyPFL | .91(.11) | .838(.045) | .94(.06) | .878(.007) | .85(.27) | .465(.001) | .82(.13) | .613(.114) |

Table 1: Summary of average (std) personalization rate and average (std) testing accuracy after 50 runs on benchmark datasets (non-IID). DyPFL shows (i) improved personalization rate and (ii) better average testing accuracy over other baselines.
| Task                | Image Classification | NLP               |
|--------------------|----------------------|-------------------|
| Dataset(IID)       | FEMNIST | Celeba | Reddit | Shakespeare |
| Methods            | P. Rate | ACC. | P. Rate | ACC. | P. Rate | ACC. | P. Rate | ACC. |
| FedAvg             | .00(.00) | .743(.139) | .00(.00) | .881(.204) | .00(.00) | .481(.109) | .00(.00) | .537(.152) |
| Per-FedAvg         | .21(.14) | .749(.112) | .15(.22) | .879(.162) | .22(.18) | .459(.173) | .34(.28) | .541(.260) |
| pFedMe             | .25(.09) | .757(.081) | .14(.19) | .872(.145) | .27(.21) | .496(.165) | .38(.13) | .558(.181) |
| FedAMP             | .27(.11) | .764(.104) | .17(.25) | .889(.107) | .25(.23) | .479(.190) | .35(.16) | .549(.174) |
| DyPFL              | .30(.09) | .773(.056) | .24(.09) | .904(.116) | .29(.15) | .499(.089) | .42(.11) | .585(.168) |

Table 2: Summary of average (std) personalization rate and average (std) testing accuracy after 50 runs on benchmark datasets (IID). DyPFL still improves the personalization rate and average testing accuracy on preferred models compared with baselines, while the improvement is smaller than in non-IID settings.

Figure 4: Performance of DyPFL compared with baselines for heterogeneous local updating epochs. The evolution of the mean testing accuracy is summarized with a random number of local epochs, which expectation equals 10. DyPFL is robust to heterogeneous local epochs and converges better than baselines.

Each node is equipped with 4 Tesla T4 GPUs and 64 Intel(R) Xeon(R) CPU E5-2683 v4 cores @ 2.10GHz. We test the following parameters for DyPFL: We configure a learning rate of 0.001 for FEMNIST and Celeba, 2 for Reddit, and 0.8 for Shakespeare. This is a set of parameters that we recommend trying with tasks. For reference, datasets’ statistics, data partition details, and implementation settings are deferred to Appendix.

4.2 Performance Comparison: Non-IID Settings

We first conduct experiments on four benchmark datasets in non-IID settings as proofs-of-concept. For all tasks, we assign each client sample from exactly two classes. In Table 1, we show that for all benchmark datasets, DyPFL incentivizes (at least 5%) more clients compared with other modified personalized FL baselines: DyPFL reaches the personalization rate of 1.05×1.16× higher than baselines on image classification tasks. Our algorithm achieves 1.03×1.16× more client personalization on NLP datasets than baselines.

The performance of FedAvg is an excellent baseline to indicate the need for personalization on non-IID datasets since it does not conduct personalization at all. We notice that DyPFL consistently shows the highest personalization rate on all datasets, while the performance of baselines is inconsistent. For example, FedAMP shows higher personalization rate performance on image tasks but deficient performance on NLP tasks. In contrast, FedAMP performs better on NLP tasks among baselines but provides a lower personalization rate on image tasks.

DyPFL also achieves the highest testing accuracy on preferred models amongst baselines, which improves the personalization rate without sacrificing the final accuracy.

4.3 Performance Comparison: IID Settings

Intuitively, one can expect that if data among clients are IID, the trained global model may sufficiently generalize to test data. Hence, clients will be less incentivized to personalize. We explore behaviors of DyPFL on four benchmark datasets in IID settings and compare its personalization rate and testing accuracy on preferred models with baselines.

Table 2 shows the personalization rate and testing accuracy of all methods being compared under the IID setting. The global method FedAvg achieves performance comparable with all the other methods on the IID data because the clients are similar, and the global model fits every client well.

The personalized FL methods Per-FedAvg and pFedMe do not perform as well as FedAvg on the simple Celeba dataset. The local fine-tuning steps of these algorithms are prone to over-fitting. However, on the more challenging datasets FEMNIST, Reddit, and Shakespeare, the performances of pFedMe and FedAMP are better than that of FedAvg due to clients’ personalization.

DyPFL performs much better than conventional personalized FL algorithms under the IID setting and incentivizes more clients to join in the personalization. The personalized
| EMNIST(Non-IID) | Batch Size | Dropped Clients |
|----------------|------------|-----------------|
| Methods        | 16         | 64              | 128           | 512       | ∞   | 10% | 30% | 50% |
| FedAvg         | .435(.389) | .457(.296)      | .591(.207)    | .552(.304) | .528(.285) | .499(.351) | .473(.396) | .428(.402) |
| Per-FedAvg     | .653(.295) | .691(.247)      | .771(.128)    | .714(.180) | .695(.203) | .732(.149) | .701(.197) | .624(.256) |
| pFedMe         | .607(.258) | .681(.209)      | .793(.064)    | .744(.167) | .702(.225) | .726(.180) | .712(.238) | .624(.375) |
| FedAMP         | .693(.286) | .761(.198)      | .804(.127)    | .782(.175) | .762(.204) | .785(.159) | .751(.180) | .643(.299) |
| DyPFL          | .768(.163) | .795(.084)      | .838(.045)    | .807(.119) | .797(.158) | .816(.091) | .791(.135) | .752(.180) |

Table 3: Performance of DyPFL compared with baselines for a different number of local batch sizes and dropped clients on EMNIST. DyPFL can handle a wide range of batch sizes and dropped clients.

![Image of medical samples](image_url)

Figure 5: Samples of each class in RSNA Intracranial Hemorrhage Detection Dataset are presented.

Intracranial Hemorrhage: Epidural, Intraparenchymal, Intraventricular, Subarachnoid, Subdural, Healthy

| Intracranial Hemorrhage | Hospitals (Clients) |
|-------------------------|---------------------|
| Methods                 | 12                  | 48                  |
| FedAvg                  | .532(1.93)          | .526(3.06)          |
| FedAMP                  | .705(2.97)          | .689(3.38)          |
| DyPFL                   | .751(1.86)          | .748(2.25)          |

Table 4: Comparisons of DyPFL with baseline methods for the different numbers of hospitals using the real-world intracranial hemorrhage detection dataset are summarized. Results of five runs are reported in mean testing accuracy (std). The best methods are highlighted in boldface.

Models of clients are similar to the global model. Thus the inventive gap $F_k(\beta_k) - \hat{F}_k(\omega)$ is around zero, incentivizing clients’ personalization (Figure 3). DyPFL achieves the best performance among all the personalized FL methods on all datasets.

4.4 Ablation Studies

Tolerance to Local Updating Epochs: Because of the heterogeneity in the system, clients may endure different local training epochs at different times. One possible effect of this scenario is that some bad local models result from the local training with a small number of epochs. To simulate the heterogeneous training, we train each client by a random number of epochs ($E_1$ and $E_2$) with an expectation of 10 during each communication round. Specifically, this random number is uniformly drawn from an integer between $[1, 19]$. To analyze impacts of heterogeneous training, we plot the mean accuracy versus the communication round for all the methods in Figure 4. We observe that DyPFL has a high tolerance to heterogeneous training and converges to overall higher testing accuracy on all four datasets than baselines.

Effects of Batch Size: The training batch size is an essential factor in optimization algorithms. We empirically evaluate how the choice of local batch size affects the model performance in Table 3. We train the model using different batch size $\{16, 64, 128, 512, \infty\}$ with fixed local epochs. $\infty$ stands for loading all of the training samples into a single batch. The results indicate that DyPFL can effectively work under a wide range of batch sizes.

Tolerance to Dropped Clients: To address unreliable operating environment challenge, we conduct the dropped clients’ experiments for DyPFL and other baselines. The results of 10%, 30%, and 50% randomly dropped clients in each round for the four non-IID benchmark datasets are shown in Table 3. We first observe that, in general, DyPFL can converge to higher mean testing accuracy than baselines for all datasets. These results demonstrate that DyPFL can robustly handle clients’ dropping. Benefiting from the incentive mechanism, DyPFL is not influenced by the dropped clients as it can adaptively incentivize online clients.

4.5 Case Study on Real-world Problem: Intracranial Hemorrhage Detection

To demonstrate the feasibility of DyPFL to the real problem, we evaluate the algorithm on a medical application for the diagnosis of brain disease, where we consider a clinical setting for disease classification using RSNA Intracranial Hemorrhage Detection dataset.

Figure 5 shows samples from six classes, including five types of intracranial hemorrhage (Epidural, Intraparenchymal, Intraventricular, Subarachnoid, and Subdural) and a healthy type. The system contains 48 hospitals (clients). We
use ReNet18 with the cross-entropy loss and Adam (Kingma and Ba, 2014) optimizer with a learning rate of $1e^{-4}$.

We report the results of mean personalization rate (std) and testing accuracy on preferred models (std) over five runs in Table 4. We can see that our proposed DyPFL method consistently outperforms baselines by 10% for different numbers of hospitals. The superiority of DyPFL in this challenging real-world FL setting further indicates the effectiveness and robustness of our algorithm. These results are inspiring and bring the hope of deploying DyPFL to the healthcare field, where data are often heterogeneous and distributed.

5 Conclusion

In this paper, we relax the fundamental assumption in personalized FL that clients always stand to benefit from personalization. We formalize the notion of incentives in the system by evaluating whether the personalized model performs better than the global model and introduce a novel metric, personalization rate, to measure the fraction of incentivized clients in the system explicitly. We propose DyPFL to maximize the personalization rate and provide convergence guarantees for the algorithm. We show that DyPFL can significantly outperform the baseline methods under different settings through extensive experiments on benchmark and real-world datasets. As a wrapper, we expect DyPFL can be further improved by incorporating advanced FL aggregation or optimization schemes for the non-IID settings.

References

Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In International Conference on Machine Learning, 2089–2099. PMLR.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020a. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In International Conference on Artificial Intelligence and Statistics, 1082–1092. PMLR.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020b. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. Advances in Neural Information Processing Systems, 33: 3557–3568.

Flanders, A. E.; Prevedello, L. M.; Shih, G.; Halabi, S. S.; Kalpathy-Cramer, J.; Bull, R.; Mongan, J. T.; Stein, A.; Kitamura, F. C.; Lungren, M. P.; et al. 2020. Construction of a machine learning dataset through collaboration: the RSNA 2019 brain CT hemorrhage challenge. Radiology: Artificial Intelligence, 2(3): e190211.

Geiping, J.; Bauermeister, H.; Droge, H.; and Moeller, M. 2020. Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems, 33: 16937–16947.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021a. Personalized Cross-Silo Federated Learning on Non-IID Data. In AAAI, 7865–7873.

Huang, Y.; Gupta, S.; Song, Z.; Li, K.; and Arora, S. 2021b. Evaluating gradient inversion attacks and defenses in federated learning. Advances in Neural Information Processing Systems, 34: 7232–7241.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2): 1–210.

Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; and Zhang, J. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal, 6(6): 10700–10714.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and robust federated learning through personalization. In International Conference on Machine Learning, 6357–6368. PMLR.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, 3730–3738.

Lyu, L.; Xu, X.; Wang, Q.; and Yu, H. 2020. Collaborative fairness in federated learning. In Federated Learning, 189–204. Springer.

Ma, Z.; Lu, Y.; Li, W.; Yi, J.; and Cui, S. 2021. PFedAtt: Attention-based Personalized Federated Learning on Heterogeneous Clients. In Asian Conference on Machine Learning, 1253–1268. PMLR.

Masnadi-Shirazi, H.; and Vasconcelos, N. 2008. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. Advances in neural information processing systems, 21.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, 1273–1282. PMLR.

Nguyen, T.; and Sanner, S. 2013. Algorithms for direct 0–1 loss optimization in binary classification. In International Conference on Machine Learning, 1085–1093. PMLR.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1711.04239.

Shamsian, A.; Navon, A.; Fetaya, E.; and Chechik, G. 2021. Personalized federated learning using hypernetworks. In International Conference on Machine Learning, 9489–9502. PMLR.
Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. *Advances in neural information processing systems*, 30.

Song, T.; Tong, Y.; and Wei, S. 2019. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, 2577–2586. IEEE.

Sun, B.; Huo, H.; Yang, Y.; and Bai, B. 2021a. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34: 23309–23320.

Sun, P.; Che, H.; Wang, Z.; Wang, Y.; Wang, T.; Wu, L.; and Shao, H. 2021b. Pain-FL: Personalized privacy-preserving incentive for federated learning. *IEEE Journal on Selected Areas in Communications*, 39(12): 3805–3820.

T Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33: 21394–21405.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, S.; Tuor, T.; Salonidis, T.; Leung, K. K.; Makaya, C.; He, T.; and Chan, K. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6): 1205–1221.

Ye, D.; Yu, R.; Pan, M.; and Han, Z. 2020. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8: 23920–23935.

Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; and Yang, Q. 2020. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 393–399.

Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; and Guo, S. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7): 6360–6368.

Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Li, P.; and Guo, S. 2021. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*.