LEARNING SHAREABLE BASES FOR PERSONALIZED FEDERATED IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized federated learning (PFL) aims to leverage the collective wisdom of clients' data while constructing customized models that are tailored to individual client's data distributions. The existing work of PFL mostly aims to personalize for participating clients. In this paper, we focus on a less studied but practically important scenario—generating a personalized model for a new client efficiently. Different from most previous approaches that learn a whole or partial network for each client, we explicitly model the clients' overall meta distribution and embed each client into a low dimension space. We propose FEDBASIS, a novel PFL algorithm that learns a set of few, shareable *basis* models, upon which each client only needs to learn the coefficients for combining them into a personalized network. FEDBASIS is parameter-efficient, robust, and more accurate compared to other competitive PFL baselines, especially in a low data regime, without increasing the inference cost. To demonstrate its applicability, we further present a PFL evaluation protocol for image classification, featuring larger data discrepancies across clients in both the image and label spaces as well as more faithful training and test splits.

1 INTRODUCTION

Recent years have witnessed a gradual shift in computer vision and machine learning from simply building a stronger model (e.g., image classifier) to taking more users' aspects into account. For instance, more attention has been paid to data privacy and ownership in collecting data for model training (Jordan & Mitchell, 2015; Papernot et al., 2016). Building models that are tailored to users' data, preferences, and characteristics have been shown to greatly improve user experience (Rudovic et al., 2018). Personalized federated learning (PFL) is a relatively new machine learning paradigm that can potentially fulfill the demands of both worlds (Kulkarni et al., 2020). On the one hand, it follows the setup of federated learning (FL): training models with decentralized data held by users (i.e., clients) (Kairouz et al., 2019). On the other hand, it aims to construct customized models for individual clients that would perform well for their respective data distributions.

While appealing, existing work of PFL has mainly focused on how to train the personalized models, e.g., via federated multi-task learning (Li et al., 2020a; Smith et al., 2017), model interpolation (Mansour et al., 2020), fine-tuning (Chen & Chao, 2022; Yu et al., 2020), etc. Specifically, existing algorithms mostly require saving for each client a whole or partial model (e.g., a ConvNet classifier or feature extractor). This implies a linear parameter complexity with respect to the number of clients, which is parameter-inefficient and unfavorable for personalized cloud service — the overall system needs a linear space of storage, not to mention the efforts for profiling, versioning, and provenance, for every client. Less attention has been paid to how to deploy and maintain the personalized system.

A practical challenge of previous work is how to fulfill new clients' queries, who did not involve in the training phase. Beyond training personalized models for the participated clients only, we focus on *preparing to serve new clients with fast, data-efficient personalization*. A promising solution is Model Agnostic Meta-Learning (MAML) (Finn et al., 2017) that aims to learn a good initialization such that it can be adapted to a new task fast, e.g., in a few SGD steps. The model-based idea has been inserted into PFL as well, by learning a model ready to be fine-tuned on each client's local data (Fallah et al., 2020). However, it still learns the parameters of a whole or partial model for each client. Several recent studies (Pillutla et al., 2022; Wu et al., 2022; Fallah et al., 2020) show that when individual clients' data are scarce, fine-tuning may suffer from overfitting and being sensitive

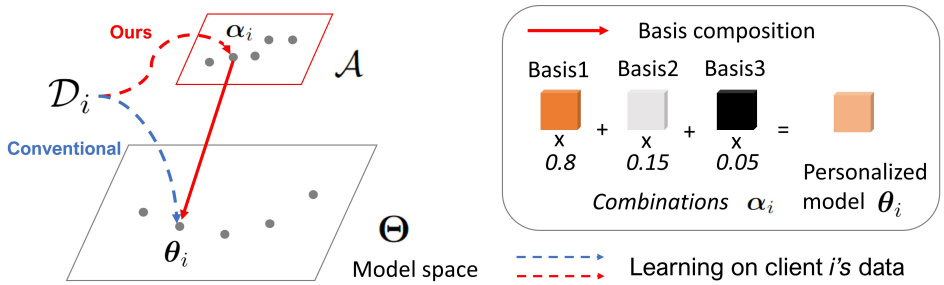

Figure 1: In conventional PFL, each client learns a high-dimensional model, the overall parameters scale with number of clients. In **our FEDBASIS**, we learn a few sharable basis models of the same network architecture. After the basis models are trained, a new client only needs to learn a short combination vector as the coefficients to combine them in the parameter space into a personalized network thus more data efficient and robust.

to hyperparameters such as learning rates and the number of steps, eventually hurting some clients' test performance, though the average personalized performance could be improved.

To address such a dilemma, we propose to improve the robustness of a personalization system by *reducing the overall parameter complexity*. Specifically, we aim to decouple the required total number of personalized parameters from the number of clients. We hypothesize that the clients' local distributions are not disjoint and could share some latent variables (e.g., domains, superclasses, etc). Learning a separate personalized model for each client could be redundant and unfavorable for the generalization of new data/clients. Specifically, we are interested in *learning a meta-model that can generate a personalized model for every client* such that the overall parameter complexity is bounded by the size of such a meta-model while providing flexibility to adapt the whole network.

We propose a novel model architecture and learning algorithm for PFL. Our idea is to learn a few, shareable *basis* models of the same architecture, which can be combined layer by layer to form a personalized model with learnable combination coefficients, inspired by (Changpinyo et al., 2016; Evgeniou & Pontil, 2007). The inference memory footprint and computation cost of the combined personalized model do not scale with #basis. An illustration is in Figure 1. It can be treated as Principal Component Analysis (PCA) on the collections of high-dimensional neural networks, *essentially learning sharable bases across clients*.

Learning the *basis* models in a federated setting, however, is nontrivial. As will be discussed in section 4, naively training them via the FEDAVG procedure (McMahan et al., 2017) — i.e., iterating between local model training for multiple epochs and global aggregation — would simply result in non-specialized bases that are unable to construct personalized models. We, therefore, present an improved coordinate descent style federated algorithm to overcome this problem. We name this architecture and algorithm FEDBASIS. FEDBASIS enjoys several desired properties. It maintains built-in overall parameter efficiency but also maintains high personalized classification accuracy. After the basis models are trained, a new client only needs to learn very few parameters, i.e., the coefficients for combining them, to accommodate the distribution discrepancy, which is more robust to learning rates and the training size. Last but not least, FEDBASIS is a stateless algorithm and does not increase inference-time cost, suitably for cross-device deployment.

To demonstrate the applicability and generalizability of FEDBASIS, we further present PFLBED, a set of benchmark datasets for cross-domain PFL. We point out some existing PFL evaluations that either pose huge distribution mismatches between training and testing (thus misleading) (Caldas et al., 2018; Li et al., 2020a) or focus on the cases that only either labels or the input domains are non-IID across clients (thus less comprehensive) (Chen & Chao, 2022; Sun et al., 2021). PFLBED is carefully designed to resolve both problems. Concretely, we split the datasets into personalized portions according to domains by leveraging either domain annotated datasets (Li et al., 2017; Venkateswara et al., 2017) or natural attributes like users, PFLBED is able to capture more diverse and realistic PFL scenarios to reflect real-world challenges.

2 RELATED WORK

Many approaches have been developed to improve different dimensions of PFL. We focus on a less studied route by learning a meta-model to summarize all the client models. Our FEDBASIS

architectures were inspired by networks proposed to improve the accuracy of a single neural network model in centralized learning (Yang et al., 2019; Chen et al., 2020; Zhang et al., 2021c). Our novelty is in extending such a concept to PFL, identifying difficulties in optimization, and resolving them accordingly. In the following, we summarize existing PFL approaches.

Multi-task learning (MTL). Many previous works formulate personalization over a group of clients as multi-task (MTL) (Zhang & Yang, 2017; Ruder, 2017; Evgeniou & Pontil, 2004; 2007; Jacob et al., 2009; Zhang & Yeung, 2010) — leveraging the clients' task relatedness to improve model generalizability. These methods typically focus on regularizer designs while each client learns for its own model. For instance, (Smith et al., 2017; Zhang et al., 2021a) encouraged related clients to learn similar models; (Li et al., 2020a; Dinh et al., 2020; Deng et al., 2020; Hanzely et al., 2020; Hanzely & Richtárik, 2020; Corinzia & Buhmann, 2019; Li & Wang, 2019) regularized local models with a learnable global model, prior, or set of data logits.

Mixture of models. Assuming the data distribution of each client is a mixture of underlying distributions, another approach is based on mixture models: (Peterson et al., 2019; Agarwal et al., 2020; Zec et al., 2020; Marfoq et al., 2021) (separately) learned global and personalized models and performed a mixture of them for prediction. However, the computation cost in inference scales linearly to the number of the models in the mixture since this approach aggregates the experts on outputs but not on the model weights like ours (which is arguably more challenging).

Personalized layers. Which layers/components in a network should be personalized (to tailor local distributions) or be shared (to collaborate across clients' data) is a crucial question that attract many research (Shen et al., 2022; Liang et al., 2020; Li et al., 2021b; Bui et al., 2019; Arivazhagan et al., 2019). Some works (Ma et al., 2022; Sun et al., 2021) propose learning-based methods for such decisions. Our goal is to summarize all personalized parameters thus orthogonal to this direction. In this paper, we consider the whole network adaptable but combining these techniques to select a partial network to further improve will be our future work.

General representations. Another line of research focuses on learning a universal feature extractor. To generate the personalized model, each client only needs to learn an output head (Collins et al., 2021; Chen & Chao, 2022), a Gaussian process tree classifier (Achituve et al., 2021), or a k-NN classifier (Marfoq et al., 2022). Such an approach is simple and remarkably strong on single-domain class-non-IID PFL but likely sub-optimal when the input domains are also non-IID (different styles, locations, data collection, etc). We agree on the concept of learning powerful, general representation but go beyond single-domain features and maintain multiple basis models shared by all clients to serve the cross-domain scenarios.

Meta-learning. Meta-learning is the most relevant to ours that also learns a meta-model that can be personalized rapidly (Khodak et al., 2019; Chen et al., 2018; Fallah et al., 2020; Jiang et al., 2019). It requires to split/reuse the training data as a meta-validation set, which might not be favorable if the training size is small. Other algorithms model the relationships between clients (Zhang et al., 2021b; Huang et al., 2021) for initializing, or regularize personalized models. However, they still need a linear parameter complexity for the final personalized models. We are inspired by (Evgeniou & Pontil, 2007) and formulate each client model as a linear combination of a few basis models. This makes our work clearly different from existing PFL works as we bypass the linear parameter complexity.

The closest work to ours is (Shamsian et al., 2021) that summarizes local models into a HyperNetwork (Ha et al., 2017). Our work is inspired by the concept of reducing model complexity, but a more effective and scalable implementation. We provide a more detailed comparison in subsection 4.3.

3 BACKGROUND

We first provide a short background. In **generic federated learning (GFL)**, the goal remains the same — to train a "global" model $h_{\boldsymbol{\theta}}$, say a classifier parameterized by $\boldsymbol{\theta}$. However, the training data are now collected and separately stored by M clients: each client $m \in [M]$ keeps a private set $\mathcal{D}_m = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{D}_m|}$, where $\boldsymbol{x}$ is the input (e.g., images) and $y \in \{1, \cdots, C\} = [C]$ is the truth label. Given the loss function ℓ (e.g., cross-entropy). Let $\mathcal{D} = \cup_m \mathcal{D}_m$ denote the *pseudo* aggregated

data from all clients and $\mathcal{L}_m$ denote the empirical risk of client m, the problem is

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{m=1}^{M} \frac{|\mathcal{D}_m|}{|\mathcal{D}|} \mathcal{L}_m(\boldsymbol{\theta}), \quad \mathcal{L}_m(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_m|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_m} \ell(y_i, h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)). \tag{1}$$

Since the data are decentralized, Equation 1 cannot be solved directly. The standard **federated averaging (FEDAVG)** (McMahan et al., 2017) algorithm decomposes the optimization into a multi-round process of iterations between local training at the clients and global aggregation at the server. Let $\bar{\boldsymbol{\theta}}^{(t)}/\tilde{\boldsymbol{\theta}}_m^{(t)}$ denote the global/local model after round t, the two steps can be formulated as

$$\textbf{Local: } \tilde{\boldsymbol{\theta}}_m^{(t)} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta}), \text{ initialized by } \bar{\boldsymbol{\theta}}^{(t-1)}; \qquad \textbf{Global: } \bar{\boldsymbol{\theta}}^{(t)} \leftarrow \sum_{m=1}^{M} \frac{|\mathcal{D}_m|}{|\mathcal{D}|} \tilde{\boldsymbol{\theta}}_m^{(t)}. \tag{2}$$

In contrast to GFL, **personalized federated learning (PFL)** aims to learn for each client m a customized model $\boldsymbol{\theta}_m$ to perform well on client m's data. While there is no agreed objective function, many existing works (Smith et al., 2017; Li et al., 2020a; Dinh et al., 2020; Hanzely et al., 2020; Hanzely & Richtárik, 2020; Li & Wang, 2019) solve an optimization problem similar to

$$\min_{\{\boldsymbol{\Omega}, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M\}} \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_m(\boldsymbol{\theta}_m) + \mathcal{R}(\boldsymbol{\Omega}, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M), \tag{3}$$

where $\mathcal{R}$ is a regularizer; $\boldsymbol{\Omega}$ is introduced to relate clients to encourage learning similar models for overcoming their limited data. Unlike Equation 1, Equation 3 seeks to minimize each client's empirical risk (plus a regularizer) by the personalized model $\boldsymbol{\theta}_m$ but not a single global model $\boldsymbol{\theta}$.

Our assumption. The clients' local data share similarity (e.g., domains, styles, classes, etc) — a common assumption made in multi-task learning (Evgeniou & Pontil, 2007) — it is likely that we can use a a much smaller set of models $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K\}$, $K \ll M$, $|\boldsymbol{v}| = |\boldsymbol{\theta}|$, to construct high-quality personalized models and meanwhile largely reduce the number of parameters.

4 FEDBASIS: PERSONALIZED FEDERATED LEARNING WITH BASES

4.1 MOTIVATION AND FORMULATION

Reducing overall parameter complexity. While both solving Equation 3 and fine-tuning the FEDAVG's global model can lead to personalized models, they require learning and saving the parameters of a whole (or partial) model for each of the M clients — i.e., linear parameter complexity $\mathcal{O}(M \times |\boldsymbol{\theta}|)$. This is particularly redundant when a huge number of clients are involved and their distributions are similar. Besides, model parameters learned specifically for each client would be vulnerable to overfitting, even with regularization.

To resolve these issues in PFL, we propose a novel way to bypass the linear parameter complexity, inspired by (Changpinyo et al., 2016; Evgeniou & Pontil, 2007). We represent each personalized model $\boldsymbol{\theta}_m$ by

$$\boldsymbol{\theta}_m(\boldsymbol{\alpha}_m, \mathcal{V}) = \sum_{k} \boldsymbol{\alpha}_m[k] \times \boldsymbol{v}_k, \tag{4}$$

where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K\}$ is a set of K *basis* models shareable among clients, and $\boldsymbol{\alpha}_m \in \Delta^{K-1}$ is a K-dimensional vector on the $(K-1)$-simplex that records the personalized convex combination coefficients. That is, each personalized model is a convex combination of a set of basis models.

With this representation, the total parameters to save for all clients become

$$\mathcal{O}(K \times |\boldsymbol{\theta}| + K \times M) \simeq \mathcal{O}(K \times |\boldsymbol{\theta}|). \tag{5}$$

Here, $\mathcal{O}(K \times M)$ corresponds to all the combination coefficients $\mathcal{A} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M\}$, which is negligible since for most of the modern neural network models, $|\boldsymbol{\theta}| \gg M$.

Objective function. Building upon the model representation in Equation 4 and the optimization in Equation 3, we define our FEDBASIS PFL problem as

$$\min_{\mathcal{A}=\{\boldsymbol{\alpha}_m\}_{m=1}^M, \mathcal{V}=\{\boldsymbol{v}_k\}_{k=1}^K} \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_m(\boldsymbol{\theta}_m), \quad \text{where } \boldsymbol{\theta}_m = \sum_k \boldsymbol{\alpha}_m[k] \times \boldsymbol{v}_k. \tag{6}$$

We note that both the basis models and the combination coefficient vectors are to be learned. We drop the regularization term in Equation 3 as the convex combination itself is a form of regularization (Evgeniou & Pontil, 2007), where we implement $\boldsymbol{\alpha}$ by a softmax function in our experiments.

Training. We discuss how to optimize Equation 6 in subsection 4.2.

Personalization for new clients. To generate the personalized model for a client, the client receives $\mathcal{V}$ and finds its combination coefficients $\boldsymbol{\alpha}_m$ by SGD with its local data. Since $|\boldsymbol{\alpha}_m|$ is merely K parameter per client, it can be robustly learned with fewer data. A single personalized model $\boldsymbol{\theta}_m$ is then constructed by convexly combining the parameters of basis models in $\mathcal{V}$ layer-by-layer, according to $\boldsymbol{\alpha}_m$. The inference time on each image thus remains a constant. This is sharply different from the mixture of experts (Reisser et al., 2021), which combines the predictions of expert models, not their parameters— the inference cost of prediction on each image is #expert times more.

4.2 FEDERATED LEARNING ALGORITHM

Similarly to Equation 1, Equation 6 cannot be solved directly since the clients' data are decentralized. A baseline training algorithm is to learn $\mathcal{V}$ with FEDAVG directly

$$\begin{aligned} &\text{Local:} \quad \{\boldsymbol{\alpha}_m^{(t)}, \tilde{\mathcal{V}}_m^{(t)}\} = \arg\min_{\{\boldsymbol{\alpha}, \mathcal{V}\}} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V}), \text{ initialized by } \{\frac{\mathbf{1}}{\boldsymbol{K}} \times K, \bar{\mathcal{V}}^{(t-1)}\}, \\ &\text{Global:} \quad \bar{\mathcal{V}}^{(t)} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \tilde{\mathcal{V}}_m^{(t)}. \end{aligned} \tag{7}$$

We use $\mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V})$ as a concise notation for $\mathcal{L}_m(\boldsymbol{\theta} = \sum_k \boldsymbol{\alpha}[k] \times \boldsymbol{v}_k)$. Note that in local training, client m only updates her own coefficients $\boldsymbol{\alpha}_m^{(t)}$, not others'; all basis models in $\tilde{\mathcal{V}}_m^{(t)}$ can potentially be updated. The embedding $\boldsymbol{\alpha}_m^{(t)}$ is initialized every round locally and we do not keep it stateful.

Bases collapse. Unfortunately, such naive training can hardly achieve better performance than using a single basis. To understand, we investigate the federated training dynamics using a preliminary experiment on the PACS image classification dataset (Li et al., 2017) (ResNet18, $K = 4$ bases, $M = 40$, local epochs $= 5$). Specifically, we check a) the average pairwise cosine similarity between the basis model parameters; b) the entropy of the learned combination vectors. High entropy implies an almost uniform combination vector. In Figure 2, we found that both the pairwise similarity and the entropy increase along with local training iterations and along with training rounds. In other words, the bases gradually *collapse* to similar parameters, and the combination vectors of all clients nearly collapse to uniform combinations. Consequently, each basis model does not learn specialized knowledge; the whole bases $\mathcal{V}$ basically degrade to a single global model.

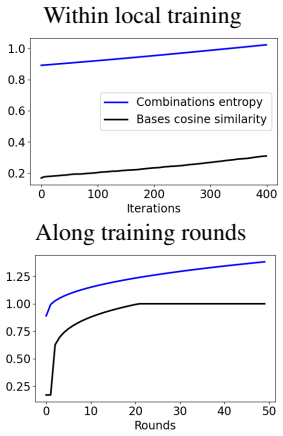

Figure 2: Cosine similarity between bases and the entropy of clients' combination vectors on PACS dataset. FEDBASIS by baseline training collapses to non-specialized bases and uniform combinations: (upper) within the first round of local training for one client; (lower) along training rounds, using basis models aggregated at the server and combinations entropy averaged over clients.

By taking a deeper look at Figure 2, we found that the collapse problem happens primarily in local training. To explain it, let us analyze the gradients derived at local training (cf. Equation 7)

$$\begin{aligned} \nabla_{\boldsymbol{v}_k} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V}) &= \boldsymbol{\alpha}[k] \times \nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta}), \\ \nabla_{\boldsymbol{\alpha}[k]} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V}) &= \boldsymbol{v}_k \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta}). \end{aligned} \tag{8}$$

Interestingly, while with different magnitudes, we found that $\nabla_{\boldsymbol{v}_k} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V})$ pushes every local basis model $\boldsymbol{v}_k \in \tilde{\mathcal{V}}_m^{(t)}$ towards the same direction (since $\boldsymbol{\alpha}[k] \geq 0$). As local basis models gets similar towards $\nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta})$,

their inner products with $\nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta})$ will become larger (i.e., positive) and similar, which would in turn push $\boldsymbol{\alpha}[k]$ to be larger via a similar strength. By forcing $\boldsymbol{\alpha}$ to be on the $(K-1)$-simplex (we do so by reparameterizing $\boldsymbol{\alpha}$ via a softmax function), $\boldsymbol{\alpha}$ will inevitably become uniform. In other words, the more SGD iterations we perform within each round of local training, the more similar the local basis models and the more uniform the combination coefficients will be. We note that this phenomenon does not appear if we aggregate every iteration, which is infeasible in FL setting due to limited communication rounds.

We propose the following treatments to prevent the collapse problem.

Coordinate descent for the combination coefficients and bases. Within each round, we propose to first update $\boldsymbol{\alpha}$ (for multiple SGD steps) while freezing $\mathcal{V}$, and then update $\mathcal{V}$ (for multiple SGD steps) while freezing $\boldsymbol{\alpha}$. We note that at the beginning of each round of local training, $\boldsymbol{v}_k \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta})$ is not necessarily positive. Updating $\boldsymbol{\alpha}$ with frozen $\mathcal{V}$ thus could potentially enlarge the difference among elements in $\boldsymbol{\alpha}$: forcing the personalized model to attend to a subset of bases. After we start to update $\mathcal{V}$, we freeze $\boldsymbol{\alpha}$ to prevent the collapse problem.

Sharpening combination coefficients Since $\boldsymbol{\alpha}[k] \geq 0$, updating $\boldsymbol{v}_k$ locally with $\nabla_{\boldsymbol{v}_k} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V})$ would inevitably increase the cosine similarity between basis models. The exception is when some $\boldsymbol{\alpha}[k] = 0$, which results in 0 gradients. We therefore propose to *artificially* and *temporally* enforce this while calculating $\nabla_{\boldsymbol{v}_k} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V})$. We implement $\boldsymbol{\alpha}$ by learning $\boldsymbol{\psi} \in \mathbb{R}^K$ and reparameterizing it via a softmax function sharpened with a temperature $1 \geq \tau \geq 0$ as $\boldsymbol{\alpha}[k] = \frac{\exp(\boldsymbol{\psi}[k]/\tau)}{\sum_{k'} \exp(\boldsymbol{\psi}[k']/\tau)}$.

Improved training algorithm. Putting these treatments together, we present an improved training algorithm for FEDBASIS based on Equation 7. See the supplementary for the pseudo codes.

$$
\begin{aligned}
\text{Local:} \quad & \text{initialize } \{\boldsymbol{\alpha}, \mathcal{V}\} \text{ by } \{\frac{\mathbf{1}}{\boldsymbol{K}} \times K, \bar{\mathcal{V}}^{(t-1)}\}, && \text{[Step 1]} \\
& \boldsymbol{\alpha}_m^{(t)} = \arg\min_{\boldsymbol{\alpha}} \mathcal{L}_m(\boldsymbol{\alpha}, \mathcal{V}), && \text{[Step 2]} \\
& \boldsymbol{\alpha}_m^{(t)\dagger} \leftarrow \textsc{Sharpen}(\boldsymbol{\alpha}_m^{(t)}; \tau), && \text{[Step 3]} \\
& \tilde{\mathcal{V}}_m^{(t)} = \arg\min_{\mathcal{V}} \mathcal{L}_m(\boldsymbol{\alpha}_m^{(t)\dagger}, \mathcal{V}), && \text{[Step 4]} \\
\text{Global:} \quad & \bar{\mathcal{V}}^{(t)} \leftarrow \frac{1}{M} \sum_{m=1}^{M} \tilde{\mathcal{V}}_m^{(t)}. && (9)
\end{aligned}
$$

4.3 THEORETICAL MOTIVATION

The benefits of such formulation that decouples the overall parameter complexity from the number of clients have been theoretically studied in the Theorem 1 in Shamsian et al. (2021), where the authors learn a linear hypernetwork of size Q to reconstruct every local model's parameters given the corresponding embedding. For brevity, the minor detailed assumptions are listed in section 4.5 in Shamsian et al. (2021). Let K be the embedding size, M be the number of clients, and L be the sum of the Lipschitz constants of the hypernetwork $\mathcal{V}$ and the embeddings $\mathcal{A}$. There exist

$$
N = \mathcal{O}(\frac{K}{\epsilon^2} \log \frac{L}{\delta} + \frac{Q}{M\epsilon^2} \log \frac{L}{\delta}) \tag{10}
$$

such that if the number of samples per client $|\mathcal{D}_m| > N$, we have with probability at least $1-\delta$ for all $\boldsymbol{\theta}_m(\boldsymbol{\alpha}_m, \mathcal{V})$ that the generalization gap between the true and empirical risk $|\tilde{\mathcal{L}}_m(\boldsymbol{\alpha}_m, \mathcal{V}) - \mathcal{L}_m(\boldsymbol{\alpha}_m, \mathcal{V})| \leq \epsilon$. The second term implies that summarizing many clients with a hypernetwork can notably improve generalization. The first term depends on $K (\ll Q)$.

Our formulation of Equation 4 is indeed a linear hypernetwork over $\{\boldsymbol{\theta}_m\}$ thus follows Equation 10. Our FEDBASIS has several advantages compared to the fully-connected network implementation of the hypernetwork in Shamsian et al. (2021). In their experiments, to handle 10 to 100 clients, the hypernetwork size is notoriously large, $Q = 100|\boldsymbol{\theta}|$, making it hard to scale to deeper modern networks. While for our FEDBASIS formulation, $Q = K|\boldsymbol{\theta}|$ with a small K (4 to 8 in our experiments), suggesting we can achieve a better bound. Moreover, reconstruction of the parameters is disconnected from the test loss; we directly learn both the embedding and the bases in local training.

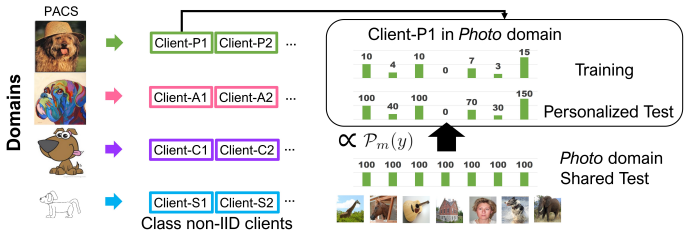

Figure 3: Our proposed construction of federated personalized datasets in PFLBED.

4.4 PRACTICAL EXTENSION

Layer-wise combinations. So far, we apply the same coefficient $\boldsymbol{\alpha}_m[k]$ to combine the whole $\boldsymbol{v}_k$ into $\boldsymbol{\theta}_m$ (cf. Equation 4). Such a formula can be slightly relaxed to decouple coefficients for different layers. For instance, in our experiments on ResNets, we use one vector for each of the 4 blocks and the classifier (instead of one vector for the whole network) for combinations.

The major basis and warm-start for the bases. One concern is that an individual basis does not learn the general knowledge since each basis is likely updated by partial clients but not trained on all data. We show this can be resolved easily by introducing two tricks. We found them generally make the learning smoother thus adopting them by default.

First, we maintain a *major basis* that is always included in the combinations. For instance, Equation 4 becomes $\boldsymbol{\theta}_m(\boldsymbol{\alpha}_m, \mathcal{V}) = \frac{1}{2}\boldsymbol{v}' + \frac{1}{2}\sum_k \boldsymbol{\alpha}_m[k] \times \boldsymbol{v}_k$, where $\boldsymbol{v}'$ is the major basis similar to the global model in FEDAVG and other bases can personalize on top of it.

Second, we see FEDBASIS as a way to summarize many clients' personalized/local models for new clients. It is suitable to serve as a post-processing tool for a conventional FL algorithm. We propose a simple way here. Practically, one can collect $\{\boldsymbol{\theta}_m\}$, cluster them into K clusters, and initialize the K basis model with the centroids. It warm-starts FEDBASIS since each basis already learns general knowledge and is somehow specialized. In our experiments, we first run FEDAVG for a few rounds and collect its global/local models (Chen & Chao, 2022) to warm-start the major/non-major bases, respectively. FEDBASIS can essentially become an extension on a generic FL method like FEDAVG.

5 PFLBED: *bases* FOR BUILDING PERSONALIZED BENCHMARKS

There have been many efforts on building datasets (Caldas et al., 2018; Hsu et al., 2020; Reddi et al., 2021) for *generic* FL. For PFL, how should a dataset be constructed into a reliable evaluation protocol for PFL algorithm development? As side contributions, we propose the following aspects:

Cross-domain with non-IID $\mathcal{P}_m(\boldsymbol{x}, y)$. A realistic personalized dataset should have the joint distribution $\mathcal{P}_m(\boldsymbol{x}, y)$ differ from client to client, not just $\mathcal{P}_m(\boldsymbol{x})$ (e.g., domains) or $\mathcal{P}_m(y)$ (i.e., class labels). Both the training data sizes and the class distributions should be skewed among clients.

Sufficient test samples and matched training/test splits. The test set should be large enough for reliable evaluations. This is challenging when there are many clients, each with a small data size. For example, the popular 62-class hand-written character FEMNIST dataset (Caldas et al., 2018) only has 226 images for each writer on average; many classes only have ≤ 1 image. It is unfaithful to split each client into train/test sets due to mismatches on $\mathcal{P}_m(y)$. Indeed, we found a large discrepancy $\frac{1}{M}\sum_m \|\mathcal{P}_m^{train}(y) - \mathcal{P}_m^{test}(y)\|_1 = 0.77$ even with a $50\%/50\%$ split.

To achieve these desired properties, we propose to transform a cross-domain dataset $\mathcal{D}$ into clients' sets $\{(\mathcal{D}_m^{train}, \mathcal{D}_m^{test/val})\}$ with the following procedures, illustrated in Figure 3.

1. Separate $\mathcal{D}$ based on its domain annotations.
2. For each domain, first split a class-balanced test/validation set which will be shared with all clients from this domain. Take the rest as the training set.
3. For each domain, create a heterogeneous partition (Hsu et al., 2019) for M' clients. An M'-dimensional vector $\boldsymbol{q}_c$ is drawn from a Dirichlet distribution for class c, and we assign the training set of class c to client m' proportionally to $\boldsymbol{q}_c[m']$. Each client's images are from a single domain.
4. Record the class distributions $\mathcal{P}_m(y)$ of each client's training set.

5. For each client in each domain, assign the whole test set of this domain as $\mathcal{D}_m^{test}$. Compute $\frac{1}{M}\sum_m \frac{\sum_i \mathcal{P}_m(y_i)\mathbf{1}(y_i=\hat{y}_i)}{\sum_i \mathcal{P}_m(y_i)}$ as the client-wise average personalized accuracy.

Evaluation on new clients with few-shot samples. As mentioned before as our focus, we consider personalizing for a new client rapidly. We thus split the clients into a participated group and an unparticipated group. After the model is trained, it is personalized with the new client's training data (which is supposed to be a small size in cross-device setup), and follows the same testing protocol.

Examples. In this paper, we consider two image object recognition datasets widely used in domain adaptation tasks. Both provides 4 *handcrafted* domain annotations including PACS (Li et al., 2017) and Office-Home (Venkateswara et al., 2017). For both datasets, following the proposed procedures, we first split the samples of each domain into 60%/20%/5%/15% for participated/unparticipated/validation/test sets. The participated/unparticipated sets are further split into $20/10$ clients per domain by class non-IID sampling from Dirichlet(0.3) (Hsu et al., 2019).

Natural domains. We further include a more realistic Google Landmark (GLD-v2) (Weyand et al., 2020) classification dataset is split into 233 clients based on photographers in the GLD-User23k version (Hsu et al., 2020), thus $\mathcal{P}_m(x, y)$ is naturally non-IID. We provide visualizations in the suppl.

6 EXPERIMENTS

Setups. We use a ResNet-18 (He et al., 2016). Following (Hsu et al., 2020), it is trained from an ImageNet pre-trained model with standard ImageNet-style data pre-processing and 224 resolutions. We use the SGD optimizer; momentum $= 0.9$, weight decay $= 1\text{e}-4$, and local learning rate $= 0.01$. For the two handcrafted/GLD datasets, we run for $100/200$ rounds with $16/64$ batch sizes with 5 local epoch for each participated client (sample $100\%/10\%$) in each round. Statistics of the datasets are summarized in Table 1. We evaluate on **new clients**. Different training sizes (**S**mall/**M**oderate/**L**arge) are used for personalization are considered for each client. See more details in suppl.

Table 1: Summary of datasets and setups.

Dataset	Size	#Class	Domain	Split	#Part./New Clients
PACS	9K	7	Handcrafted	PFLBED	80/40
Office-Home	15.5K	65	Handcrafted	PFLBED	80/40
GLD23K	23K	203	Natural	User ID	117/116

Baselines. Besides FEDAVG (McMahan et al., 2017), we compare to state-of-the-art methods including KNN-PER (Marfoq et al., 2022) and FEDREP (Collins et al., 2021) that are based on general features. FEDBN (Li et al., 2021b) keeps local batchnorm parameters in training and we learn the batchnorm parameters for each new client. PFEDHN (Shamsian et al., 2021) based on hypernetworks and PER-FEDAVG (Fallah et al., 2020) based on MAML are the relevant approaches to ours. As pointed out by Yu et al. (2020); Wang et al. (2019); Chen & Chao (2022); Cheng et al. (2021), fine-tuning (**FT**) serves as a very strong baseline. We also consider linear probe (**LP**) that trains a linear classifier only for all applicable methods.

FEDBASIS. We train FEDBASIS with 5 local epochs for both $\boldsymbol{\alpha}$ and $\mathcal{V}$ as coordinate descent described in subsection 4.2. We warm-start with FEDAVG for 30% of the total rounds and finish the rest, as described in subsection 4.4. We set the temperature $\tau = 0.1$ for sharpening the combinations, and the number of bases is $4/4/8$ besides the major basis for PACS/Office-Home/GLD, respectively. Since the clients are class non-IID, we learn the combinations and the whole classifier for personalization.

Main studies: new clients with low data. Table 2 summarize the results for personalization on class non-IID new clients of the three cross-domain datasets. We observe fine-tuning perform generally stronger than linear probe, but less robust (larger gaps between the last and best epoch). Since the data have large domain discrepancies, the ideal features are likely domain-specific. We found with fine-tuning, FEDAVG is competitive against recent PFL methods like FEDREP and KNN-PER. Interestingly, we observe nearest-neighbor KNN-PER seem to be less effective in such low-data regimes, consistent with Marfoq et al. (2022)[1]. We found PFEDHN is hard to achieve better performance than FEDAVG+FT, as the hypernetwork for a ResNet requires a large number of parameters thus less generalized (supporting subsection 4.3). The most effective baseline in such few-shot scenario is PER-FEDAVG+FT, acknowledging that modeling few-shot personalization from a meta view is a promising direction. Our FEDBASIS conceptually also maintains a meta model over clients by learning to combine basis models, outperforming the baselines especially on harder

[1]We were able to reproduce the results in the original paper and did observe better performance than FEDAVG when each client has more samples (e.g., thousands).

Table 2: Averaged personalized test accuracy (%) on class non-IID new clients sampled from Dirichlet(0.3). Each method is learned on each client's training data of different sizes for 20 epochs with learning rate selected from $\{0.005, 0.01, 0.05\}$. We report both the **Last** epoch and the **Best** by validation.

Method/Dataset	PACS						Office-Home						GLD23k								
Training Size	**S**			**M**			**S**			**M**			**S**			**M**			**L**		
Epoch	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$	Last	Best	$\|\Delta\|$
FedRep+LP	87.4	87.4	0.0	92.5	92.4	0.1	75.6	75.6	0.0	76.0	76.1	0.1	75.7	77.6	1.9	78.8	78.8	0.0	80.1	80.8	0.7
FedRep+FT	89.8	89.8	0.0	92.4	92.5	0.1	74.2	76.1	1.9	75.2	76.4	1.2	79.2	79.9	0.7	81.5	82.5	1.0	81.5	83.5	2.0
FedBN+LP	86.2	88.2	2.0	92.4	92.4	0.0	76.9	77.0	0.1	78.1	78.1	0.0	74.1	74.5	0.4	76.6	76.6	0.0	76.4	76.5	0.1
FedBN+FT	90.8	92.1	1.3	93.0	93.1	0.1	82.3	82.5	0.2	79.0	79.2	0.2	68.1	70.5	2.4	77.8	81.8	4.0	80.5	83.9	3.4
pFedHN	85.4		-	85.5		-	74.1		-	74.3		-	74.5		-	75.6		-	77.2		-
pFedHN+LP	90.4	90.4	0.0	90.6	90.6	0.0	75.1	75.1	0.0	77.4	77.4	0.0	77.0	77.6	0.6	78.5	78.5	0.0	79.1	79.5	0.4
pFedHN+FT	90.5	91.2	0.7	90.4	91.4	1.0	76.2	77.2	1.0	77.1	77.6	0.5	77.6	81.4	3.8	78.6	81.6	3.0	80.2	82.2	2.0
Per-FedAvg+FT	**95.4**	**95.6**	0.2	**96.2**	**96.3**	0.1	84.3	84.4	0.1	86.1	86.2	0.1	78.5	85.3	6.8	79.9	85.2	5.3	82.2	86.1	3.9
kNN-Per	71.6		-	71.6		-	50.4		-	54.5		-	54.0		-	57.4		-	69.2		-
kNN-Per+FT	72.7	72.7	0.0	79.4	79.7	0.3	51.6	52.4	0.8	54.2	54.4	0.2	54.2	54.5	0.3	57.1	57.8	0.7	69.5	70.2	0.7
FedAvg	88.1		-	88.1		-	73.1		-	73.1		-	45.4		-	45.4		-	45.4		-
FedAvg+LP	88.2	90.1	1.9	90.5	90.5	0.0	76.6	76.6	0.0	77.0	77.0	0.0	80.8	81.5	0.7	80.9	81.8	0.9	83.3	83.3	0.0
FedAvg+FT	86.1	91.9	5.8	90.5	90.5	0.0	76.1	77.4	1.3	78.2	78.5	0.3	81.5	84.2	2.7	81.6	84.5	2.9	84.1	86.1	2.0
FedBasis	95.2	95.2	0.0	**96.2**	96.2	0.0	**87.4**	**87.5**	0.1	**87.5**	**87.7**	0.2	**87.4**	**87.4**	0.0	**87.6**	**87.6**	0.0	**89.0**	**89.1**	0.1

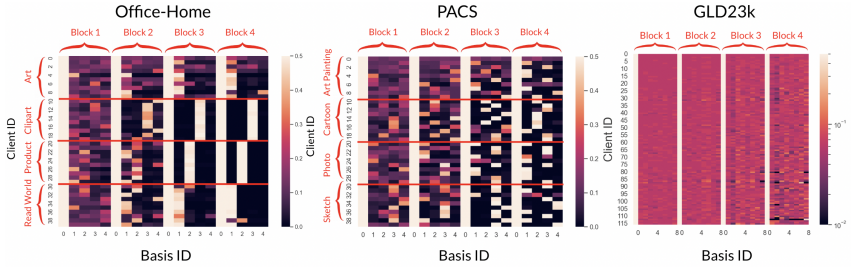

Figure 4: Visualization of the learned combinations $\{\boldsymbol{\alpha}_m\}$. Note that basis 0 is the major basis and each ResNet block shares a combination vector (e.g., $(4+1$ major) bases $\times 4$ blocks for PACS).

datasets Office-Home and GLD. Notably, we learn much fewer parameters per client but still remain competitive to fine-tuning the whole model and more robust to the choice of epochs.

Visualization. To understand what FedBasis learns, we visualize the learned combinations in Figure 4. The clients are cross-domains and class non-IID. Interestingly, we see the clients group according to domains (see Office-Home Block3&4).

Ablations. We provide ablation study (training size **M**) in Table 3, verifying our designs in subsection 4.2 and subsection 4.4.

Robustness of personalization. FedBasis can personalize the features with only a few parameters of the combinations. Comparing to fine-tuning in Table 4 (Office-Home (**M**)), we observe it is much less sensitive to learning rates and training epochs. We note that selecting the *best* epoch may not be always feasible in practice since the clients may not have enough data for validation, thus we believe it is important to consider such robustness, especially in few-shot personalization.

Table 3: Ablation studies of FedBasis: **c**oordinate **d**escent, **m**ajor **b**asis, and temperature τ.

CD	MB	τ	Office	GLD
✗	✓	0.1	83.5	85.8
✓	✗	0.1	87.2	83.3
✓	✓	1.0	87.1	85.5
✓	✓	0.1	87.5	87.6

Table 4: FedBasis is more robust to personalization learning rates.

Method / Learning rates	Last/Best Acc. 0.005	0.05
FedAvg+FT	78.2/78.5	65.4/75.6
Per-FedAvg+FT	86.1/86.2	60.5/83.4
FedBasis	87.5/87.7	87.6/87.6

7 Conclusion

We study personalized federated learning (PFL) for new clients. We aim to bypass the parameter complexity in maintaining personalized models and overcome their vulnerability to hyperparamters when personalized with few training data. We propose a novel PFL architecture and algorithm FedBasis, which constructs each personalized model by a few, shareable basis models. Our training algorithm is designed systematically and mathematically soundly to overcome the difficulty of optimization. We also present a carefully designed evaluation PFLBed. Our empirical studies demonstrate the effectiveness of FedBasis, opening up a new direction for further PFL research.

REFERENCES

Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. *Advances in Neural Information Processing Systems*, 34:8392–8406, 2021. 3

Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *arXiv preprint arXiv:2003.12880*, 2020. 3

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 3, 21

Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *CoRR*, abs/2004.10340, 2020. 16

Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin. Federated user representation learning. *arXiv preprint arXiv:1909.12535*, 2019. 3

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018. 2, 7

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 2, 4

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and X. He. Federated meta-learning with fast convergence and efficient communication. *arXiv: Learning*, 2018. 3

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 7, 8

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 14

Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 2021. 8

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, 2021. 3, 8, 21

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pp. 3213–3223, 2016. 16

Luca Corinzia and Joachim M Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019. 3

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 3, 21

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *NeurIPS*, 2020. 3, 4

An Evgeniou and Massimiliano Pontil. Multi-task feature learning. In *NeurIPS*, 2007. 2, 3, 4, 5

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *KDD*, 2004. 3

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. In *NeurIPS*, 2020. 1, 3, 8, 19, 21

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. 1

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017. 3

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020. 3, 4, 21

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. In *NeurIPS*, 2020. 3, 4

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 7, 8

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision*, pp. 76–92. Springer, 2020. 7, 8, 16

Yutao Huang, Lingyang Chu, Z. Zhou, Lanjun Wang, J. Liu, Jian Pei, and Yanxin Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI*, 2021. 3

Laurent Jacob, Francis Bach, and Jean-Philippe Vert. Clustered multi-task learning: A convex formulation. In *NeurIPS*, 2009. 3

Yihan Jiang, Jakub Konecný, Keith Rush, and S. Kannan. Improving federated learning personalization via model agnostic meta learning. *ArXiv*, abs/1909.12488, 2019. 3

Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. 1

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 1

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 2020. 21

M. Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. In *NeurIPS*, 2019. 3

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021. 16

V. Kulkarni, Milind Kulkarni, and A. Pant. Survey of personalization techniques for federated learning. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 794–797, 2020. 1

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017. 2, 5, 8

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 3, 4

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through. *arXiv preprint arXiv:2012.04221*, 2020a. 1, 2, 3, 4

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020b. 21

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021a. 21

Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fed{bn}: Federated learning on non-{iid} features via local batch normalization. In *ICLR*, 2021b. 3, 8

Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 3, 21

Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10092–10101, 2022. 3

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. 1

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34, 2021. 3

Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pp. 15070–15092. PMLR, 2022. 3, 8

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. 2, 4, 8, 21

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016. 1

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019. 16

Daniel Peterson, Pallika Kanani, and Virendra J Marathe. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733*, 2019. 3

Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pp. 17716–17758. PMLR, 2022. 1

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *ICLR*, 2021. 7

Matthias Reisser, Christos Louizos, Efstratios Gavves, and Max Welling. Federated mixture of experts. *arXiv preprint arXiv:2107.06724*, 2021. 5

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

Ognjen Rudovic, Jaeryoung Lee, Miles Dai, Björn Schuller, and Rosalind W Picard. Personalized machine learning for robot perception of affect and engagement in autism therapy. *Science Robotics*, 3(19):eaao6760, 2018. 1

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *ICML*, 2021. 3, 6, 8

Yiqing Shen, Yuyin Zhou, and Lequan Yu. Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10041–10050, 2022. 3

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *NeurIPS*, 2017. 1, 3, 4, 21

Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34:23309–23320, 2021. 2, 3

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017. 2, 8

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Franccoise Beaufays, and D. Ramage. Federated evaluation of on-device personalization. *ArXiv*, abs/1910.10252, 2019. 8

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2575–2584, 2020. 8

Shanshan Wu, Tian Li, Zachary Charles, Yu Xiao, Ziyu Liu, Zheng Xu, and Virginia Smith. Motley: Benchmarking heterogeneity and personalization in federated learning. *arXiv preprint arXiv:2206.09262*, 2022. 1

Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 14

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020. 1, 8

Edvin Listo Zec, Olof Mogren, John Martinsson, Leon René Sütfeld, and Daniel Gillblad. Federated learning using a mixture of experts. *arXiv preprint arXiv:2010.02056*, 2020. 3

Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34, 2021a. 3

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *ICLR*, 2021b. 3

Mingda Zhang, Chun-Te Chu, Andrey Zhmoginov, Andrew Howard, Brendan Jou, Yukun Zhu, Li Zhang, Rebecca Hwa, and Adriana Kovashka. Basisnet: Two-stage model synthesis for efficient inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3081–3090, 2021c. 3, 14

Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017. 3

Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. 2010. 3

Supplementary Materials

We provide details omitted in the main paper.

- Appendix A: pseudo codes and more discussion of FEDBASIS (cf. section 4 of the main paper).
- Appendix B: additional experimental details, results and analyses (cf. section 4 and section 6 of the main paper).
- Appendix C: additional discussion on the datasets (cf. section 5 of the main paper).
- Appendix D: additional results during the rebuttal.

A FEDBASIS ALGORITHM

Algorithm 1: FEDBASIS— federated training

Server input : initial global basis parameter $\bar{\mathcal{V}}$;
Client m's input : initial local loss $\mathcal{L}_m$, temperature τ;
1 **for** $t \leftarrow 1$ **to** T *rounds* **do**
2 **Communicate** $\bar{\mathcal{V}}$ to all clients $m \in [M]$;
3 **for** *each client* $m \in [M]$ *in parallel* **do**
4 **Initialize** $\{\boldsymbol{\alpha}_m, \mathcal{V}\}$ by $\{\frac{\mathbf{1}}{K} \times K, \bar{\mathcal{V}}\}$;
5 $\boldsymbol{\alpha}_m^\star = \arg\min_{\boldsymbol{\alpha}_m} \mathcal{L}_m(\boldsymbol{\alpha}_m, \mathcal{V})$
6 $\boldsymbol{\alpha}_m^{\star\dagger} \leftarrow \text{SHARPEN}(\boldsymbol{\alpha}_m^\star; \tau)$;
7 $\mathcal{V}_m^\star = \arg\min_{\mathcal{V}} \mathcal{L}_m(\boldsymbol{\alpha}_m^{\star\dagger}, \mathcal{V})$;
8 **Communicate** $\mathcal{V}_m^\star$ to the server;
9 **end**
10 **Construct** $\bar{\mathcal{V}} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{V}_m^\star$;
11 **end**
Server output : $\bar{\mathcal{V}}$;

Algorithm 2: FEDBASIS— generate a personalized model

Client m's input : initial global basis parameter $\mathcal{V}$, local loss $\mathcal{L}_m$;
Initialize $\boldsymbol{\alpha}_m$ by $\frac{\mathbf{1}}{K} \times K$;
$\boldsymbol{\alpha}_m^\star = \arg\min_{\boldsymbol{\alpha}_m} \mathcal{L}_m(\boldsymbol{\alpha}_m, \mathcal{V})$
Construct $\boldsymbol{\theta}_m(\boldsymbol{\alpha}_m, \mathcal{V}) = \sum_k \boldsymbol{\alpha}_m[k] \times \boldsymbol{v}_k$;
Client m's output : $\boldsymbol{\theta}_m$;

We provide a summary in algorithm 1 for training our FEDBASIS (cf. subsection 4.2 in the main paper) and algorithm 2 shows how use it for generating a personalized model. Similar to the FEDAVG algorithm, our FEDBASIS also executes a multi-round training procedure between the local training at the clients and aggregation at the server.

The goal of FEDBASIS is to collaboratively train K basis models $\mathcal{V} = \{\boldsymbol{v}_k\}_{k=1}^K$ which can be used to combine into personalized models based on each client's combination coefficient $\boldsymbol{\alpha}_m \in \mathbb{R}^K$ (or more specifically, $\Delta^{(K-1)}$; see Equation 4) within limited T rounds of communications. The parameters are linearly combined layer by layer. Such specialized layers (Yang et al., 2019; Chen et al., 2020; Zhang et al., 2021c) improve the performance with little extra inference cost. Our contribution is to extend such concepts to personalization in FL setting, identify optimization issues, and resolve them.

To effectively learn the bases for personalization, in subsection 4.2, we introduce several important techniques in the local training to avoid bases collapse and encourage each basis to learn specialized knowledge. In each round of local training at a client m, it first initializes the bases $\mathcal{V}$ using the one broadcast by the server. Next, we train $\boldsymbol{\alpha}_m$ and $\mathcal{V}$ with coordinate descent. We update $\boldsymbol{\alpha}_m$ (for multiple SGD steps) while freezing $\mathcal{V}$ (line 5 in algorithm 1). To force the personalized model to attend to a subset of bases, we sharpen $\boldsymbol{\alpha}_m$ by injecting a temperature into the Softmax function (line 6 in algorithm 1). Then, we update $\mathcal{V}$ (for multiple SGD steps) while freezing $\boldsymbol{\alpha}_m$. Finally, the updated bases are sent back to the server for a basis-wise average with other clients' updates.

The FEDBASIS formulation enjoys several desired properties.

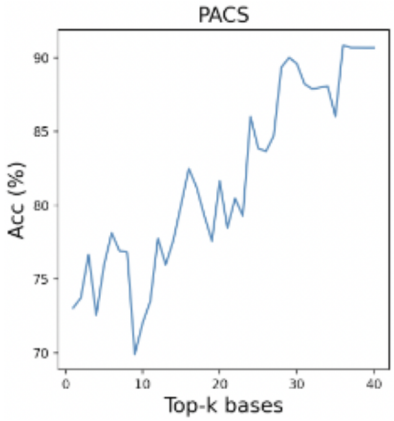

Figure 5: Reducing 40 fine-tuned personalized models into top-k bases by PCA.

- The total parameter size of models does not scale with the number of clients. FEDBASIS ultimately outputs the bases $\mathcal{V}$ with combination coefficients $\boldsymbol{\alpha}_m$ for each client m. Each client only has $|\boldsymbol{\alpha}_m| = K$ personalized parameters, which is negligible compared to the model.
- For local training, the forwarding combined model only needs to be generated per mini-batch but not per instance, making it scalable to batch sizes. The size of communications is K times more but K is typically small.
- FEDBASIS does not increase clients' computation cost in inference. After training, the basis models are combined into a single personalized model. This is sharply different from the mixture of experts in that input needs to go through every expert and ensembles the predictions, where the cost is linear to the number of experts.

B MORE DETAILS AND RESULTS

B.1 SPLIT NEW CLIENTS FOR EVALUATION

In our experiments, to demonstrate the data efficiency of each of the methods, we consider different training sizes (**S**mall/**M**oderate/**L**arge) for personalization each client. Concretely, for Office-Home and PACS, we use $50\%/100\%$ of each client's training set as the **S/M** setting for personalization, respectively. We note that, in PFLBED, we already split a relatively small set (20% of the overall data) and further split it into several new clients. On the other hand, for the GLD-v2 dataset, the clients are already split by User IDs, we thus randomly split $10\%/20\%/40\%$ of each new client's data as the training set and take the rest as the test/validation sets (we split 20% for validation).

B.2 ANOTHER BASELINE: PRINCIPAL COMPONENT ANALYSIS (PCA)

Our FEDBASIS architecture is to represent personalized models by a set of few basis models. In the main paper, due to the page limit, we mainly present methods that directly learn the basis models. Here, we present another baseline, building upon a reverse way of thinking: *How can we summarize many personalized models into combinations of a few basis models given the federated constraint that no data are available at the server?* A straightforward way to achieve such model compression is to perform Principal Component Analysis (PCA) on the collection of all the personalized model parameters. That is, we can try to represent each personalized model with a few eigenvectors (as $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$) with the top-$k$ eigenvalues found by PCA.

Different from our more challenging experiments in the main paper, we consider an *ideal* case of personalization in centralized setting, we train a global model with mini-batches SGD on PACS datasets. We first train a global model and fine-tune it on each client's full dataset to obtain 40 personalized models $\{\boldsymbol{\theta}_m\}$. Then, we perform PCA on their vectorized parameters.

As shown in Figure 5, we observe the averaged personalized performance drops drastically as the number of eigenvectors decreases. For instance, reducing into 4 bases leads to slumps in the accuracy

of 18.1% for PACS. It demonstrates the challenge of this problem. We hypothesize that the poor performance is likely because (1) personalized models produced by fine-tuning do not simply lie on a small-dimensional space and/or (2) such PCA linear method cannot guarantee to maintain the accuracy since the reconstruction of parameters is not tied to the real loss such as Equation 3, as we can also observe some fluctuations on accuracy along with the changes of top-k.

Alternately, we instead investigate using k-means clustering on the personalized models $\{\boldsymbol{\theta}_m\}$ parameters to cluster them into $k = 4$ models and use each client's assigned centroid as the personalized models. We again see a significant accuracy drop of 21.4% for PACS.

Therefore, we are motivated to solve our proposed objective Equation 6 that aims to directly learn the bases such that all personalized models can be their linear combinations while minimizing the local empirical risks.

C More discussions on the datasets

C.1 Discussions on PFLBed

In section 5, we provide several aspects including cross-domain and class non-IID $\mathcal{P}_m(\boldsymbol{x}, y)$, sufficient test samples, matched training/test splits, and distributional robustness evaluated with the class-balanced accuracy. We propose a standardized process called PFLBed to construct a faithful personalized dataset for PFL algorithm development. As examples, we propose to transform some existing datasets including PACS and Office-Home, that are widely used in bench-marking domain adaption tasks, into PFL datasets. These datasets are suitable for experimental use in research since they are created with clear domain differences such as image styles like *Photo* or *Art*. We also propose to use the existing naturally partitioned dataset GLD-v2, a dataset consisting of landmark photographs taken from various locations around the world by different photographers where each partition contains 1 photographer's photos. For this reason, we can view the style difference amongst the photographers as the domain gap thus treating each partition (a.k.a client) as an independent domain. Because the number of samples contained in each partition also varies, we naturally believe this dataset is a faithful personalized dataset for evaluating PFL algorithms along with PFLBed.

Here we discuss future work of PFLBed to extend to more datasets. We identify some promising datasets. DomainNet (Peng et al., 2019) contains 6 domains of 345 different objects. The WILDs benchmark (Koh et al., 2021) collects several datasets across different applications and each domain is defined by attributes such as users, locations, different cameras (Beery et al., 2020), etc. (Hsu et al., 2020) presents two realistic datasets of species classification and landmark recognition split by locations or users for generic FL but not for PFL. Cityscape (Cordts et al., 2016) is a popular self-driving dataset that contains driving scene data from many cities in Germany. In this paper, we highlight the importance of PFL dataset construction for faithful evaluation and focus on some more experimental datasets. We hope our efforts can inspire future work to propose more datasets suitable for PFL research.

C.2 Visualizations of PFLBed dataset client distribution

Here we show example client distributions of our proposed datasets for PFLBed. For PACS and Office-Home datasets, we follow the procedures outlined in section 5 where each client is sampled from Dirichlet(0.3) within each domain. For each dataset, we first record the occurrences N_{md} of each label $d \in D$ within each client $m \in M$ as $C_m^{1\times D}$ for D total labels and M total clients. We then concatenate all M clients' label counts as $C^{M\times D}$. We visualize the distribution using ${C^{M\times D}}^T = C^{D\times M}$ so that the size of each point is proportional to label count N_{md} for a total of $M \times D$ points and each column $C_m^{D\times 1}$ can be viewed as a single client's label distribution. As we can see, our clients show both label space $\mathcal{P}_m(y)$ and domain space $\mathcal{P}_m(x)$ heterogeneity. It's worth noting that although Figure 6 does not directly show domain differences through color differences, each client can be directly treated as an independent domain for the reason described in subsection C.1.

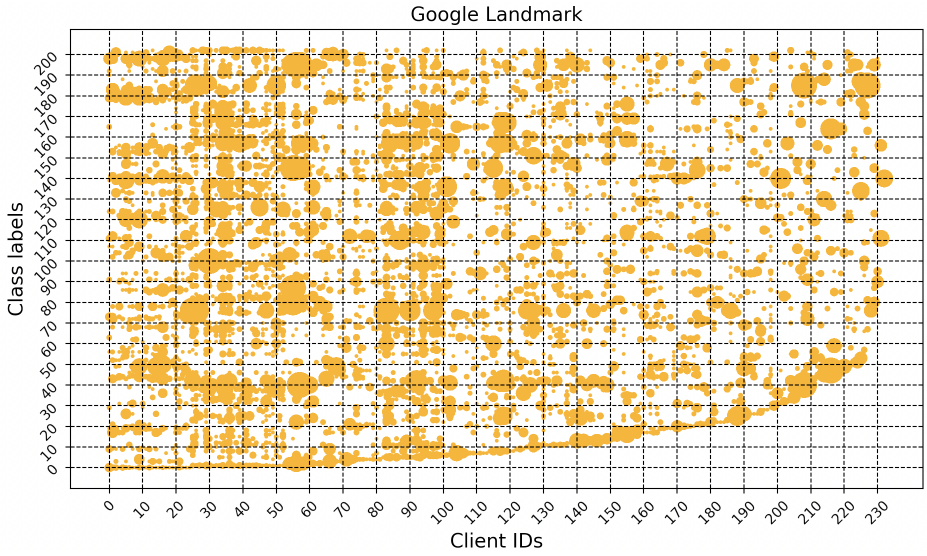

Figure 6: Clients distribution of GLD23k dataset. Each number on the horizontal axis represents a particular client for a total of $M = 233$ clients. Each number on the vertical axis represents a particular class label for a total of $D = 203$ classes. The maximum number of samples per class is 100.

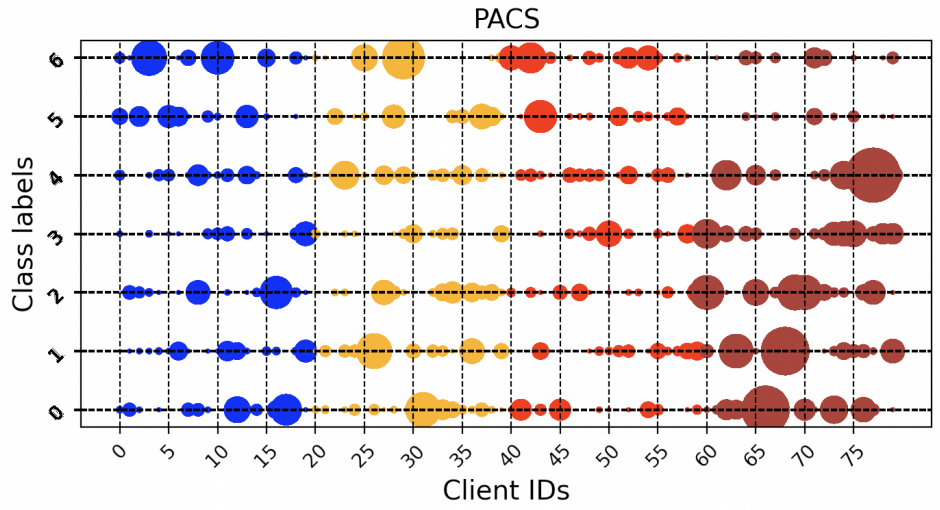

Figure 7: Clients distribution of PACS dataset across 4 different domains. Each number on the horizontal axis represents a particular client for a total of $M = 80$ clients. Each number on the vertical axis represents a particular class label for a total of $D = 7$ classes. The maximum number of samples per class is 256.

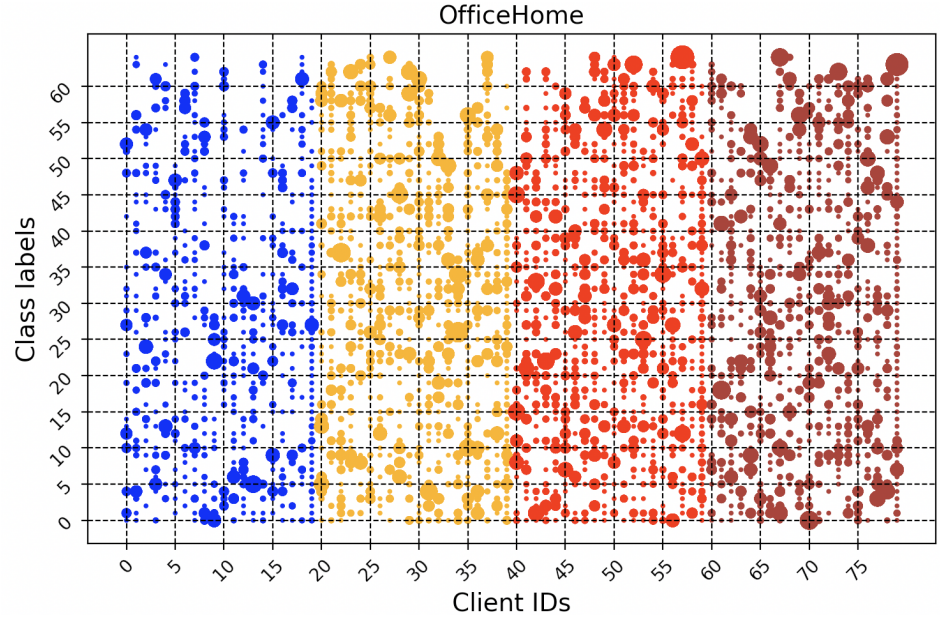

Figure 8: Clients distribution of Office-Home dataset across 4 different domains. Each number on the horizontal axis represents a particular client for a total of $M = 80$ clients. Each number on the vertical axis represents a particular class label for a total of $D = 65$ classes. The maximum number of samples per class is 49.

C.3 ADDITIONAL VISUALIZATIONS OF GLD23K CLIENT DISTRIBUTION

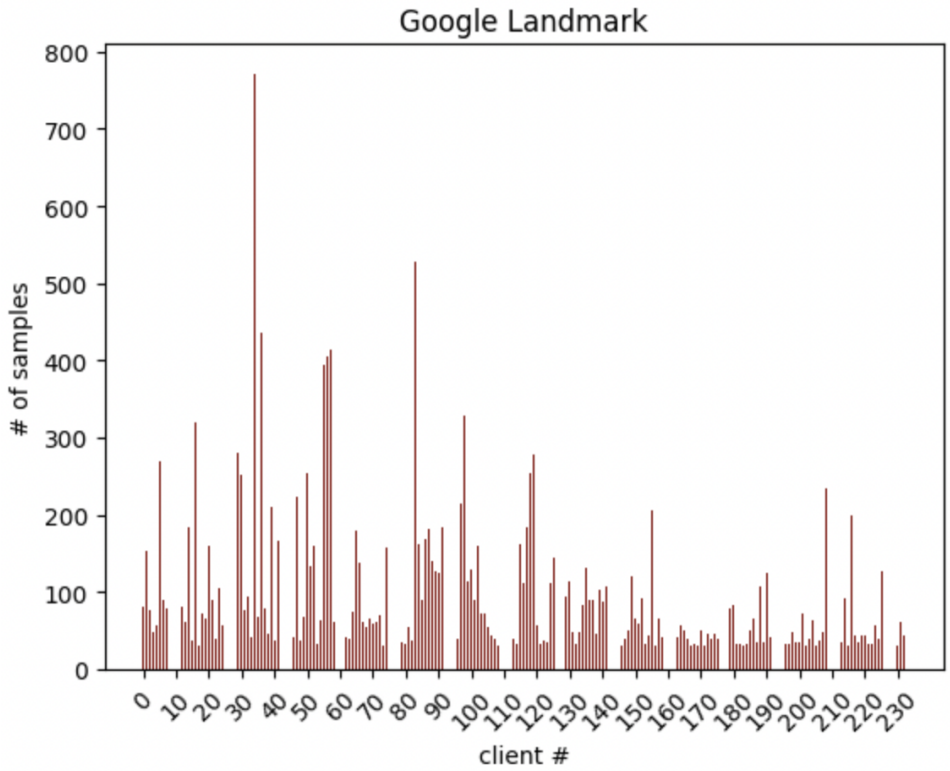

Figure 9: Clients label distribution of GLD23K dataset. Each number on the horizontal axis represents a particular client for a total of $M = 233$ clients. Vertical axis represents the number of samples each client has.

D ADDITIONAL RESULTS DURING REBUTTAL

D.1 MORE RESULTS ABOUT THE ROBUSTNESS OF FEDBASIS

In both Table 2 and Table 4 in the main paper, we demonstrate the robustness of the FEDBASIS on the choices of learning rates and stopping epochs when it is fine-tuned for new clients, compared to other baselines. Note that, in the current Table 2, for each method and each dataset, we highlight the difference ($|\Delta|$) between stopping the fine-tuning by the last epoch or by the best epoch selected by validation.

As requested by **Reviewer 2JAA**, we plotted out the dynamics of training on new clients, featuring more sets of different learning rates along the fine-tuning epochs for both our FEDBASIS and the most competitive baseline PER-FEDAVG+FT in Table 2. We focus on the more challenging datasets Office-Home with the small training size setting. As shown in Figure 10 and Figure 12, it is quite clear that FEDBASIS is much more robust to various learning rates and does not require early stopping. We attribute it to the clear advantage that FEDBASIS only needs to personalize much fewer parameters when adapting to a new client, thus enjoying the robustness. We further note for PER-FEDAVG+FT, although with proper tuning it can achieve decent performance (still lower than ours), this requires a validation set for each client thus likely not practical in the real world.

We focus on the **FO** version of PER-FEDAVG+FT (Fallah et al., 2020) due to its better accuracy and training efficiency on the datasets in our experiments. In Table 5, we provide a comparison on PER-FEDAVG+FT with the two variants FO and HF introduced in (Fallah et al., 2020) and we confirmed FO is better in the performance.

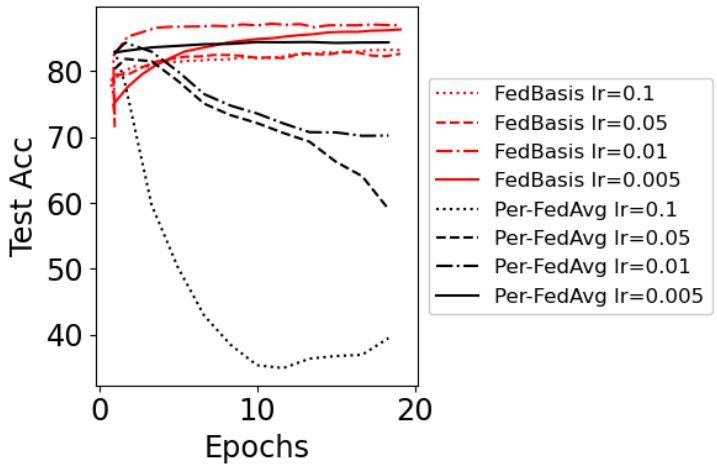

Figure 10: Fine-tuning training curves of Office-Home (small) dataset (cf. Table 2 in the main paper) of PER-FEDAVG+FT and FEDBASIS with various fine-tuning learning rates.

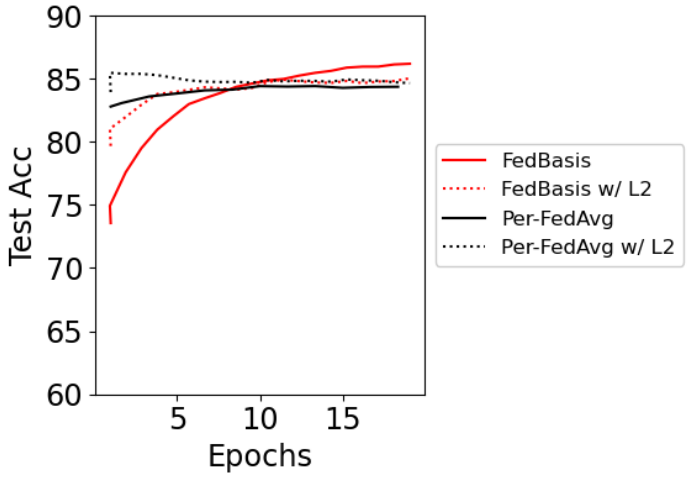

Figure 11: Fine-tuning training curves of Office-Home (small) dataset (cf. Table 2 in the main paper) of PER-FEDAVG+FT and FEDBASIS with fine-tuning learning rate $= 0.005$ with ℓ-2 regularization.

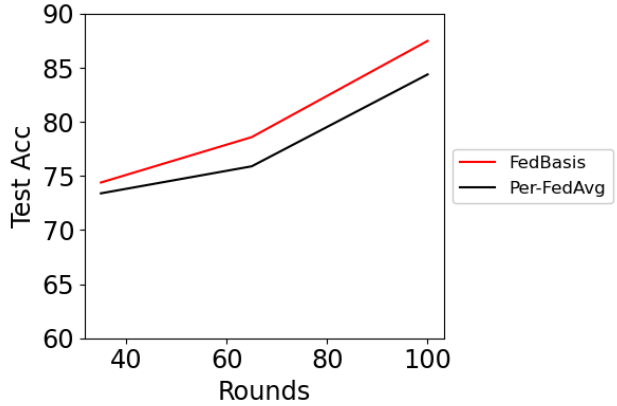

Figure 12: Federated training curves of Office-Home (small) dataset (cf. Table 2 in the main paper) of PER-FEDAVG+FT and FEDBASIS along the rounds. In each evaluated round (note that FEDBASIS warm-starts from 30 rounds of FEDAVG, as described in subsection 4.4), we run the adaption procedure to evaluate on the new clients to report the averaged personalized accuracy as the same as cf. Table 2.

Table 5: More results of PER-FEDAVG+FT with FO and HF variants in Fallah et al. (2020). The results on based on Office-Home (**M**oderate training size) and cf. Table 4 in the main paper.

Method Learning rates	Last/Best Acc. 0.001	 0.005	 0.01	 0.05	 0.1
FEDAVG+FT	78.1/78.1	78.2/78.5	70.5/76.6	65.4/75.6	38.1/73.1
PER-FEDAVG+FT (FO)	86.1/86.1	86.1/86.2	65.3/85.4	60.5/83.4	40.1/82.6
PER-FEDAVG+FT (HF)	85.3/85.3	85.5/85.5	63.4/83.6	43.7/81.6	34.8/80.7
FEDBASIS	87.6/87.6	87.5/87.7	87.6/87.7	87.6/87.6	87.5/87.6

Table 6: Average test accuracies on various partitions of CIFAR10 and CIFAR100 with participation rate $= 0.1$, following the same experiment setting of Table 1 in Collins et al. (2021), where the results of the baselines are copied from there as well. Our results use the authors' codes and their pipeline (Collins et al., 2021) by plugging in our algorithm. We report the mean and variances of the accuracy over 3 runs of different random seeds.

	CIFAR10	CIFAR100
(# clients, # classes per client)	$(100, 5)$	$(100, 5)$
Local Only	70.68	75.29
FedAvg (McMahan et al., 2017)	51.78	23.94
FedAvg+FT	73.68	79.34
FedProx (Li et al., 2020b)	50.99	20.17
FedProx+FT	72.75	78.52
SCAFFOLD (Karimireddy et al., 2020)	47.33	20.32
SCAFFOLD+FT	68.23	78.88
Fed-MTL (Smith et al., 2017)	58.31	71.47
PerFedAvg (Fallah et al., 2020)	67.20	72.05
LG-Fed (Liang et al., 2020)	63.02	72.44
L2GD (Hanzely & Richtárik, 2020)	59.98	72.13
APFL (Deng et al., 2020)	72.29	78.20
Ditto (Li et al., 2021a)	70.34	78.91
FedPer (Arivazhagan et al., 2019)	73.84	76.00
FedRep (Collins et al., 2021)	75.68	79.15
FedBasis (Ours, # basis$= 3$)	75.5 ± 0.71	80.8 ± 0.54

Table 7: Variance (σ^2) of personalized test accuracy (%) over 3 different runs for cf. Table 2.

Method/Dataset	PACS				Office-Home				GLD23k					
Training Size	**S**		**M**		**S**		**M**		**S**		**M**		**L**	
Epoch	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best	Last	Best
FEDREP+LP	0.22	0.16	0.21	0.22	0.31	0.26	0.33	0.25	0.55	0.56	0.47	0.52	0.44	0.29
FEDREP+FT	0.36	0.29	0.41	0.38	0.66	0.56	0.71	0.39	0.78	0.89	0.88	0.75	0.74	0.71
FEDBN+LP	0.15	0.16	0.31	0.15	0.12	0.23	0.28	0.19	0.36	0.41	0.29	0.21	0.51	0.39
FEDBN+FT	0.33	0.45	0.41	0.42	0.67	0.59	0.55	0.62	0.68	0.66	0.56	0.48	0.65	0.62
PFEDHN	0.78	0.64	0.56	0.57	0.46	0.51	0.48	0.55	0.41	0.28	0.55	0.56	0.39	0.28
PFEDHN+LP	0.36	0.44	0.29	0.36	0.27	0.31	0.28	0.25	0.87	0.86	0.82	0.75	0.78	0.58
PFEDHN+FT	0.85	0.97	0.56	0.77	0.77	0.75	0.64	0.70	1.01	1.12	0.89	0.88	0.91	1.15
PER-FEDAVG+FT	0.51	0.46	0.37	0.41	0.70	0.61	0.63	0.66	0.51	0.25	0.48	0.45	0.34	0.38
KNN-PER	0.30	0.34	0.19	0.38	0.56	0.58	0.52	0.57	1.14	1.56	1.28	0.85	0.95	0.95
KNN-PER+FT	1.75	1.41	1.25	1.39	0.57	0.60	0.78	0.69	0.27	0.56	0.48	0.71	0.55	0.61
FEDAVG	0.23	0.25	0.29	0.24	0.38	0.41	0.50	0.42	0.54	0.39	0.55	0.56	0.54	0.27
FEDAVG+LP	0.15	0.21	0.20	0.17	0.29	0.22	0.31	0.44	0.63	0.65	0.48	0.59	0.64	0.59
FEDAVG+FT	0.52	0.39	0.44	0.51	0.57	0.46	0.60	0.58	0.68	0.71	0.59	0.58	0.61	0.48
FEDBASIS	0.52	0.56	0.45	0.50	0.38	0.39	0.42	0.45	0.52	0.66	0.47	0.68	0.29	0.45

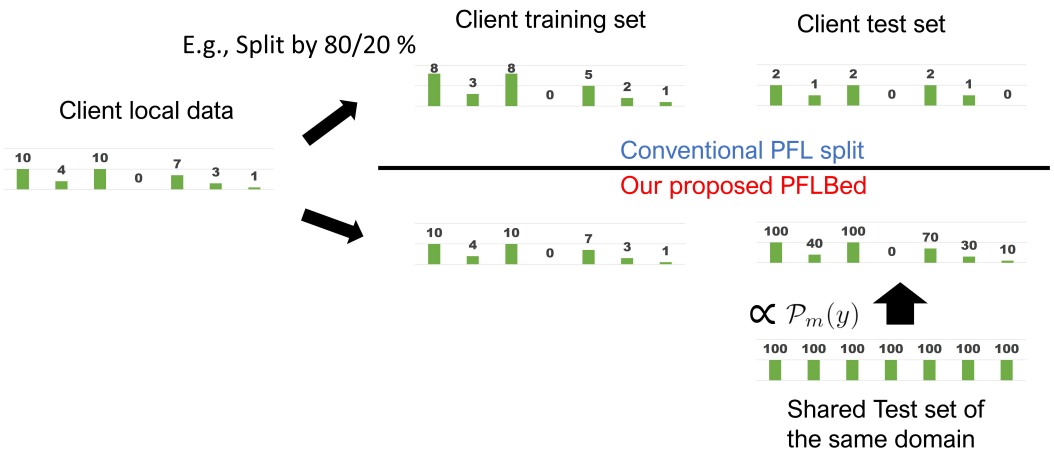

Figure 13: Illustration of the difference between traditional PFL split and our proposed PFLBED in cf. section 5 for a client. For the conventional way, given that each client may have limited data per class, after a training/test split, the distribution is no longer matched, leading to a unfaithful evaluation. On the contrary, Our proposed way use a shared test set from the same domain and re-weight the examples in evaluation by classes (e.g., weighted accuracy).