Personalized Federated Learning via Feature Distribution Adaptation

Connor J. McLaughlin, Lili Su
Northeastern University, Boston, MA 02115
{mclaughlin.co,l.su}@northeastern.edu

Abstract

Federated learning (FL) is a distributed learning framework that leverages commonalities between distributed client datasets to train a global model. Under heterogeneous clients, however, FL can fail to produce stable training results. Personalized federated learning (PFL) seeks to address this by learning individual models tailored to each client. One approach is to decompose model training into shared representation learning and personalized classifier training. Nonetheless, previous works struggle to navigate the bias-variance trade-off in classifier learning, relying solely on limited local datasets or introducing costly techniques to improve generalization. In this work, we frame representation learning as a generative modeling task, where representations are trained with a classifier based on the global feature distribution. We then propose an algorithm, pFedFDA, that efficiently generates personalized models by adapting global generative classifiers to their local feature distributions. Through extensive computer vision benchmarks, we demonstrate that our method can adjust to complex distribution shifts with significant improvements over current state-of-the-art in data-scarce settings. Our source code is available on GitHub[1].

1 Introduction

The success of deep learning models relies heavily on access to large, diverse, and comprehensive training data. However, communication constraints, user privacy concerns, and government regulations on centralized data collection often pose significant challenges to this requirement [31, 37, 18]. To address these issues, Federated Learning (FL) [34] has gained considerable attention as a distributed learning framework, especially for its privacy-preserving properties and efficiency in training deep networks.

The FedAvg algorithm, introduced in the seminal work [34], remains one of the most widely adopted algorithms in FL applications [32, 45, 38, 49, 40, 7]. It utilizes a parameter server to maintain a global model, trained through iterative rounds of distributed client local updates and server aggregation of client models. While effective under independent and identically distributed (i.i.d.) client data, its performance deteriorates as client datasets become more heterogeneous (non-i.i.d.). Data heterogeneity leads to the well-documented phenomenon of client drift [19], where distinct local objectives cause the model to diverge from the global optimum, resulting in slow convergence [20, 28] and suboptimal local client performance [42]. Despite extensive efforts [27, 19, 46, 6] to enhance FedAvg for non-i.i.d. clients, the use of a single global model remains too restrictive for many FL applications.

Personalized federated learning (PFL) has emerged as an alternative framework that produces separate models tailored to each client. The success of personalization techniques depends on

[1] https://github.com/cj-mclaughlin/pFedFDA

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2451

balancing the bias introduced by using global knowledge that may not generalize to individual clients, and the variance inherent in learning from limited local datasets. Popular PFL techniques include regularized local objectives [41, 26], local-global parameter interpolation [6], meta-learning [9, 16], and representation learning [35, 5, 48, 29]. While these techniques have shown significant improvements for clients under limited types of synthetic data heterogeneity (e.g., imbalanced partitioning of an otherwise i.i.d. dataset), we find that current methods still struggle to navigate the bias-variance trade-off with the additional challenge of feature distribution shift and data scarcity, conditions commonly encountered in cross-device FL.

As such, we look to design a method capable of handling real-world distribution shifts, e.g., covariate shift caused by weather conditions or poor camera calibration, (see clients 1 and 2 in Fig. 1) with limited local datasets. To this end, we approach PFL through shared representation learning guided by a global, low-variance generative classifier. Specifically, we select a probability density p with desirable statistical properties (e.g., one that admits an efficient Bayesian classifier) and iteratively estimate the global parameters of this distribution and representation layers to produce features from the estimated distribution (Fig. 1a).

To further navigate the bias-variance trade-off, we introduce a local-global interpolation method to adapt the global estimate to the distribution of each client. At inference time, clients use their adaptive local distribution estimate in a personalized Bayesian classifier (Fig. 1b).

Contributions. We propose a novel **P**ersonalized **Fed**erated Learning method based on **F**eature **D**istribution **A**daptation (**pFedFDA**). We contextualize our algorithm using a class-conditional multivariate Gaussian model of the feature space in a variety of computer vision benchmarks. Our empirical evaluation demonstrates that our proposed method consistently improves average model accuracy in benchmarks with covariate shift or client data scarcity, obtaining over 6% in multiple settings. At the same time, our method remains competitive with current state-of-the-art (often within 1%) on more general benchmarks with more moderate data heterogeneity. To summarize, our contributions are three-fold:

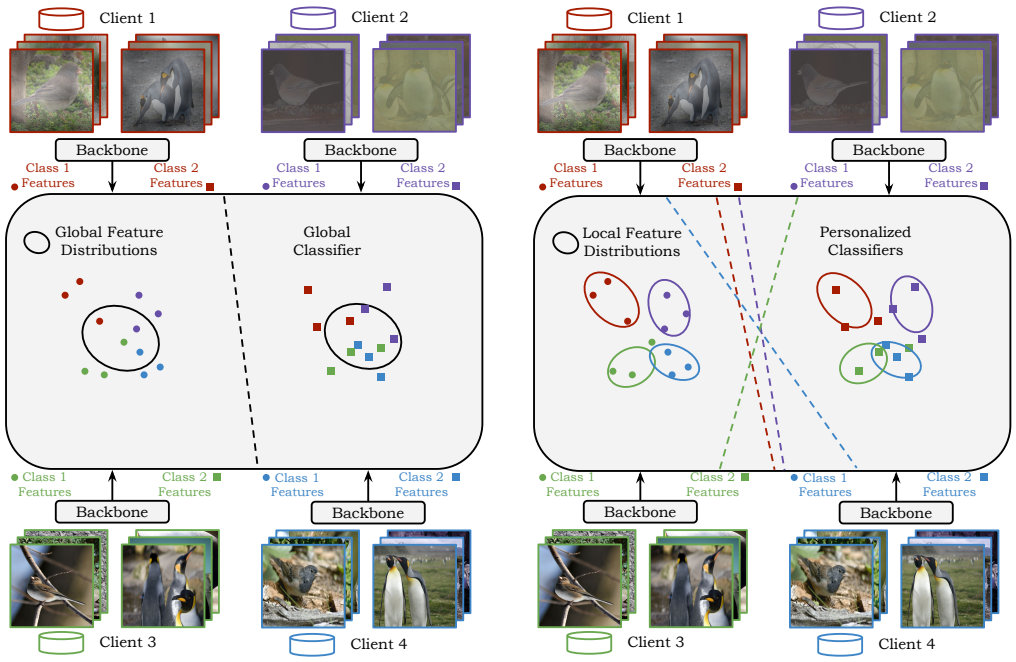
- A novel generative modeling perspective for federated representation learning is proposed to enable a new bias-variance trade-off for client classifier learning.
- We propose a personalized federated learning method, pFedFDA, which leverages awareness of latent data distributions to guide representation learning and client personalization.
- Extensive experiments on image classification datasets with varying levels of natural data heterogeneity and data availability demonstrate the advantages of pFedFDA in challenging settings.

2 Related Work

Federated Learning with Non-i.i.d. Data. Various studies have worked to understand and improve the ability of FL to serve heterogeneous clients. In non-i.i.d. scenarios, the traditional FedAvg method [34] is susceptible to client drift [19], resulting in slow convergence and poor local client accuracy [28, 27]. To tackle this challenge, [27, 1, 21] proposed the use of regularized local objectives to reduce the bias on the global model after local training. Another approach focuses on rectifying the bias of local updates [19, 10] through techniques such as control variates. Other strategies include loss-balancing [15, 47, 3], knowledge distillation [30, 54], prototype learning [43], and contrastive learning [25]. Despite promising results on non-i.i.d. data, their reliance on a single global model poses limitations for highly heterogeneous clients [17].

Personalized Federated Learning. In response to the limitations of a single global model, PFL seeks to overcome heterogeneity by learning models tailored to each client. In this framework, methods attempt to strike a balance between being flexible enough to fit the local distribution and relying on global knowledge to prevent over-fitting on small local datasets. Popular strategies include meta-learning an initialization for client adaptation [16, 9], multi-task learning with local model regularization [41, 26], local and global model interpolation [6], personalized model aggregation [51, 50], client clustering [39, 8], and decoupled representation and classifier learning [5, 29, 35, 48, 3]. Our work focuses on this latter approach, in which the neural network is typically decomposed into the first $L-1$ layers used for feature extraction, and the final classification layer.

Existing works in this category share feature extraction parameters between clients and rely on client classifiers for personalization. These approaches differ primarily in the acquisition of client classifiers



(a) Global Representation Learning (b) Local Distribution Adaptation

Figure 1: Overview of pFedFDA. (Left) Heterogeneous clients collaboratively train representation parameters under a generative classifier derived from a global estimate of class feature distributions. (Right) At test time, clients adapt the generative classifier to their feature distributions to obtain personalized classifiers.

during training, which influences representation learning. For example, FedRep [5] sequentially trains a strong local classifier while holding the representation fixed, then updates the representation under the fixed classifier. FedBABU [35] proposes to use fixed dummy classifiers to align client objectives, only fine-tuning the classifier layers after the representation parameters have converged. Similarly, FedRoD [3] aims to train a generic representation model and classifier in tandem via balanced softmax loss, later obtaining personalized classifiers through fine-tuning or hypernetworks. FedPAC [48] adopts the learning algorithm of FedRep, but additionally regularizes the feature space to be similar across clients, before learning a personalized combination of classifiers across clients to improve generalization. However, this collaboration comes with an additional computational overhead that scales with the number of active clients. pFedGP [2] leverages a shared feature extractor as a kernel for client Gaussian processes. Although this approach offers improved sample efficiency, it comes at the cost of increased computational complexity and reliance on an inducing points set.

In a similar spirit to our method, FedEM [33] estimates the latent data distribution of clients in parallel to training classification models. FedEM estimates each client data distribution as a mixture of latent distributions, where personalized models are a weighted average of mixture-specific models. Notably, this introduces a significant overhead in both communication and computation as separate models are trained for each mixture. In contrast, our work estimates the distribution of client features in parallel to training a global representation model.

3 Problem Formulation

FL System and Objective. We consider an FL system where a parameter server coordinates with M clients to train personalized models θ_i, $i = 1, 2, \cdots, M$. Each client i has a local training dataset $\mathcal{D}_i = \{(x_i^j, y_i^j)\}_{j=1}^{n_i}$, where $x \in \mathbb{R}^m$ and $y \in \{1, \cdots, C\}$. The model training objective in PFL is:

$$\min_{\theta_1,\ldots,\theta_M \in \mathcal{Q}} f(\theta_1, ..., \theta_M) := \frac{1}{M} \sum_{i=1}^{M} F_i(\theta_i), \tag{1}$$

https://doi.org/10.52202/079017-2451

where $\mathcal{Q}$ is feasible set of model parameters, $F_i(\theta_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[L(\theta_i(x), y)]$ is the empirical risk of dataset $\mathcal{D}_i$, and L is a loss function of the prediction errors (e.g., cross-entropy). The client population M in FL can be large, resulting in partial client participation [17]. Let q denote the participation rate, meaning that in each round, a client participates in model training with probability q.

Following [5, 48], we approach this as a problem of global representation learning and local classification, in which each θ_i consists of a shared backbone, ϕ, responsible for extracting low-level features ($z \in \mathbb{R}^d = \phi(x)$), and a local classifier, h_i, for learning a client-specific mapping between features and labels. Considering this decomposition of parameters $\theta_i = (h_i \circ \phi)$, we can rewrite the original PFL objective as the following:

$$\min_{\phi\in\Phi} \frac{1}{M} \sum_{i=1}^{M} \min_{h_i\in\mathcal{H}} F_i(h_i \circ \phi), \tag{2}$$

where Φ and $\mathcal{H}$ are the feasible sets of neural network and classifier parameters, respectively.

In our generative modeling framework, we consider $\mathcal{H}$ to be the probability simplex over $\{1, \cdots, C\}$, and our algorithm uses approximations of the posterior distributions as classifiers h_i. However, for fair comparison with existing work (as well as other nice properties, discussed in Section 4.1), we select a generative model of the feature space such that h_i can be represented with an equivalent linear layer.

Data Heterogeneity. The data distribution of each client i is a joint distribution on $\mathcal{X} \times \mathcal{Y}$, which can be written as $p_i(x, y)$, $p_i(y)p_i(x|y)$, or $p_i(x)p_i(y|x)$. Using the terminology of [17], we refer to each case of data heterogeneity as follows: *prior probability shift* ($p_i(y) \neq p_{i'}(y)$), *concept drift* ($p_i(x|y) \neq p_{i'}(x|y)$), *covariate shift* ($p_i(x) \neq p_{i'}(x)$), and *concept shift* ($p_i(y|x) \neq p_{i'}(y|x)$). Furthermore, the local dataset volumes $\mathcal{D}_i$ may have *quantity skew*, i.e., $n_i \neq n_{i'}$.

4 Methodology

In this section, we introduce pFedFDA, a personalized federated learning method that utilizes a generative modeling approach to guide global representation learning and adapt to local client distributions. We present our method using a class-conditional Gaussian model of the feature space, with additional discussion of the selected probability density in Section 4.1.

Algorithm 1 describes the workflow of pFedFDA.

Our algorithm begins with a careful initialization of parameters for the feature extractor ϕ, Gaussian means $\boldsymbol{\mu} = \{\mu^c\}_{c=1}^C$, and covariance Σ (Lines 1-2). We initialize ϕ with established techniques (e.g., [12]) such that the output features follow a Gaussian distribution with controlled variance. We similarly use a spherical Gaussian to ensure a stable initialization of the corresponding generative classifier (see Section 4.1).

Algorithm 1: pFedFDA

1 Server initializes feature extractor ϕ_g^0 with random Gaussian weights.
2 Server and clients initialize feature distribution estimates $(\boldsymbol{\mu_g^0}, \Sigma_g^0)$, $(\boldsymbol{\mu_i^0}, \Sigma_i^0)$ as random spherical Gaussians.
3 **for** *each round* $r = 0, \cdots, R-1$ **do**
4 Send ϕ_g^r, $\boldsymbol{\mu_g^r}$, Σ_g^r to participating clients
5 **for** *each active client* i **do**
6 $\phi_i^r \leftarrow \phi_g^r$
7 Train ϕ_i^r for E epochs using loss (5)
8 Estimate local $\widehat{\boldsymbol{\mu}}_i, \widehat{\Sigma}_i$ via (6) and (7)
9 Estimate interpolation parameter β_i via (9)
10 $\boldsymbol{\mu_i^r} \leftarrow \beta_i \widehat{\boldsymbol{\mu}} + (1-\beta_i)\boldsymbol{\mu}_g$
11 $\Sigma_i^r \leftarrow \beta_i \widehat{\Sigma} + (1-\beta_i)\Sigma_g$
12 Send $\phi_i^r, \boldsymbol{\mu_i^r}, \Sigma_i^r$ to the server
13 **end**
14 Update $\phi_g^r, \boldsymbol{\mu}_g^r, \Sigma_g^r$ with a weighted average of each client's $\phi_i^r, \boldsymbol{\mu}_i^r$, and Σ_i^r
15 **end**

At the start of each FL round r, the server broadcasts the current $\phi_g^r, \boldsymbol{\mu}_g^r, \Sigma_g^r$ to each participating client. The local training of each client consists of two key components: (1) *global representation learning*, in which clients train ϕ to maximize the likelihood of local features under the global feature distribution $\boldsymbol{\mu}_g^r, \Sigma_g^r$ (Line 7); (2) *local distribution adaptation*, in which clients obtain robust estimates of their local feature distribution $\boldsymbol{\mu}_i^r, \Sigma_i^r$, using techniques for efficient low-sample Gaussian estimation (Line 8) and local-global parameter interpolation (Lines 9-11). After local training, clients send their $\phi_i^r, \boldsymbol{\mu}_i^r, \Sigma_i^r$ to the parameter server for aggregation (Line 14).

In the following sections, we provide detailed explanations of each algorithmic component. In Section 4.1 we discuss the benefits of a generative modeling framework and provide the justification for our selected class-conditional Gaussian model. We outline how the resulting generative classifier can be used to guide representation learning in Section 4.2 and describe how we obtain personalized generative classifiers in Section 4.3.

4.1 Generative Model of Feature Distributions

Motivation for Generative Classifiers. A central theme in FL is exploiting inter-client knowledge to train more generalizable models than any client could attain using only their local dataset. This presents an important bias-variance trade-off, as incorporating global knowledge naively can introduce significant bias. Fortunately, under a generative modeling approach, this bias can be naturally handled, enabling efficient inter-client collaboration.

First note that local class priors $p_i(y)$ can be approximated with local counts: $p_i(y = c) \approx \frac{n_i^c}{\sum_{c' \in C} n_i^{c'}} := \pi_i^c$, where n_i^c is the number of local samples whose labels are c. This leaves the primary source of bias to the mismatch between local and global feature distributions $p_g(z|y)$ and $p_i(z|y)$. Crucially, it turns out that this bias is controllable due to the dependence of z on global representation parameters ϕ. Consequentially, we propose to minimize this bias through our classification objective, which we discuss further in Section 4.2.

Class-Conditional Gaussian Model. In this work we approximate the distribution of latent representations using a class-conditional Gaussian with tied covariance, i.e., $p_i(z|y = c) = \mathcal{N}(z|\mu_i^c, \Sigma_i)$. We show the resulting generative classifier under this model in Eq. 4. Note that it has a closed form and results in a decision boundary that is linear in z. I.e., if we know the underlying local feature distribution mean and covariance, we can efficiently compute the optimal header parameters h_i for the inner objective in Eq. 2.

In addition to the convenient form of the Bayes classifier, we select this distribution as the Gaussianity of latent representations is likely to hold in practice. Notably, by adopting the common technique of Gaussian weight initialization (e.g., [12]), the resulting feature space is highly Gaussian at the start of training. It has also been observed that the standard supervised training of neural networks with cross-entropy objectives results in a feature space that is well approximated by a class-conditional Gaussian distribution [24], i.e., the corresponding generative classifier Eq. 4 has equal accuracy to the learned discriminative classifier. We provide a further discussion of this modeling assumption in Appendix A.

$$p(y = c|z) = \frac{\mathcal{N}(z|\mu^c, \Sigma)p(y = c)}{\sum_{c' \in \mathcal{C}} \mathcal{N}(z|\mu^{c'}, \Sigma)p(y = c')}, \tag{3}$$

$$\log p(y = c|z) \propto z^\top \Sigma^{-1}\mu^c - \frac{1}{2}(\mu^c)^\top \Sigma^{-1}\mu^c + \log p(y = c). \tag{4}$$

4.2 Global Representation Learning

Next, we describe our process for training the shared feature extractor ϕ. Similar to existing works [5, 48], our local training consists of training ϕ via gradient descent to minimize the cross-entropy loss of predictions from fixed client classifiers. We obtain our client classifiers through Eq. 4, using global estimates of $\boldsymbol{\mu}_g, \Sigma_g$ and local estimated priors π_i. For computational efficiency, we avoid inverting the covariance matrix by estimating $\Sigma^{-1}\mu^c$ with the least-squares solution $w = \min_{w'} \|\Sigma w' - \mu^c\|$.

The loss of client i for an individual training sample (x, y) is provided in Eq. 5.

$$L(x, y; \phi, \boldsymbol{\mu}, \Sigma, \pi) = \sum_{c=1}^{C} y^c \log p(y^c|\phi(x), \mu^c, \Sigma, \pi). \tag{5}$$

Note that for a spherical Gaussian $\Sigma = \mathbf{I}$ and uniform prior π, we recover a nearest-mean classifier under Euclidean distance. This resembles the established global prototype regularization [43], which minimizes the Euclidean distance of features from their corresponding global class prototypes. Notably, FedPAC [48] uses this prototype loss to align client features. However, this implicitly assumes that all feature dimensions have equal variance, and additionally requires a hyperparameter

https://doi.org/10.52202/079017-2451

λ to balance the amount of regularization with the primary objective. In contrast, our generative classifier naturally aligns the distribution of client features by training ϕ with our global generative classifier.

4.3 Local Distribution Adaptation

Local Estimation. A key component of pFedFDA is the estimation of local feature distribution parameters, used both for model personalization and for updating the global distribution for representation learning.

Given a set of n extracted features Z with n^c examples per class c, a maximum likelihood estimate of the class means and an unbiased estimator of the covariance, respectively, are given by:

$$\widehat{\mu}^c = \frac{1}{n^c} \sum_{j=1}^{n} \mathbf{1}_{\{y_j=c\}} z_j \qquad (6) \qquad\qquad \widehat{\Sigma} = \frac{1}{n-1} \bar{Z}^\top \bar{Z}, \qquad (7)$$

where, with slight abuse of notation, $\bar{Z} \in \mathbb{R}^{n \times d}$ denotes the matrix of centered features with rows corresponding to each original feature z_j centered by their respective means, i.e.,

$$\bar{z}_j = z_j - \sum_{c \in C} \mathbf{1}_{\{y_j=c\}} \widehat{\mu}^c. \qquad (8)$$

Estimators Eq. 6 and Eq. 7 may be noisy on clients with limited local data. To illustrate this, consider the common practical scenario where $n_i \ll d$. The feature covariance matrix Σ_i at client i will be degenerate; in fact, it will have a multitude of zero eigenvalues. In these cases, we can add a small diagonal $\epsilon \mathbf{I}$ to Σ, and replace the non-positive-definite matrices with the nearest positive definite matrix with identical variance. This can be efficiently computed by clipping eigenvalues in the corresponding correlation matrix and followed by converting it back to a covariance matrix with normalization to maintain the initial variance. We refer readers to [11] for a review of low-sample covariance estimation.

Local-Global Interpolation. We introduce this fusion because even with the aforementioned correction to ill-defined covariances, the variance of the local estimates remains highly noisy, indicating the necessity of leveraging global knowledge. It is essential to consider that in the presence of data heterogeneity, clients with differing local data distributions and dataset sizes have varying requirements for global knowledge.

For our Gaussian parameters $\boldsymbol{\mu}, \Sigma$, we consider the introduction of global knowledge through a personalized interpolation between local and global estimates, which can be viewed as a form of prior. We provide an analysis of the high-probability bound on estimation error for an interpolated mean estimate in simple settings in Theorem 1. The full derivation is deferred to Appendix E.

Theorem 1 (Bias-Variance Trade-Off)**.** *Let $C = 1$. Define μ_i as the sample mean of client i's local features $\mu_i := \frac{1}{n_i} \sum_{j=1}^{n_i} z_i^j$, and μ_g as the global sample mean using all N samples across M clients: $\mu_g := \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{n_i} z_i^j$. Assume client features are independent and distributed as $z_i \sim \mathcal{N}(\theta_i, \Sigma_i)$, with true global feature distribution $\mathcal{N}(\theta_g, \Sigma_g)$. We consider the use of global knowledge at client i through an interpolated estimate: $\widehat{\mu}_i := \beta \mu_i + (1-\beta)\mu_g$, where $\beta \in [0,1]$. For any $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that*

$$\|\widehat{\mu}_i - \theta_i\|_2^2 \leq (1-\beta)^2 \|\theta_g - \theta_i\|_2^2 + \left[1 + 4\left(\sqrt{\frac{\log 1/\delta}{c}} + \frac{\log 1/\delta}{c}\right)\right] \left(\frac{2\beta}{n_i} \operatorname{Tr}(\Sigma_i) + \frac{(1-\beta)^2}{N} \operatorname{Tr}(\Sigma_g)\right),$$

where $c > 0$ is an absolute constant.

Intuitively, the estimation error and optimal β depend on the bias introduced by using global knowledge $\|\theta_g - \theta_i\|_2^2$, the variance of local and global features, and the respective data volumes.

We formulate this as an optimization problem, in which clients estimate interpolation coefficients β_i to combine local and global estimates of $(\boldsymbol{\mu}, \Sigma)$ with minimal k-fold validation loss:

$$\beta_i \in \min_{0 \leq \beta \leq 1} \frac{1}{k} \sum_{k} \sum_{(x,y) \in \mathcal{D}_k} L(x, y, \phi, \beta' \widehat{\boldsymbol{\mu}}_k + (1-\beta')\boldsymbol{\mu}_g, \beta' \widehat{\Sigma}_k + (1-\beta')\Sigma_g, \pi_i), \qquad (9)$$

where $\mathcal{D}_k$ is the dataset consisting of the validation samples for the k-th fold, and $(\widehat{\boldsymbol{\mu}}_k, \widehat{\Sigma}_k)$ are the local distribution estimates Eq. 6 and Eq. 7 estimated using the training samples from the k-th fold. In our experiments, we avoid additional forward passes on the local dataset by preemptively storing the feature-label pairs obtained over the latest round of training.

We solve Eq. 9 using off-the-shelf quasi-newton methods (e.g., L-BFGS-B). We additionally explore using separate β terms for the means and covariance (Section 5.3) and recommend the use of a single β term for most applications.

After obtaining β, we set our local estimates of $\boldsymbol{\mu}_i, \Sigma_i$ to their interpolated versions. These estimates are then sent to the server for aggregation. Notably, the server update rule can be viewed as a moving average [52] between the previous round estimate and the client average scaled by β, reducing the influence of local noise in the global distribution estimate. At test time, clients use their local distribution estimates for inference through the classification rule in Eq. 4.

5 Experiments

5.1 Experimental Setup

Datasets, Tasks, and Models: We consider image classification tasks and evaluate our method on four popular datasets. The EMNIST [4] dataset is for 62-class handwriting image classification. The CIFAR10/CIFAR100 [22] datasets are for 10 and 10-class color image classification. The TinyImageNet [23] dataset is for 200-class natural image classification. For EMNIST and CIFAR10/100 datasets, we adopt the 4-layer and 5-layer CNNs used in [48]. On the larger TinyImageNet dataset, we use the ResNet18 [13] architecture. Notably, the feature dimension d for EMNIST/CIFAR CNNs is 128, and 512 for ResNet. We provide additional details in Appendix C.1.

Clients and Dataset Partitions: The EMNIST dataset has inherent covariate shifts due to the individual styles of each writer. We partition the dataset by writer following [6], and train with $M = 1000$ total clients (writers), participating with rate $q = 0.03$. On CIFAR and TinyImageNet datasets, we simulate prior probability shift and quantity skew by partitioning the dataset according to a Dirichlet distribution with parameters $\alpha \in (0.1, 0.5)$, where lower α indicates higher levels of heterogeneity. On these datasets, we use $M = 100$ clients with participation rate $q = 0.3$. Additional details of the partitioning strategy are provided in Appendix C.1.2.

We split each client's data partition 80-20% between training and testing.

Covariate Shift and Data Scarcity: We introduce two modifications to client partitions to simulate the challenges of real-world cross-device FL. We first consider common sources of input noise for natural images, which may result from the qualities of the measuring devices (e.g., camera calibration, lens blur) or environmental factors (e.g., weather, lighting). To simulate this, we select ten image corruptions at five levels of severity defined in [14], and corrupt the training and testing samples of the first 50 clients in CIFAR10/100 with unique corruption-severity pairs. We leave the remaining 50 client datasets unchanged. We refer to these datasets with natural covariate shifts as CIFAR10-S/CIFAR100-S and detail the specific corruptions in Appendix C.1.1.

Second, we perform uniform subsampling of client training sets, leaving them with (75%, 50%, or 25%) of their original samples. These low-sample settings are more realistic for cross-device FL, where clients rely more on knowledge sharing.

Baselines and Metrics: We compare pFedFDA to the following baselines: Local, in which each client trains its model in isolation; FedAvg [34] and FedAvg with fine-tuning (FedAvgFT); APFL [6]; Ditto [26]; pFedMe [41]; FedRoD [3]; FedBABU [35]; FedPAC [48]; FedRep [5]; and LG-FedAvg [29]. We report the average and standard deviation of client test accuracies.

Model Training: We train all algorithms with mini-batch SGD for $E = 5$ local epochs and $R = 200$ global rounds. We apply no data augmentation besides normalization into the range $[-1, 1]$. For pFedFDA, we use $k = 2$ cross-validation folds to estimate a single β_i term for each client. Additional training details and hyperparameters for each baseline method are provided in Appendix C.2.

5.2 Numerical Results

Performance under covariate shift and data scarcity. We first present our evaluation under natural client covariate shift with varying data scarcity in Table 1. In all experiments, pFedFDA outperforms

https://doi.org/10.52202/079017-2451

Table 1: Average (standard deviation) test accuracy on CIFAR10/100-S for varying proportions of training data.

Dataset	CIFAR10-S Dir(0.5)				CIFAR100-S Dir(0.5)			
% Samples	100	75	50	25	100	75	50	25
Local Only	.586(.12)	.476(.16)	.461(.15)	.435(.14)	.157(.05)	.136(.05)	.123(.04)	.093(.04)
FedAvg	.464(.13)	.410(.19)	.389(.17)	.321(.14)	.233(.06)	.212(.06)	.187(.05)	.114(.04)
FedAvgFT	.682(.10)	.579(.19)	.561(.17)	.526(.16)	.302(.06)	.273(.05)	.241(.06)	.160(.05)
APFL	.611(.12)	.520(.17)	.508(.16)	.504(.16)	.164(.05)	.148(.04)	.131(.05)	.105(.04)
Ditto	.668(.10)	.578(.18)	.558(.17)	.527(.16)	.295(.05)	.274(.06)	.239(.05)	.141(.05)
FedBABU	.602(.12)	.522(.17)	.495(.16)	.467(.15)	.187(.05)	.170(.05)	.148(.05)	.107(.04)
FedPAC	.679(.09)	.642(.19)	.594(.16)	.533(.18)	.360(.07)	.330(.07)	.283(.07)	.162(.05)
FedRep	.612(.10)	.541(.17)	.510(.16)	.486(.16)	.176(.05)	.158(.05)	.131(.04)	.100(.04)
FedRoD	.655(.11)	.554(.18)	.537(.18)	.499(.14)	.218(.05)	.186(.05)	.150(.04)	.115(.04)
LG-FedAvg	.584(.13)	.483(.16)	.466(.15)	.433(.14)	.166(.05)	.153(.05)	.127(.05)	.091(.04)
pFedMe	.679(.10)	.583(.18)	.549(.17)	.523(.16)	.289(.06)	.268(.06)	.237(.06)	.153(.05)
pFedFDA	**.724(.09)**	**.706(.10)**	**.661(.11)**	**.595(.12)**	**.361(.08)**	**.342(.08)**	**.326(.08)**	**.227(.07)**

Table 2: Average (standard deviation) test accuracy on multiple datasets.

Dataset	EMNIST	CIFAR10		CIFAR100		TinyImageNet	
Partition	Writers	Dir(0.1)	Dir(0.5)	Dir(0.1)	Dir(0.5)	Dir(0.1)	Dir(0.5)
Local	.242(.23)	.865(.13)	.585(.13)	.368(.09)	.150(.05)	.270(.07)	.099(.03)
FedAvg	.790(.14)	.545(.12)	.625(.07)	.245(.06)	.252(.05)	.155(.04)	.150(.04)
FedAvgFT	.844(.10)	.902(.10)	.742(.08)	.499(.09)	.314(.06)	.384(.07)	.213(.04)
APFL	.841(.10)	.882(.11)	.656(.11)	.388(.09)	.169(.05)	.350(.09)	.177(.05)
Ditto	.843(.10)	.898(.10)	.736(.08)	.504(.08)	.308(.06)	.386(.07)	.211(.04)
FedBABU	.728(.13)	.887(.11)	.678(.11)	.395(.09)	.193(.04)	.365(.07)	.179(.04)
FedPAC	**.856(.09)**	**.908(.09)**	**.767(.07)**	**.560(.08)**	.378(.06)	.366(.07)	.180(.04)
FedRep	.735(.12)	.889(.10)	.668(.10)	.398(.09)	.182(.05)	.359(.07)	.145(.04)
FedRoD	.747(.15)	.885(.11)	.713(.09)	.424(.08)	.224(.05)	.382(.07)	.209(.05)
LG-FedAvg	.666(.13)	.866(.13)	.599(.12)	.381(.09)	.162(.05)	.280(.07)	.105(.03)
pFedMe	.842(.10)	.900(.10)	.740(.09)	.493(.08)	.311(.06)	**.388(.07)**	**.218(.04)**
pFedFDA	.844(.10)	.902(.09)	.763(.07)	.523(.08)	**.385(.07)**	.384(.07)	.214(.04)

the other methods in test accuracy, demonstrating the effectiveness of our method in adapting to heterogeneous client distributions. Additionally, pFedFDA has an increasing benefit relative to other methods in data-scarce settings: on CIFAR10, we improve 4.2% over the second-best method with 100% of training samples and 6.9% with 25%. On CIFAR100, the same improvements range from 0.1% to 6.5%. This indicates the success of our method in navigating the bias-variance trade-off.

Evaluation in more moderate scenarios. Our evaluation of all four datasets in the traditional setting (no added covariate shift, full training data) is presented in Table 2. We note that: (1) our method is still competitive, always ranking within the top 3 methods, and (2) the gap between top methods is smaller than in the previous experimental setting. For example, on EMNIST/CIFAR10, we see that FedAvgFT, FedPAC, and pFedFDA are within $\sim$1% accuracy. We observe larger performance gaps for CIFAR100, with FedPAC and pFedFDA having the best results.

Results under extreme data scarcity. We present additional results at the limits of data scarcity on CIFAR10/100 datasets in Table 3, where we assign a single mini-batch (50) of training examples to each client. Notably, even as $n_i \ll d$, which poses a challenge to local covariance estimation, pFedFDA clients obtain the best test accuracy, indicating the robustness of our local-global adaptation.

Generalization to new clients. We further analyze the ability of our generative classifiers to generalize on clients unseen at training time. To simulate this setting, we first train the server model model using half of the client population. We then evaluate each method on the set of clients not encountered throughout training, using their original input data, as well as their dataset

Table 3: Results under extreme data scarcity on CIFAR10/CIFAR100 Dir(0.5).

	FedAvgFT	Ditto	APFL	FedBABU	pFedMe	FedRoD	FedPAC	pFedFDA
CIFAR10	.692(.17)	.708(.17)	.698(.17)	.684(.19)	.683(.16)	.631(.19)	.710(.16)	**.725(.16)**
CIFAR100	.324(.14)	.308(.14)	.405(.15)	.369(.15)	.305(.14)	.306(.14)	.407(.15)	**.499(.16)**

Table 4: Evaluation of new-client generalization on CIFAR10 Dir(0.5).

	Original Clients	New Clients										
		Clean Data	Motion Blur	Defocus Blur	Gauss Noise	Shot Noise	Impulse Noise	Frost	Fog	JPEG Comp.	Brightness	Contrast
FedAvg	.592(.07)	.584(.08)	.512(.09)	.554(.08)	.568(.07)	.575(.07)	.569(.07)	.465(.08)	.467(.08)	.580(.07)	.557(.08)	.359(.10)
FedAvgFT	.716(.08)	.709(.08)	.689(.08)	.704(.09)	.695(.09)	.699(.09)	.696(.09)	.680(.09)	.672(.09)	.711(.08)	.707(.08)	.688(.09)
FedBABU	.703(.10)	.691(.09)	.682(.08)	.685(.09)	.683(.09)	.680(.09)	.679(.09)	.651(.10)	.661(.09)	.690(.08)	.689(.09)	.670(.09)
FedPAC	.727(.09)	.724(.09)	.695(.09)	.708(.09)	.714(.09)	.712(.09)	.705(.09)	.682(.10)	.683(.09)	.716(.09)	.718(.09)	.667(.09)
pFedFDA	**.738(.08)**	**.738(.08)**	**.702(.09)**	**.719(.09)**	**.729(.08)**	**.739(.07)**	**.725(.08)**	**.695(.09)**	**.684(.09)**	**.738(.08)**	**.733(.08)**	**.689(.09)**

transformed using each corruption from CIFAR-S. Further benchmark details, including fine-tuning (personalization) procedures, are provided in Appendix C.3. As demonstrated in Table 4, our method generalizes well even on clients with covariate shifts not encountered at training time. Moreover, observe that pFedFDA has the highest accuracy on the original clients, highlighting the efficacy of structured generative classifiers when less training data is available (i.e., having 50 rather than 100 clients).

5.3 Ablation of Method Components

We conduct two studies to verify the efficacy of our local-global interpolation method. In Table 5, we see that our interpolated estimates always perform better than using only local data, indicating the benefits of harnessing global knowledge. Learning separate β terms for the means and covariance may be beneficial in low-sample or covariate-shift settings when the local distribution estimate may fluctuate further from the global estimate. However, using a single scalar β appears sufficient and comes with the lowest computational cost (associated with the time to solve Eq. 9).

Table 5: Ablation study on CIFAR100 with Dir(0.1) partition. **NB** denotes clients using only local data to estimate their feature distribution ($\beta_i = 1$). **SB** denotes each client estimating a single β_i for both the means and covariance, **MB** denotes clients computing β_i terms for the means and covariance separately. We show the average computational overhead across all settings.

β Strategy			Dir(0.1) Test Accuracy			Dir(0.5) Test Accuracy			Computation Overhead
NB	SB	MB	CIFAR100	CIFAR100-25%	CIFAR100-S	CIFAR100	CIFAR100-25%	CIFAR100-S	(% seconds/iter.)
✓			.458(.08)	.382(.09)	.436(.08)	.320(.06)	.216(.05)	.296(.06)	(0%)
	✓		.523(.08)	.396(.09)	.487(.08)	.385(.06)	.266(.06)	.361(.08)	($\downarrow$ 22.35%)
		✓	.514(.08)	.423(.09)	.480(.08)	.379(.06)	.275(.06)	.373(.07)	($\downarrow$ 36.11%)

We additionally visualize the spread of learned β across clients as a function of their dataset corruption in Fig. 2. As expected, clients with clean datasets rely more on global knowledge (smaller β values) than corrupted clients. Moreover, corruptions with higher β values (e.g., contrast) often align with the more difficult corruptions encountered in Table 4.

5.4 Communication and Computation

The parameter count and relative communication load of our generative classifiers compared to a simple linear classifier varies depending on class count C and feature dimension d. In our experimental configurations (datasets, architectures), the overhead in total parameter count ranges from 1.1% to 6.8%. See Appendix D.3 for additional details.

In Table 6, we compare the local training time (client-side computation) and total runtime of pFedFDA to baseline methods on CIFAR10. We observe a slight increase in training time relative to FedAvg, which can be attributed primarily to cost of learning our parameter interpolation coefficient β. However, this increase is comparable to the existing methods and is lower than representation-

https://doi.org/10.52202/079017-2451

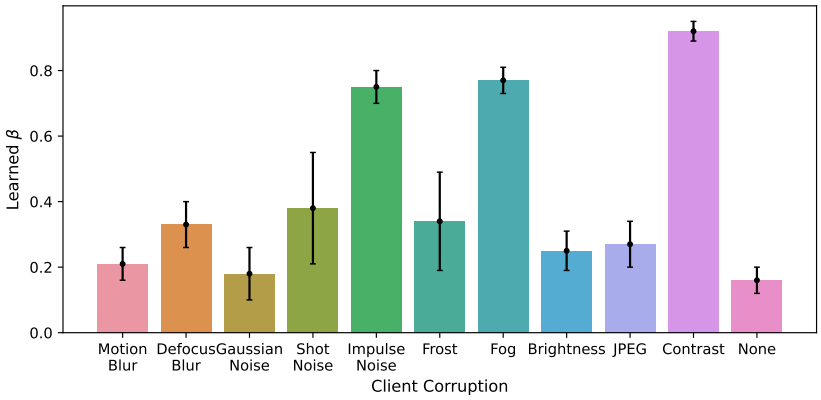


Figure 2: Comparison of client β and local dataset corruption on CIFAR10-S.

learning methods FedRep and FedPAC. This demonstrates the relative efficiency of our generative classifier formulation in comparison to classifiers obtained through local fine-tuning.

Table 6: Comparison of system runtime on the CIFAR10 dataset.

	FedAvg	APFL	Ditto	FedBABU	FedRep	FedPAC	pFedFDA
Local Training (sec)	18.42	21.41	19.97	17.41	34.82	33.60	22.58
Total Runtime (min)	61.41	71.35	66.87	58.11	116.95	146.3	77.71

6 Conclusion

Balancing local model flexibility and generalization remains a central challenge in personalized federated learning (PFL). This paper introduces pFedFDA, a novel approach that addresses the bias-variance trade-off in client personalization through representation learning with generative classifiers. Our extensive evaluation on computer vision tasks demonstrates that pFedFDA significantly outperforms current state-of-the-art methods in challenging settings characterized by covariate shift and data scarcity. Furthermore, our approach remains competitive in more general settings, showcasing its robustness and adaptability. The promising results underline the potential of our method to improve personalized model performance in real-world federated learning applications. Future work will focus on exploring the scalability of pFedFDA and its application to other domains.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the support from the National Science Foundation CAREER award under Grant No. 2340482, the Army Research Laboratory under Cooperative Agreement Number W911NF-23-2-0014, the Sony Faculty Innovation Award, and the National Defense & Engineering Graduate (NDSEG) Fellowship Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the National Science Foundation, or the U.S. government. The U.S. government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation herein. We also thank Ming Xiang for valuable discussions and feedback on this work.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

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

[4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

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

[7] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society*, 1:45–61, 2020.

[8] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel & Distributed Systems*, 33(11):2661–2674, November 2022.

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[10] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

[11] Yaqian Guo, Trevor Hastie, and Robert Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics (Oxford, England)*, 8:86–100, 02 2007.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[15] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020.

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

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

[18] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, Andreas Saleh, Marcus Makowski, Daniel Rueckert, and Rickmer Braren. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, Jun 2021.

 https://doi.org/10.52202/079017-2451

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

[20] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[21] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *International Conference on Machine Learning*, pages 11058–11073. PMLR, 2022.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[23] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[24] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

[25] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021.

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

[27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[28] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

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

[30] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[31] Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V Vasilakos. Privacy and security issues in deep learning: A survey. *IEEE Access*, 9:4566–4593, 2020.

[32] Nathalie Majcherczyk, Nishan Srishankar, and Carlo Pinciroli. Flow-fl: Data-driven federated learning for spatio-temporal predictions in multi-robot systems. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8836–8842. IEEE, 2021.

[33] 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*, 34:15434–15447, 2021.

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

[35] Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[37] Kilian Pfeiffer, Martin Rapp, Ramin Khalili, and Jörg Henkel. Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Computing Surveys*, 55(14s):1–27, July 2023.

[38] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

[39] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2020.

[40] Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*, pages 92–104. Springer, 2019.

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

[42] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[43] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[44] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[45] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. Atpfl: Automatic trajectory prediction model design under federated learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6563–6572, June 2022.

[46] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33:7611–7623, 2020.

[47] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10165–10173, 2021.

[48] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[50] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023.

[51] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.

 https://doi.org/10.52202/079017-2451

[52] Sixin Zhang, Anna Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd, 2015.

[53] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 561–578. Springer, 2020.

[54] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.

A Limitations

- The selected class-conditional Gaussian distribution may not work well for all neural network architectures. For example, if the output features are the result of an activation such as ReLU, a truncated Gaussian distribution may be a better model. Future work can look to exploit knowledge of the neural network architecture to improve the accuracy of the feature distribution estimate.
- In this work, we leverage the insights from a fusion of global and local feature space. As in many applications there is often an underlying cluster structure between clients datasets, future works may explore the identification and efficient estimation of feature distributions of client clusters, in order to reduce the degree of bias introduced in client collaboration.

B Broader Impacts

Federated learning has become the main trend for distributed learning in recent years and has deployed in many popular consumer devices such as Apple's Siri, Google's GBoard, and Amazon's Alexa. Our paper addresses the practical limitations of personalization methods in adapting to clients with covariate shifts and/or limited local data, which is a central issue in cross-device FL applications. We are unaware of any potential negative social impacts of our work.

C Details of Experimental Setup

All experiments are implemented in PyTorch 2.1 [36] and were each trained with a single NVIDIA A100 GPU. Compute time per experiment ranges from approximately 2 hours for CIFAR10/100 and 20 hours for TinyImageNet. Code for re-implementing our method is provided at the following GitHub URL: `https://github.com/cj-mclaughlin/pFedFDA`.

C.1 Dataset Description

The EMNIST [4] dataset contains over 730,000 28×28 grayscale images of 62 classes of handwritten characters. The CIFAR10/CIFAR100 [22] datasets contain 60,000 32×32 color images in 10 and 100 different classes of natural images, respectively. TinyImageNet [23] contains 120,000 64×64 color images of natural images.

C.1.1 CIFAR-S Generation.

We implement the following 10 common image corruptions at 5 levels of severity as described in [14]: Gaussian noise, shot (Poisson) noise, impulse noise, defocus blur, motion blur, fog, brightness, contrast, frost, JPEG compression. We apply a unique corruption-severity pair to all samples of the first 50 clients.

C.1.2 Non-i.i.d. Partitioning.

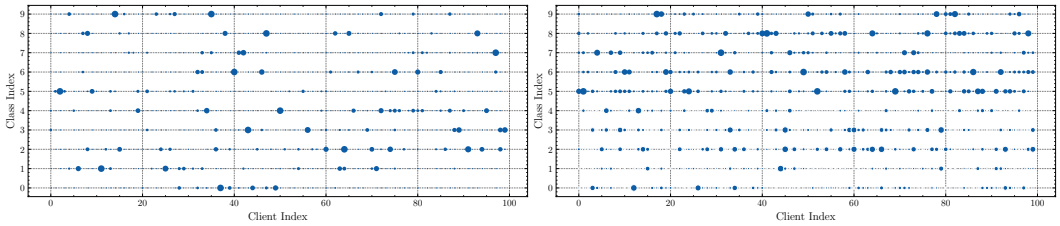
On CIFAR and TinyImageNet datasets, we distribute the proportion of samples of class C across M clients according to a Dirichlet distribution: $q_c, m \sim \texttt{Dir}_M(\alpha)$, where we consider $\alpha \in (0.1, 0.5)$ as in [30].

We provide a visualization of Dirichlet partitioning strategies on CIFAR10 below. The size of each point represents the number of allocated samples. Notably, as α increases, Dir(α) becomes less heterogeneous.

C.2 Training Settings

All methods are trained using mini-batch SGD for 200 global rounds with 5 local epochs of training. We use a fixed learning rate of 0.01, momentum of 0.5, and weight decay of 5e-4. The batch size is set to 50 for all experiments, except for EMNIST, where we use a batch size of 16. We sample the set of active clients uniformly with probability q=0.3 for CIFAR and TinyImageNet and q=0.03 for EMNIST. The last global round of training employs full client participation. We split the data of each client 80-20% between training and testing.

https://doi.org/10.52202/079017-2451



(a) CIFAR10 Dirichlet(0.1) (b) CIFAR10 Dirichlet(0.5)

Figure 3: Comparison of Dirichlet Partitions on CIFAR10.

Hyper-parameters. For APFL, we tune α over [0.25, 0.5, 0.75, 1.0], and set $\alpha = 0.25$. For pFedMe, we tune λ over [1.0, 5.0, 10.0, 15.0] and set $\lambda = 5.0$. For Ditto, we use five local epochs for personalization and tune μ over [0.05, 0.1, 0.5, 1.0, 2.0] and set $\mu = 1.0$. For FedRep and FedBABU, we use five local epochs for training the head parameters. For FedPAC, we tune λ over [0.1, 0.5, 1.0, 5.0, 10.0], and set $\lambda = 1.0$. FedPAC uses one local epoch for training head parameters with a higher learning rate of 0.1, following the original implementation.

C.3 Evaluation on New Clients

Our fine-tuning procedure on new clients largely follows the methodology above. For FedAvgFT, we fine-tune the global model for five local epochs. For FedBABU and FedPAC, we personalize the model in 2 different ways and report the best result: (1) fine-tuning only the head for 5 local epochs, and (2) fine-tuning both the body and head for 5 local epochs. For pFedFDA, each new client estimates their local interpolated statistics (i.e., lines 8-11 of Algorithm 1) to obtain a personalized generative classifier.

For our covariate shift evaluation, we apply a medium severity corruption (level 3) to all samples.

D Additional Results

D.1 Multi-Domain FL

In Table 7, we present results on the DIGIT-5 domain generalization benchmark [53]. This presents an alternate form of covariate shift, as the data from each client is drawn from one of 5 datasets (SVHN, USPS, SynthDigits, MNIST-M, and MNIST). In particular, we use 20 clients trained with full participation, and assign 4 clients to each domain. Within each domain, we use the Dirichlet(0.5) partitioning strategy to assign data to each client. We observe that pFedFDA is effective in all settings, but has the most significant benefits over prior work in the low-data regime.

Table 7: Results on multi-domain DIGIT-5 benchmark for varying data volumes.

DIGIT-5 % Samples	25	50	75	100	Avg. Improvement
Local	76.84	83.11	86.97	88.51	-
FedAvg	81.75 (+4.91)	85.09 (+1.98)	87.41 (+0.44)	88.19 (+0.32)	1.91
FedAvgFT	85.61 (+8.77)	88.72 (+5.61)	90.75 (+3.78)	**91.73 (+3.22)**	5.34
Ditto	83.85 (+7.01)	85.53 (+2.42)	87.43 (+0.46)	88.80 (+0.29)	2.54
FedPAC	82.78 (+5.94)	87.94 (+4.83)	**91.12 (+4.15)**	91.04 (+2.53)	4.36
pFedFDA	**86.54 (+9.70)**	**90.05 (+6.94)**	90.75 (+3.78)	91.56 (+3.05)	**5.86**

D.2 Effect of Local Epochs

In many FL settings, we would like clients to perform more local training between rounds to reduce communication costs. However, too much local training can cause the model to diverge. In Fig. 4, we compare the effect of the local amount of epochs for CIFAR100 and CIFAR100-S-25% sample datasets. We observe that (1) pFedFDA outperforms FedAvgFT at all equivalent budgets of E, (2)

both methods follow exhibit a general plateau in accuracy after $E = 5$, and (3) pFedFDA learns much faster than FedAvgFT, with significantly higher accuracy for $E = 1$.

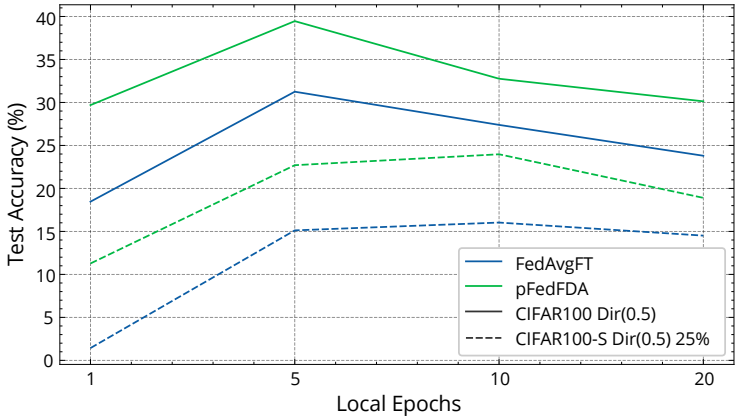


Figure 4: Comparison of average test accuracy with varying local epochs on CIFAR100.

D.3 Communication Load Examples

In Table 8, we compare the number of distinct parameters in our Gaussian estimates to that of a typical linear classifier for the models and datasets used in this paper, along with some additional examples. We display the resulting overhead relative to the base parameter count of the shared representation backbone.

Table 8: Comparison of communication load (parameters/iter.) between our Gaussian distribution parameters (μ, Σ) and standard linear classifiers.

Parameters	Backbone	Linear Classifier $(C \times (d+1))$	Gaussian $(\boldsymbol{\mu}, \Sigma)$ $(C \times d + \frac{1}{2}(d^2 + d))$	Δ Overhead
EMNIST-CNN (EMNIST, $C = 62$)	115776	7998	16192	6.620%
CIFAR-CNN (CIFAR100, $C = 100$)	106400	12900	21056	6.837%
ResNet18 (TinyImageNet, $C = 200$)	11167212	102600	233728	1.164%
MobileNetV3-Small (ImageNet, $C = 1000$)	927008	1615848	1548800	-2.637%

D.4 Runtime of Method Components

In Table 9, we evaluate the proportion of each local iteration of pFedFDA associated with each line of our algorithm. **Network Passes** refers to the time taken to train the base network parameters ϕ (Line 7 of Alg. 1). **Mean/Covariance Est.** refers to the time taken to estimate the local mean and covariance from features extracted during model training (Line 8 of Alg. 1). **Interpolation Optimization** refers to the time taken to optimize the local coefficient β (Line 9 of Alg. 1). Overall, we find that the majority of the overhead of our method comes from estimating the interpolation parameter β.

Table 9: Percentage (average (std)) of the local training time associated each component of our algorithm.

	Network Passes	Mean/Covariance Est.	Interpolation Optimization
CIFAR10	84.88 (6.79)%	0.765 (0.281)%	14.36 (6.62)%
CIFAR100	77.70 (5.75)%	2.861 (0.899)%	19.43 (5.70)%
TinyImageNet	87.41 (1.50)%	2.701 (0.659)%	9.886 (1.14)%

https://doi.org/10.52202/079017-2451

E On the Bias-Variance Tradeoff

This section justifies the bias-variance tradeoff under some simplified technical assumptions. For simplicity, we assume that at any given round, the extracted feature vectors for a class are independent. We illustrate the bias-variance tradeoff in estimating the mean feature of a given class c at round t.

Proof of Theorem 1. For ease of exposition, we drop the time index and class index.

Let i be an arbitrary client with local dataset size n_i of class c. Let N be the total data volume of class c over the entire FL system. Assuming that the distribution of client i's features z follow a multivariate Gaussian distribution $\mathcal{N}(\theta_i, \Sigma_i)$, and the global feature distribution follows $\mathcal{N}(\theta, \Sigma)$ where $\theta_g := \sum_{i=1}^{M} n_i\theta_i/(\sum_{i\in[M]} n_i)$, $\Sigma_g := \sum_{i=1}^{M} n_i^2\Sigma_i/(\sum_{i\in[M]} n_i)^2$. Note θ_i, θ_g are deterministic parameters.

We denote the local and global mean estimates as:

$$\mu_i := \frac{1}{n_i}\sum_{j=1}^{n_i} z_i^j, \text{ and } \mu_g := \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{n_i} z_i^j.$$

Let $\widehat{\mu}_i$ be the local estimate that interpolates between local and global knowledge, defined as

$$\widehat{\mu}_i := \beta\mu_i + (1-\beta)\mu_g. \tag{10}$$

We will focus on bounding the high probability local estimation error $\|\widehat{\mu}_i - \mathbb{E}[\mu_i]\|_2$.

Note that Eq.(10) can be further expanded as

$$\begin{aligned}
\widehat{\mu}_i &= \beta\mu_i + (1-\beta)\left(\frac{n_i}{N}\mu_i + \sum_{i'\neq i}\frac{n_{i'}}{N}\mu_{i'}\right) \\
&= (\beta + (1-\beta)\frac{n_i}{N})\mu_i + (1-\beta)\sum_{i'\neq i}\mu_{i'} \\
&= \gamma\mu_i + (1-\beta)\bar{\mu},
\end{aligned}$$

where $\gamma := \beta + (1-\beta)\frac{n_i}{N}$, and $\bar{\mu} := \frac{1}{N}\sum_{i'\neq i}\sum_{j=1}^{n_{i'}}\mu_{i'}^j$.

Thus $\mu_i \sim \mathcal{N}(\theta_i, \frac{1}{n_i}\Sigma_i)$ and $\bar{\mu} \sim \mathcal{N}(\frac{N\theta_g - n_i\theta_i}{N}, \frac{N\Sigma_g - n_i\Sigma_i}{N^2})$. Since μ_i and $\bar{\mu}$ are independent, we have

$$\widehat{\mu}_i - \theta_i \sim \mathcal{N}\left((1-\beta)(\theta_g - \theta_i),\ \gamma^2\frac{1}{n_i}\Sigma_i + (1-\beta)^2\frac{N\Sigma_g - n_i\Sigma_i}{N^2}\right).$$

Let $\boldsymbol{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We have

$$\widehat{\mu}_i - \theta_i = (1-\beta)(\theta_g - \theta_i) + \bar{\Sigma}^{1/2}\boldsymbol{g},$$

where $\widehat{\Sigma}^{1/2}$ is the square root matrix of $\widehat{\Sigma} := \gamma^2\frac{1}{n_i}\Sigma_i + (1-\beta)^2\frac{N\Sigma_g - n_i\Sigma_i}{N^2}$. It holds that

$$\begin{aligned}
\|\widehat{\mu}_i - \theta_i\|_2^2 = (1-\beta)^2\|\theta_g - \theta_i\|_2^2 + 2(1-\beta)\left\langle\theta_g - \theta_i, \widehat{\Sigma}^{1/2}\boldsymbol{g}\right\rangle \\
+ \left\langle\widehat{\Sigma}^{1/2}\boldsymbol{g}, \widehat{\Sigma}^{1/2}\boldsymbol{g}\right\rangle.
\end{aligned}$$

Taking the expectation with respect to the randomness in the Gaussian random variable $\boldsymbol{g}$ and by the law of total expectation, we have

$$\mathbb{E}\left[2(1-\beta)\left\langle\theta_g - \theta_i, \widehat{\Sigma}^{1/2}\boldsymbol{g}\right\rangle\right] = 2(1-\beta)\left\langle\theta_g - \theta_i, \widehat{\Sigma}^{1/2}\mathbb{E}[\boldsymbol{g}]\right\rangle = 0,$$

and

$$\begin{aligned}\mathbb{E}\left[\left\langle \widehat{\Sigma}^{1/2}\boldsymbol{g}, \widehat{\Sigma}^{1/2}\boldsymbol{g}\right\rangle\right] &\overset{(a)}{=} \mathbb{E}\left[\boldsymbol{g}^\top \widehat{\Sigma}\boldsymbol{g}\right] \\ &= \mathbb{E}\left[\boldsymbol{g}^\top \gamma^2 \frac{1}{n_i}\Sigma_i \boldsymbol{g}\right] + \mathbb{E}\left[\boldsymbol{g}^\top (1-\beta)^2 \frac{N\Sigma_g - n_i\Sigma_i}{N^2}\boldsymbol{g}\right] \\ &= \gamma^2 \frac{1}{n_i}\operatorname{Tr}(\Sigma_i) + (1-\beta)^2 \frac{N\operatorname{Tr}(\Sigma_g) - n_i \operatorname{Tr}(\Sigma_i)}{N^2}.\end{aligned}$$

where equality (a) holds because $(\widehat{\Sigma}^{1/2})^\top(\widehat{\Sigma}^{1/2}) = \widehat{\Sigma}$ as $\widehat{\Sigma}^{1/2}$ is symmetric.

By Hanson-Wright inequality [44, Theorem 6.2], we conclude that with probability at least $1-\delta$ (for any given $\delta \in (0,1)$),

$$\begin{aligned}\|\widehat{\mu}_i - \theta_i\|_2^2 &\le (1-\beta)^2 \|\theta_g - \theta_i\|_2^2 \\ &\quad + (\frac{\beta^2}{n_i} + \frac{2\beta(1-\beta)}{N})\operatorname{Tr}(\Sigma_i) + (1-\beta)^2\frac{1}{N}\operatorname{Tr}(\Sigma_g) \\ &\quad + 4\left\|(\frac{\beta^2}{n_i} + \frac{2\beta(1-\beta)}{N})\operatorname{Tr}(\Sigma_i) + (1-\beta)^2\frac{1}{N}\operatorname{Tr}(\Sigma_g)\right\|_{\mathrm{F}} \max\left\{\sqrt{\frac{\log 1/\delta}{c}}, \frac{\log 1/\delta}{c}\right\} \\ &\overset{(b)}{\le} (1-\beta)^2 \|\theta_g - \theta_i\|_2^2 \\ &\quad + \frac{\beta^2 + 2\beta(1-\beta)}{n_i}\operatorname{Tr}(\Sigma_i) + \frac{(1-\beta)^2}{N}\operatorname{Tr}(\Sigma_g) \\ &\quad + 4\left\|(\frac{\beta^2}{n_i} + \frac{1}{2})\operatorname{Tr}(\Sigma_i) + (1-\beta)^2\frac{1}{N}\operatorname{Tr}(\Sigma_g)\right\|_{\mathrm{F}} \max\left\{\sqrt{\frac{\log 1/\delta}{c}}, \frac{\log 1/\delta}{c}\right\} \\ &\overset{(c)}{\le} (1-\beta)^2 \|\theta_g - \theta_i\|_2^2 \\ &\quad + \frac{2\beta - \beta^2}{n_i}\operatorname{Tr}(\Sigma_i) + \frac{(1-\beta)^2}{N}\operatorname{Tr}(\Sigma_g) \\ &\quad + 4\left(\sqrt{\frac{\log 1/\delta}{c}} + \frac{\log 1/\delta}{c}\right)\left(\frac{2\beta-\beta^2}{n_i}\sqrt{\operatorname{Tr}(\Sigma_i^2)} + \frac{(1-\beta)^2}{N}\sqrt{\operatorname{Tr}(\Sigma_g^2)}\right) \\ &\overset{(d)}{\le} (1-\beta)^2 \|\theta_g - \theta_i\|_2^2 \\ &\quad + \left[1 + 4\left(\sqrt{\frac{\log 1/\delta}{c}} + \frac{\log 1/\delta}{c}\right)\right]\left(\frac{2\beta}{n_i}\operatorname{Tr}(\Sigma_i) + \frac{(1-\beta)^2}{N}\operatorname{Tr}(\Sigma_g)\right),\end{aligned}$$

where $c > 0$ is some absolute constant, inequality (b) holds as $a \cdot b \le (\frac{a+b}{2})^2$, and $N \ge n_i$, inequality (c) holds because of triangular inequality $\|A+B\|_{\mathrm{F}} \le \|A\|_{\mathrm{F}} + \|B\|_{\mathrm{F}}$, that $\|A\|_{\mathrm{F}} = \sqrt{\|A\|_{\mathrm{F}}^2} = \sqrt{\operatorname{Tr}(A^\top A)} = \sqrt{\operatorname{Tr}(A^2)}$ if matrix A is symmetric, and that $\max\{a,b\} \le a+b$, inequality (d) holds because $\operatorname{Tr}(A^2) \le (\operatorname{Tr}(A))^2$ for positive semidefinite matrix A and that $\operatorname{Tr}(\Sigma_i)$, $\beta^2 \ge 0$, and $\operatorname{Tr}(\Sigma_g)$ are by definition non-negative.

□

The first term $(1-\beta)^2\|\theta_g - \theta_i\|_2^2$ is the bias introduced when client i uses global knowledge; the smaller the β, the more bias introduced. The last term reveals the interaction of β and the tradeoff between local and global variance. When β approaches 0, we have the global feature variance $\operatorname{Tr}(\Sigma)$ reduced by the average of N global samples. When β approaches 1, we have local feature variance $\operatorname{Tr}(\Sigma_i)$ reduced by the average of only n_i local data. Thus the bias-variance tradeoff on client i crucially depends on the degree of local-global distribution shift, $\|\theta_g - \theta_i\|_2^2$, the local data volume n_i and its quality (i.e., Σ_i), and the volume and quality of the data across clients N, Σ_g.

https://doi.org/10.52202/079017-2451

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The scope of the paper is on an important topic of client model personalization in federated learning. We faithfully state our contributions in both the abstract and introduction.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and considerations of our Gaussian modelling of the feature space both in the main text and with additional notes in the appendix. While the focus of the paper is in improving client personalization in the challenging setting of data scarcity and client distribution shift, we additionally benchmark our method in more general settings to demonstrate the widespread applicability of our work. Finally, we provide an assessment of the communication and computation overhead of our method compared to state-of-art approaches.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the key assumptions in the main text. The missing proof is deferred to Appendix E.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed experimental setups and review the hyperparameters in Section 5.4 and Appendix C.2. We additionally provide code and instructions to train our method.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our evaluations are based on open-accessed datasets that are publically available. An official implementation code is provided in the supplementary materials.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed experimental setups and review the hyperparameters in Section 5.4 and Appendix C.2. We additionally provide code and instructions to train our method.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For our main experiments we include the standard deviation of client accuracies, and include std error bars in our ablation visualization of the method components.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please find the software/hardware specifications in Appendix C.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The NeurIPS code of ethics is strictly enforced throughout our research.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the broader impacts of our work in Appendix B. Please find details therein.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The existing assets used in this paper has been adequately cited or credited to.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our attached code is well documented and comes with a README file indicating how reviewers may set up our experiments and train the proposed method.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

https://doi.org/10.52202/079017-2451

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects