Capture Global Feature Statistics for One-Shot Federated Learning

Zenghao Guan[1, 2, 3], **Yucan Zhou**[1, 3*], **Xiaoyan Gu** [1, 2, 3*]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Key Laboratory of Cyberspace Security Defense
zenghaoguan.cs@gmail.com, {zhouyucan, guxiaoyan}@iie.ac.cn

Abstract

Traditional Federated Learning (FL) necessitates numerous rounds of communication between the server and clients, posing significant challenges including high communication costs, connection drop risks and susceptibility to privacy attacks. One-shot FL has become a compelling learning paradigm to overcome above drawbacks by enabling the training of a global server model via a single communication round. However, existing one-shot FL methods suffer from expensive computation cost on the server or clients and cannot deal with non-IID (Independent and Identically Distributed) data stably and effectively. To address these challenges, this paper proposes **FedCGS**, a novel **Fed**erated learning algorithm that **C**apture **G**lobal feature **S**tatistics leveraging pre-trained models. With global feature statistics, we achieve training-free and heterogeneity-resistant one-shot FL. Furthermore, we extend its application to personalization scenario, where clients only need execute one extra communication round with server to download global statistics. Extensive experimental results demonstrate the effectiveness of our methods across diverse data-heterogeneity settings.

Code — https://github.com/Yuqin-G/FedCGS

Introduction

Federated Learning (FL) is an emerging framework that enables multiple parties to participate in collaborative learning under the coordination of a central server, which aggregates model updates rather than private data, enhancing privacy in distributed learning (McMahan et al. 2017). However, typical FL requires numerous communication rounds between the server and clients, leading to significant challenges. First, clients must maintain constant connections with the server to upload and receive updates, resulting in high communication costs and the risk of connection drops (Li et al. 2020b; Kairouz et al. 2021; Chen et al. 2023), which is unbearable for bandwidth-limited or real-time FL applications. Second, frequent communication increases the system's vulnerability to data and model poisoning attacks (Mothukuri et al. 2021; Rao et al. 2024; Guan et al. 2024; Yazdinejad et al. 2024), as adversaries can refine their strategies by exploiting global model updates, compromising the learning process and model integrity.

To solve above challenges, one-shot FL, which restricts the communication rounds between clients and the server to a single iteration (Guha, Talwalkar, and Smith 2019), has emerged as a promising solution. Existing one-shot FL methods can be broadly divided into three categories: 1) Knowledge distillation methods (Li, He, and Song 2021; Zhang et al. 2022; Dai et al. 2024). 2) Generative methods (Heinbaugh, Luz-Ricca, and Shao 2023; Yang et al. 2024a, 2023, 2024b). 3) Bayesian methods (Neiswanger, Wang, and Xing 2013; Jhunjhunwala, Wang, and Joshi 2024; Hasan et al. 2024). Knowledge distillation methods obtain global models by distilling knowledge from an ensemble of client models using auxiliary public data or synthetic data generated by the ensemble. However, these distillation steps impose a significant computational cost on the server and require careful hyperparameter tuning (Kurach et al. 2019). Generative methods aim to train the global model on the server by using synthetic samples that match the distribution of each client's data. These approaches, though effective, brings about privacy concerns when the generated data closely mimics the original client data in the server (Rao et al. 2024; Carlini et al. 2023). Bayesian methods approximate each client's posterior distribution and aggregate them into a global model within a single communication round. However, these methods face high computational and memory costs, and using approximations to reduce these costs can lead to significant errors, compromising both calibration and accuracy (Neiswanger, Wang, and Xing 2013; Hasan et al. 2024). Moreover, both of these three kinds of methods have poor performance and robustness when dealing with non-IID (Independent and Identically Distributed) data.

Recently, FedPFT (Beitollahi et al. 2024) has been proposed to leverage pre-trained models to improve both the accuracy and computation efficiency of one-shot FL. Specifically, each client uploads the Gaussian Mixture Models (GMMs) learned from class-conditional feature. Subsequently, the server trains a classifier using synthetic features generated from GMMs on the server. With competitive performance, low computation overhead and robustness to data heterogeneity, FedPFT (Beitollahi et al. 2024) shows the feasibility of adapting the classifier with fixed

*Corresponding author

pre-trained backbone in one-shot FL. However, GMMs may not effectively fit the local feature distributions. Additionally, different sampling and training strategies will influence the final performance. Actually, there is no need to generate clients' features for training the classifier. We can obtain the parameter-free classifier heads using feature statistics if we could get the global feature statistics. In this paper, we propose **FedCGS**, a novel **Fed**erated algorithm that **C**apture **G**lobal feature **S**tatistics in a communication-efficient and privacy-preserving way by leveraging pre-trained models. We then utilize these statistics to achieve global one-shot FL and personalized one-shot FL with competitive performance. Our key contributions are highlighted as follows:

- We propose FedCGS, a FL framework that utilizes pre-trained model to capture global feature statistics in computation-efficient and privacy-preserving manner.
- Leveraging global feature statistics, we make use a parameter-free Naive Bayes classifier instead of the learnable linear classifier to achieve heterogeneity-resistant one-shot FL with competitive performance. Additionally, we propose personalized one-shot FL that each client uses these statistics as feature alignment regularizer for local training through one extra communication round to download the global statistics.
- Extensive experiments show that FedCGS improve communication-accuracy frontier both in label shift and feature shift settings.

Related Work

One-shot Federated Learning

Existing one-shot FL methods can be broadly divided into three categories: 1) Knowledge distillation methods (Li, He, and Song 2021; Zhang et al. 2022; Dai et al. 2024). 2) Generative methods (Heinbaugh, Luz-Ricca, and Shao 2023; Yang et al. 2024a, 2023, 2024b). 3) Bayesian methods (Neiswanger, Wang, and Xing 2013; Jhunjhunwala, Wang, and Joshi 2024; Hasan et al. 2024).

Guha et al. (Guha, Talwalkar, and Smith 2019) introduce the first one-shot FL algorithm, which uses an ensemble of local models as the final global model. Knowledge distillation methods aim to reduce storage by consolidating the ensemble into a single model through knowledge distillation. FedKT (Li, He, and Song 2021) achieve knowledge distillation using public data. DENSE (Zhang et al. 2022) eliminates the dependency on public data by using the ensemble of client-uploaded local models to train a generator, which then produces synthetic data for knowledge distillation. Similarly, Co-Boosting (Dai et al. 2024) progressively enhances both the ensemble model and the synthesized data to improve knowledge distillation. However, these methods demand substantial computation on the server and require meticulous hyperparameter tuning.

Generative methods are proposed to train the global model on the server using generated data. DOSFL (Zhou et al. 2020) trains the global model on the server using the distilled synthetic data from local clients. FedCVAE (Heinbaugh, Luz-Ricca, and Shao 2023) trains a conditional variation auto-encoder (CVAE) on each client, after which the decoders and the local label distribution are sent to the server to generate data for training a global model. FedDISC (Yang et al. 2024a) utilize data features to generate samples through auxiliary pre-trained diffusion model in context of semi-supervised FL. Like this, FGL (Zhang et al. 2023), FedCADO (Yang et al. 2023) and FedDEO (Yang et al. 2024b) upload text prompts, classifiers and descriptions related with local data distribution respectively to provide suitable guidance for diffusion model. Nevertheless, generating data on the server that closely resembles client data raises privacy concerns (Carlini et al. 2023).

Bayesian methods estimate the posterior distribution of each client model within a Bayesian framework and aggregate them into a global model. (Neiswanger, Wang, and Xing 2013) demonstrate how to aggregate local posteriors in a single communication round in FL. (Hasan et al. 2024) introduce a novel aggregation technique that interpolates between predictions from the Bayesian Committee Machine (Tresp 2000), reducing errors from approximations (Neiswanger, Wang, and Xing 2013). FedFisher (Jhunjhunwala, Wang, and Joshi 2024) and FedLPA (Liu et al. 2023) utilize the empirical Fisher information matrix to approximate local posteriors, which are then used to compute the mode of the global posterior. Although Bayesian one-shot FL methods offer strong theoretical guarantees, they suffer from high computational costs, especially when dealing with large models.

Recently, FedPFT (Beitollahi et al. 2024) was introduced to enhance both accuracy and computational efficiency in one-shot FL by leveraging pre-trained models. In this approach, each client uploads the GMMs learned from class-conditional features, and the server then trains a global classifier using synthetic features generated from these GMMs. FedPFT (Beitollahi et al. 2024) demonstrates the effectiveness of adapting a classifier with a fixed pre-trained backbone in one-shot FL. However, sampling features to train a linear classifier can lead to a performance bottleneck, as GMMs may not effectively fit the local feature distributions, especially when local data is insufficient. Moreover, the variations of sampling strategies and classifier training configurations will affect the final performance. Distinct from this, we capture global feature statistics through a carefully designed uploading scheme and directly obtain the Gaussian Naive Bayes head as the global classifier in a training-free manner.

Federated Learning with pre-trained Models

Pre-trained models, such as ResNet (He et al. 2016), ViT (Dosovitskiy et al. 2021) or BERT(Devlin et al. 2019), have been widely used to benefit downstream tasks. Recently, FL with pre-trained Models is becoming a popular topic with the increasing prevalence of pre-trained models (Nguyen et al. 2022; Chen et al. 2022). Due to the millions of parameters in pre-trained models, fine-tuning the entire model in FL leads to high communication costs and memory footprint issues. Therefore, recent research suggests that using parameter-efficient tuning methods, such as Adapter (Chen et al. 2024), Prompt Tuning (Guo, Guo, and Wang 2023; Guo et al. 2023; Li et al. 2024; Bai et al. 2024), Low-Rank

Adaption (Wu et al. 2024; Nguyen, Munoz, and Jannesari 2024). In our method, we fix the backbone of the pre-trained model, and utilize it to extract features from local dataset for subsequent steps.

Method

Problem Formulation

Federated Learning (FL) involves a server coordinating with several clients to collaboratively train a global model without sharing private data. Suppose a FL system that consists of a central server and M clients with their local datasets $D_1, \ldots, D_M$, correspondingly. These local datasets are sampled from M distinct distributions and have different sizes. The goal of FL is to train a global model w that optimizes the following loss function across all clients:

$$\min_{\theta} \mathcal{L}(\theta) = \frac{1}{M} \sum_{i \in [M]} \mathcal{L}_i(\theta), \tag{1}$$

where $\mathcal{L}_i(\theta) := \frac{1}{|D_i|} \sum_{\delta_i \in D_i} \ell(\theta; \delta_i)$ is the empirical risk objective computed on the local data set D_i at the i-th client, $\ell(\cdot, \cdot)$ is a loss function and δ_i denotes a sample from D_i.

In contrast, the goal of personalized Federated Learning (pFL) is to personalize the global model with labeled data of each client to better adapt to the local data distribution:

$$\min_{\theta_{1:M}} \mathcal{L}(\theta_{1:M}) = \frac{1}{M} \sum_{i \in [M]} \mathcal{L}_i(\theta_i). \tag{2}$$

Here, $\theta_{1:M}$ denotes the collection of local models.

Capture Global Feature Statistics

As shown in Fig 1, in our FedCGS, each client i extracts class-specific features from its own dataset for each class $j \in \{1, \ldots, C\}$ and calculates the statistics $S_i = \{B_i\} \cup \{(A_i^j, N_i^j) \mid j = 1, \ldots, C\}$ as follows:

$$A_i^j = \sum\nolimits_{x \in D_i^j} f(x), \tag{3}$$

$$B_i = \sum\nolimits_{x \in D_i} f(x)^\top f(x), \tag{4}$$

$$N_i^j = |D_i^j|, \tag{5}$$

where $D_i^j = \{x_k \in D_i \mid y_k = j\}$ and f denotes the feature extractor of the pre-trained model. After that, each client uploads S_i to the server. Now, we demonstrate how to calculate global prototype and global empirical covariance (of features) with these uploaded parameters:

global prototype $\boldsymbol{\mu}$ $= [\mu^1, \mu^2, \cdots, \mu^C]$. We denote N^j as the total number of instances belonging to class j over all clients ($N^j = \sum_{i=1}^M N_i^j$) and A^j as the sum of features belonging to class j over all clients ($A^j = \sum_{i=1}^M A_i^j$):

$$\mu^j = \frac{1}{N^j} \sum_{i=1}^{M} \sum_{x \in D_i^j} f(x) = \frac{1}{N^j} \sum_{i=1}^{M} A_i^j = \frac{1}{N^j} A^j, \tag{6}$$

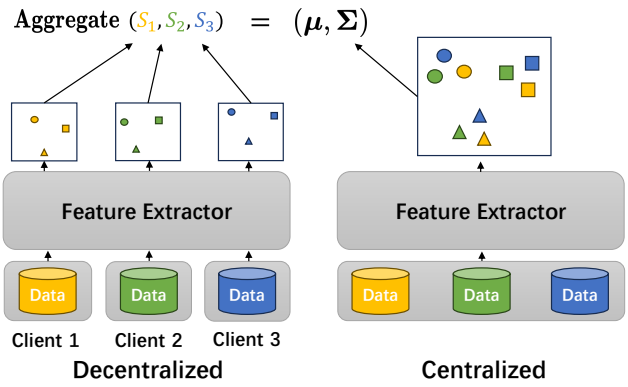

Figure 1: Framework of our FedCGS. S_i is the local statistics of client i as shown in . We obtain global feature statistics through aggregating local feature statistics

global empirical covariance of features $\boldsymbol{\Sigma}$. Here, we denote $A = \sum_{j=1}^C A^j$, $B = \sum_{i=1}^M B_i$, $N = \sum_{j=1}^C N^j$ and $f(x) \triangleq f$ for simplicity:

$$\begin{aligned}
\boldsymbol{\Sigma} &= \frac{1}{N-1} \sum_{i=1}^{M} \sum_{x \in D_i} (f-\mu)^\top (f-\mu) \\
&= \frac{1}{N-1} \sum_{i=1}^{M} \sum_{x \in D_i} (f^\top f - \mu^\top f - f^\top \mu + \mu^\top \mu) \\
&= \frac{1}{N-1} \sum_{i=1}^{M} (B_i - \mu^\top A_i - A_i^\top \mu + N_i \mu^\top \mu) \\
&= \frac{1}{N-1} (B - \mu^\top A - A^\top \mu + N \mu^\top \mu),
\end{aligned} \tag{7}$$

where μ denotes the global mean of features. We could get μ as follow:

$$\mu = \frac{1}{N} \sum_{i=1}^{M} \sum_{x \in D_i} f(x) = \frac{1}{N} \sum_{i=1}^{M} A_i = \frac{1}{N} A. \tag{8}$$

Algorithm 1 summarizes the procedure of FedCGS with pseudocode. Noticed that each client is required to upload the class-wise label count N_i^j, which could potentially lead to the leakage of label information. However, for subsequent calculations, only the aggregated label counts N^j is necessary. Therefore, we can employ Secure Aggregation (Bonawitz et al. 2017) as shown in line 5, ensuring that the server does not gain access to any individual client's label information. Detailed discussion about privacy can be found in Discussion.

Subsequently, we illustrate how to utilize global feature statistics to achieve global one-shot FL and personalized one-shot FL with outstanding performance.

FedCGS for global one-shot FL

Instead of sampling synthetic features from GMMs to directly train the linear head (Beitollahi et al. 2024), we fix the pre-trained models as backbone and use a Gaussian Naive

Algorithm 1: Procedure of FedCGS

Server Executes:

for client $i = 1$ to M **do** {**in parallel**}
  $\{A_i^j, N_i^j\}_{j=1}^C, B_i \leftarrow$ **ClientStats**(D_i)
end for
for each class $j \in C$ **do**
  $N^j \leftarrow \mathbf{SecureAgg}\left(\sum_{j=1}^M N_i^j\right)$
  $A^j \leftarrow \sum_{i=1}^M A_i^j$
  $\mu^j \leftarrow \frac{1}{N^j} A^j$
end for
$N \leftarrow \sum_{j=1}^C N^j$
$A \leftarrow \sum_{i=1}^M \sum_{j=1}^C A_i^j$
$B \leftarrow \sum_{i=1}^M B_i$
$\mathbf{\Sigma} \leftarrow \frac{1}{N-1}\left(B - \mu^\top A - A^\top \mu + N\mu^\top \mu\right)$

ClientStats(D_i)

for each class $j \in C$ **do**
  Let $D_i^j = \{x_k \in D_i \mid y_k = j\}$, $|D_i^j| = N_i^j$
  $A_i^j \leftarrow \sum_{x \in D_i^j} f(x)$
end for
$B_i \leftarrow \sum_{x \in D_i} f(x)^\top f(x)$
return $S_i = \{B_i\} \cup \{(A_i^j, N_i^j) \mid j = 1, \ldots, C\}$

Bayes classifier as the head of network, which can be configured directly from feature statistics. For class j, the class probability for a test point x^* is:

$$p(y^* = j | f(x^*), \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{\pi_j \mathcal{N}(f(x^*) | \mu^j, \mathbf{\Sigma}))}{\sum_{j'}^C \pi_{j'} \mathcal{N}(f(x^*) | \mu^{j'}, \mathbf{\Sigma})} \tag{9}$$

$$= \frac{\pi_j \exp\left((\mu^j)^\top \mathbf{\Sigma}^{-1} f(x^*) - \frac{1}{2}(\mu^j)^\top \mathbf{\Sigma}^{-1} \mu^j\right)}{\sum_{j'}^C \pi_{j'} \exp\left((\mu^{j'})^\top \mathbf{\Sigma}^{-1} f(x^*) - \frac{1}{2}(\mu^{j'})^\top \mathbf{\Sigma}^{-1} \mu^{j'}\right)}, \tag{10}$$

where $\pi_j = \frac{N^j}{N}$. Therefore, the weight $W \in \mathbb{R}^{C \times d}$ and the bias $b \in \mathbb{R}^C$ for the classifier can be expressed as:

$$w_j = \mathbf{\Sigma}^{-1} \mu^j, b_j = \log \pi_j - \frac{1}{2}(\mu^j)^\top \mathbf{\Sigma} \mu^j. \tag{11}$$

Refer to appendix for detailed calculation.

FedCGS for personalized one-shot FL

Unlike existing one-shot FL, FedCGS can benefit personalized FL leveraging global feature statistics. The clients only need to execute one extra communication round to download global feature statistics after obtaining global feature statistics, which is the reason we call it personalized one-shot FL.

In personalized FL, each client usually owns insufficient data, making the locally learned feature representation prone to overfitting and poor generalization. Here, each client uses global prototypes $\boldsymbol{\mu}$ downloaded from server as a regularization term for better local feature representation learning (finetune entire model on local datasets). Specifically, taking client i as an example, the objective of local training can be formulated as:

$$\min_{\theta_i} \mathcal{L}(\theta_i) = \mathcal{L}_i(\theta_i) + \lambda \mathcal{R}(\theta_i, \boldsymbol{\mu}),$$
$$= \mathcal{L}_i(\theta_i) + \frac{\lambda}{N_i^j} \sum_{j=1}^C \sum_{x \in D_i^j} \left\| f(x; \theta_i) - \mu^j \right\|_2^2. \tag{12}$$

Feature alignment regularizer $\mathcal{R}(\cdot, \cdot)$ has been proven to be highly effective in few-shot learning, domain adaption and federated learning (Li et al. 2020a; Tan et al. 2022; Xu, Tong, and Huang 2023). Different from existing personalized FL methods that involves feature alignment, our global prototype $\boldsymbol{\mu}$ remains fixed while others, like FedProto (Tan et al. 2022), updates it in the server with each communication round.

Experiments

To show the effectiveness of our proposed FedCGS, we conduct experiments on both the global one-shot FL and the personalized one-shot FL. More details and extra results are included in the appendix.

Global one-shot FL

Datasets and Partitions. Our experiments are conducted for classification task on the following image datasets: SVHN (Netzer et al. 2011), CIFAR10 (Krizhevsky, Nair, and Hinton 2009b), CIFAR100 (Krizhevsky, Nair, and Hinton 2009a), PACS (Li et al. 2017), and OfficeHome (Venkateswara et al. 2017). For label shift scenario, we use Dirichlet distribution to generate disjoint non-IID client training datasets as same as other global one-shot FL methods (Zhang et al. 2022; Heinbaugh, Luz-Ricca, and Shao 2023; Dai et al. 2024) for fair comparison. For feature shift scenario, we follow the domain generalization settings in (Bai et al. 2024). Specifically, we select three domains for training and distribute their data across M clients. Data from a single domain may be spread across multiple clients, but each client belongs to only one domain. The global model is then tested on the target domain. For all datasets, we adopt a commonly used ResNet18 (He et al. 2016) pre-trained on ImageNet as the backbone.

Baselines. To evaluate our proposed method, we compare it with the canonical baseline FedAvg (McMahan et al. 2017), Ensemble and other state-of-the-art (SOTA) methods in one-shot FL : Dense (Zhang et al. 2022), Co-Boosting (Dai et al. 2024), and FedPFT (Beitollahi et al. 2024). DENSE is the first data-free knowledge distillation method for one-shot federated learning, which distills knowledge from an ensemble of client models. Ensemble maintains an ensemble of local models at the server, serving as the upper bound for DENSE. Ensemble is a strong baseline, but it requires storing all local models on the server, which leads to significant storage overhead and limits scalability and efficiency. Co-Boosting is the SOTA one-shot FL method based on data-free knowledge distillation, which improves the quality of the ensemble model and data generation simultaneously. We avoid comparing with baselines that are inherently multi-round, such as FedProx (Sahu et al. 2019), as

	α	FedAvg (one-shot)	Ensemble	DENSE	Co-Boosting	FedPFT	FedCGS
CIFAR10	0.05	13.97±1.26	38.81±0.89	31.26±0.73	44.37±0.42	56.08±0.49	**63.95±0.00**
	0.1	27.28±1.34	57.29±0.54	56.21±0.24	60.41±0.67	56.43±0.23	**63.95±0.00**
	0.5	51.91±0.46	66.00±0.42	62.42±0.43	**67.43±0.36**	56.80±0.18	63.95±0.00
CIFAR100	0.05	17.57±0.58	22.43±0.63	14.31±0.61	20.30±0.76	36.79±0.21	**39.95±0.00**
	0.1	21.46±0.38	28.07±0.47	17.21±0.36	24.63±0.64	37.16±0.34	**39.95±0.00**
	0.5	35.26±0.22	37.89±0.49	26.49±0.32	34.43±0.45	37.95±0.27	**39.95±0.00**
SVHN	0.05	16.75±0.63	42.26±0.54	37.49±0.42	41.90±0.38	42.55±0.24	**57.77±0.00**
	0.1	24.88±0.39	53.34±0.23	51.53±0.37	57.13±0.28	43.03±0.17	**57.77±0.00**
	0.5	44.39±0.46	82.93±0.29	77.44±0.26	**84.65±0.24**	43.84±0.42	57.77±0.00

Table 1: Test accuracy (%) of the global model for different methods over three datasets at three levels of statistical heterogeneity (where a lower α signifies more heterogeneity).

	Domain	FedPFT	FedCGS
PACS	P	87.98±1.13	**91.02±0.00**
	A	58.34±2.30	**64.21±0.00**
	C	46.15±1.55	**52.30±0.00**
	S	36.96±1.62	**41.03±0.00**
	Avg.	57.36±1.65	**62.14±0.00**
OfficeHome	P	67.98±0.71	**68.75±0.00**
	A	55.74±0.46	**57.44±0.00**
	C	39.26±1.01	**40.99±0.00**
	R	67.75±0.68	**73.28±0.00**
	Avg.	57.68±0.72	**60.12±0.00**

Table 2: Test accuracy (%) of the global model of different methods in PACS and OfficeHome dataset on the domain generalization test mechanism.

their performance would be similar to FedAvg after a single round. Additionally, we exclude algorithms that need auxiliary data (Lin et al. 2020) or pre-trained generative models (Heinbaugh, Luz-Ricca, and Shao 2023; Yang et al. 2024b,a, 2023) to ensure fairness of comparison. In feature shift scenario, since the official implementations for this settings are not provided, we mainly compare our FedCGS with FedPFT. Additional comparisons with other baselines can be found in **appendix**.

Configurations. In the label shift scenario, we simulate a federated learning environment with 10 clients and set the client participation ratio ρ to 1, as in existing one-shot federated learning methods (Zhang et al. 2022; Dai et al. 2024). In the feature shift scenario, the data from each source domain is randomly split across 5 clients, resulting in 15 clients in total for 3 domains. For methods involving backpropagation training (FedAvg, Ensemble, DENSE, Co-Boosting, FedPFT), we set the batch size to 128, the number of epochs to 50, and use the Stochastic Gradient Descent (SGD) optimizer with momentum = 0.9 and the learning rate = 0.01. For data-free knowledge distillation, we use the same generator as in (Dai et al. 2024; Zhang et al. 2022). It is trained with the Adam optimizer, a learning rate of 1e-3, for 30 epochs. FedPFT uploads the GMMs $\mathcal{G}(K_g)$ with a diagonal covariance. Here $\mathcal{G}(K_g)$ is the family of all Gaussian mixture distributions comprised of K_g components. Specifically, we set K_g as 10. For each client, the class-wise label counts are used to sample the corresponding number of features from each GMMs to construct the training dataset.

Experimental Results. We verify that the proposed FedCGS outperforms existing global one-shot FL methods in most cases. As shown in Table 1 and Table 2, we test the performance of global one-shot method in the label-shift setting and feature-shift setting, respectively. For label shift setting, we have the following observations: (1) Traditional FL algorithm FedAvg, which usually requires multiple rounds to converge, performs poorly in this one-shot scenario. (2) Ensemble is the upper bound of DENSE, and it is surpassed by Co-Boosting in some cases. Co-Boosting introduces several strategies to improve both the generated data and the quality of the ensemble model. However, these improvements cannot be achieved under high statistical heterogeneity. (3) All baselines except FedPFT suffer from a large performance deterioration under high statistical heterogeneity ($\alpha = 0.1, 0.05$) (4) Most existing methods exhibit considerable performance variability with different random seeds. (5) In our FedCGS, dataset partitioning does not affect the global quantities A and B, so our FedCGS is immune to statistical heterogeneity and achieves competitive results compared to most existing global one-shot FL methods, especially under extremely high statistical heterogeneity ($\alpha = 0.05$). Specifically, FedCGS surpasses the best baseline by substantial margins with 7.87%, 3.16%, 15.22% on CIFAR10, CIFAR100, SVHN respectively. Additionally, since no training is involved, the performance remains stable. For feature shift, as demonstrated in our results, FedCGS achieves the highest average accuracy and consistently outperforms FedPFT across all target domains. This further substantiates the efficacy of our proposed method.

Personalized one-shot FL

Datasets and Partitions. As same as global one-shot FL, we conduct experiments on SVHN (Netzer et al. 2011), CIFAR10 (Krizhevsky, Nair, and Hinton 2009b), CIFAR100 (Krizhevsky, Nair, and Hinton 2009a) using ResNet18 pre-

trained on ImageNet. For data partitions, we follow the previous personalized FL methods (Zhang et al. 2020; Huang et al. 2021; Xu, Tong, and Huang 2023) that all clients have same data size, owning $s\%$ of data (20% by default) uniformly sampled from all classes and $(100 - s)\%$ from a set of dominant classes.

Baselines. We compare the performance of our method against following baselines: Local-only, where each client trains its model locally; FedAvg (McMahan et al. 2017) with vanilla local training; FedAvg-FT that learns a single global model and locally fine-tuned on local datasets, a simple but strong baseline; FedProto (Tan et al. 2022), FL with only prototype sharing.

Configurations. We use ResNet18 pre-trained on ImageNet for all datasets. During local training phase for each client, we employ mini-batch SGD as the local optimizer and set the batch size to 128, the local epoch to 1 for traditional personalized FL, 200 for Local-only and ours. The momentum is set to 0.5, the learning rate is set to 0.01, the weight decay is set to 5e-4 as (Xu, Tong, and Huang 2023). The number of global communication rounds for traditional personalized FL is set to 100 across all datasets. All results are reported averaged across 3 random seeds.

Experimental Results. As shown in Table 3, FedProto (Tan et al. 2022) and ours consistently outperform the baseline FedAvg (McMahan et al. 2017) and Local-only. This suggests that the discrepancy between local and global feature distributions could lead to higher generalization error and the feature alignment regularization is a good solution. Additionally, We observer that our FedCGS performs similarly to, and in some cases better than, FedProto (Tan et al. 2022) (e.g., on CIFAR10). That means that fixed global prototypes in FedCGS plays same role as the updated global prototypes in FedProto (Tan et al. 2022), which illustrates that features extracted by pre-trained models are sufficiently strong enough to capture meaningful patterns (Janson et al. 2022; Beitollahi et al. 2024). FedAvg-FT achieves best results across all datasets, which verifies the statement that FedAvg-FT is a strong baseline, even outperforming many personalized FL approaches. We want to emphasize that the contribution of our personalized one-shot FL lies in providing a method to enhance local model performance without requiring multiple communication rounds.

	CIFAR10	CIFAR100	SVHN
Local-only	76.25±0.26	54.06±0.28	80.73±0.37
FedAvg	75.70±0.58	48.17±0.51	81.16±0.41
FedAvg-FT	80.47±0.32	60.97±0.24	84.03±0.23
FedProto	78.07±0.76	55.97±0.72	82.14±0.57
FedCGS	78.30±0.39	55.56±0.61	81.20±0.45

Table 3: The comparison of final test accuracy (%) on different datasets.

In-deepth Study

Naive Gaussian classifier head. We use the real global feature statistics captured by FedCGS to generate features in a GMM style and train the linear head as FedPFT. Additionally, we train a linear head in the centralized scenario (using all raw features) and compare the results with our Naive Gaussian classifier head. As shown in Fig 2, the Naive Gaussian classifier head achieves performance comparable to "Centralized" and outperforms "Linear" (the linear head trained with generated features). "Linear" can be regarded as the upper bound of FedPFT, since it uses global feature statistics to generate features for training, while FedPFT uses local statistics. This explains why FedCGS outperforms FedPFT.

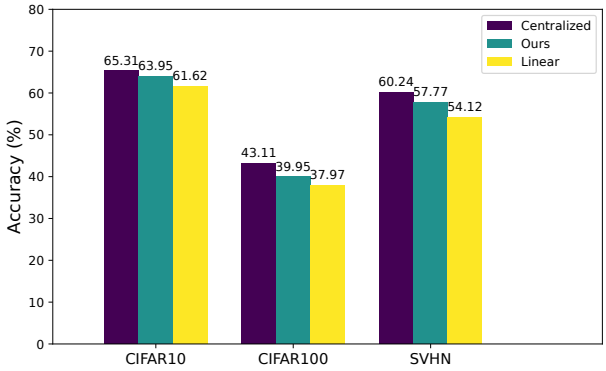

Figure 2: Performance comparison using different classifier configurations.

Feature expansion. Given that pre-trained feature extractors may not expressive enough for clear class separation, we inject a same random projection layer with nonlinear activation between each client's pre-trained feature representations and output to enhance linear separability. We conduct experiments on three datasets using pre-trained ResNet18. As shown in Fig 3, feature expansion improves performance across all datasets, with particularly notable gains on the more challenging CIFAR100 dataset. Feature expansion significantly enhances performance but also increases communication overhead (larger d), necessitating a trade-off between the performance and communication overhead.

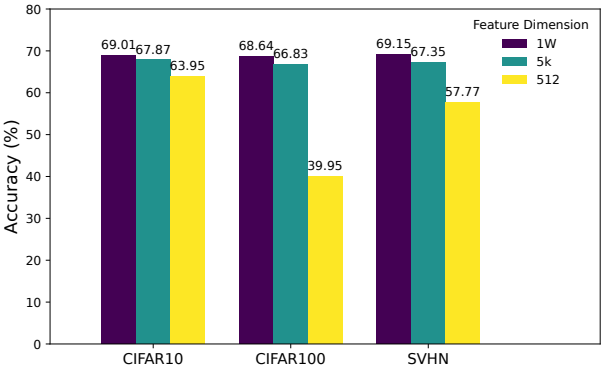

Figure 3: Results of Feature expansion.

Different pre-trained models. Our method relies on the feature extraction capabilities of pre-trained models, so here

we conducted experiments on three datasets using different pre-trained models including ResNet18, ResNet50, MobileNetV2, EfficientNetB0. As shown in Table 4, using pre-trained ResNet50 consistently outperforms others. This indicates that our method could achieve significantly great performance if we own a strong enough pre-trained model.

	CIFAR10	CIFAR100	SVHN
ResNet18	63.95	39.95	57.77
ResNet50	70.32	46.08	60.31
MobilenetV2	57.87	24.39	37.49
EfficientNetB0	56.80	31.20	53.80

Table 4: Evaluation (%) of FedCGS using different pre-trained model.

Comparison with CCVR. We note that CCVR (Luo et al. 2021) shares similar idea as our proposed FedCGS. Specifically, CCVR uploads all clients' local label distribution, class-wise mean and class-wise covariance, and utilizing them to compute the global features distribution statistics in the server. Finally, virtual features are generated using GMMs to retrain the classifier as the final global classifier. However, there are several key differences which we would like to mention. Firstly, in CCVR, each client uploads the class-wise covariance, which causes significant communication overhead. Secondly, the computation process for global statistics in CCVR is incompatible with Secure Aggregation, like ours. Thirdly, CCVR faces a challenge similar to FedPFT (Beitollahi et al. 2024), where variations in sampling strategies and training configurations affect the results.

Discussion

Communication Overhead

Our proposed method only transmits local feature statistics S_i, rather than model updates, to the server. Taking the case that using pre-trained ResNet18 on CIFAR10 as an example, the transmitted parameters of FedAvg/DENSE/Co-Boosting is $|\theta| = 11{,}181{,}642$, FedPFT is $(2d+1)K_gC = 102{,}500$ and ours is $(C+d)\times d + C = 267{,}274$. Our proposed FedCGS achieves higher and more stable accuracy while keeping communication overhead low.

Privacy Discussion

Within the framework of the FedCGS algorithm, each client i uploads statistics S_i as shown in Algorithm 1. Uploading these statistics may arise concerns of potential privacy risk. However, FedCGS only utilizes the aggregated values N^j,A^j,B rather than individual values N_i^j, A_i^j, B_i related with a specific client i, therefore Secure Aggregation (Bonawitz et al. 2017; Mai, Yan, and Pang 2024) can be employed to ensure that the server only receive the aggregated values. Existing one-shot FL methods (Zhang et al. 2022; Dai et al. 2024; Heinbaugh, Luz-Ricca, and Shao 2023; Jhunjhunwala, Wang, and Joshi 2024; Yang et al. 2024a, 2023, 2024b; Beitollahi et al. 2024) all need to utilize the individual values from each client for subsequent steps. For example, Co-Boosting requires the parameters of each client model to build the ensemble, while FedPFT needs the class information from each client for sampling. This means that Secure Aggregation cannot be employed to enhance their privacy-preserving capability.

Considered that applying Secure Aggregation may incur additional communication overhead. We offer two options: one is to perform Secure Aggregation for all the local uploaded variables, and the other is to apply it only to client label counts $\{N_i^j\}_{j=1}^C$ to prevent leakage of the local label distribution, as shown in Algorithm 1. For $\{A_i^j\}_{j=1}^C$ and B_i, it's difficult to recover private data of client i from these variables, as they represent the sum of many features. To empirically analyze it, we utilize the feature inversion technology (Ulyanov, Vedaldi, and Lempitsky 2018). Consider a favorable scenario for the attacker, where client j has only a few samples from the same class. In this case, the server, acting as an attacker, attempts to reconstruct a specific data sample from client j using the uploaded variables. The results in Figure 4 indicate that the uploaded variables cannot be used to recover the client's original data.

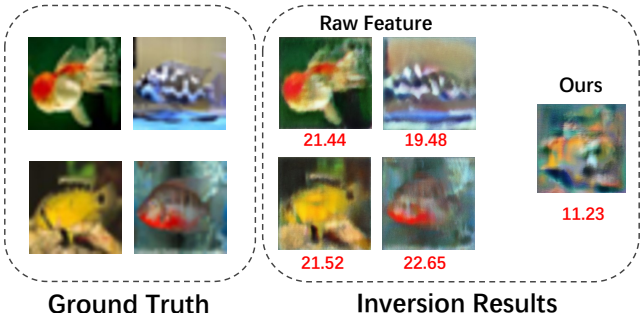

Figure 4: Results of inversion attacks on CIFAR100. Assume a client has only 4 "aquarium fish" samples, as shown in **Ground Truth**. The server attempts to reconstruct a specific data sample from this client. If the server has access to the **Raw Feature** of each sample, the reconstructed results are clear. However, when using our uploaded variables, the results are poor. The PSNR value (red) is displayed below each reconstructed image as a quantitative measure.

Conclusion

We introduce FedCGS, a novel FL framework that leverages pre-trained models to capture global feature statistics. By utilizing these global feature statistics, we employ a parameter-free Naive Bayes classifier instead of a learnable linear classifier, enabling heterogeneity-resistant one-shot FL with competitive performance. Additionally, we propose a personalized one-shot FL approach, where each client uses these statistics as a feature alignment regularizer for local training, facilitated by one additional communication round to download the global statistics. Extensive experiments demonstrate that FedCGS enhances the communication-accuracy trade-off in various scenarios.

Acknowledgements

This work was supported by the Chinese Academy of Sciences under grant No. XDB0690302.

References

Bai, S.; Zhang, J.; Guo, S.; Li, S.; Guo, J.; Hou, J.; Han, T.; and Lu, X. 2024. DiPrompT: Disentangled Prompt Tuning for Multiple Latent Domain Generalization in Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27284–27293.

Beitollahi, M.; Bie, A.; Hemati, S.; Brunswic, L. M.; Li, X.; Chen, X.; and Zhang, G. 2024. Parametric Feature Transfer: One-shot Federated Learning with Foundation Models. *arXiv preprint arXiv:2402.01862*.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwag, V.; Tramer, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, 5253–5270.

Chen, H.; Zhang, Y.; Krompass, D.; Gu, J.; and Tresp, V. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11285–11293.

Chen, H.-Y.; Tu, C.-H.; Li, Z.; Shen, H.-W.; and Chao, W.-L. 2022. On the importance and applicability of pre-training for federated learning. *arXiv preprint arXiv:2206.11488*.

Chen, R.; Wan, Q.; Prakash, P.; Zhang, L.; Yuan, X.; Gong, Y.; Fu, X.; and Pan, M. 2023. Workie-talkie: accelerating federated learning by overlapping computing and communications via contrastive regularization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16999–17009.

Dai, R.; Zhang, Y.; Li, A.; Liu, T.; Yang, X.; and Han, B. 2024. Enhancing One-Shot Federated Learning Through Data and Ensemble Co-Boosting. *arXiv preprint arXiv:2402.15070*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186. Association for Computational Linguistics.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Guan, Z.; Zhou, Y.; Gu, X.; and Li, B. 2024. GIE: Gradient Inversion with Embeddings. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE Computer Society.

Guha, N.; Talwalkar, A.; and Smith, V. 2019. One-shot federated learning. *arXiv preprint arXiv:1902.11175*.

Guo, T.; Guo, S.; and Wang, J. 2023. pFedPrompt: Learning Personalized Prompt for Vision-Language Models in Federated Learning. In *Proceedings of the ACM Web Conference 2023*, 1364–1374.

Guo, T.; Guo, S.; Wang, J.; Tang, X.; and Xu, W. 2023. PromptFL: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*.

Hasan, M.; Zhang, G.; Guo, K.; Chen, X.; and Poupart, P. 2024. Calibrated one round federated learning with bayesian inference in the predictive space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 12313–12321.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Heinbaugh, C. E.; Luz-Ricca, E.; and Shao, H. 2023. Data-Free One-Shot Federated Learning Under Very High Statistical Heterogeneity. In *The Eleventh International Conference on Learning Representations*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7865–7873.

Janson, P.; Zhang, W.; Aljundi, R.; and Elhoseiny, M. 2022. A simple baseline that questions the use of pretrained-models in continual learning. *arXiv preprint arXiv:2210.04428*.

Jhunjhunwala, D.; Wang, S.; and Joshi, G. 2024. FedFisher: Leveraging Fisher Information for One-Shot Federated Learning. In *International Conference on Artificial Intelligence and Statistics*, 1612–1620. PMLR.

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

Krizhevsky, A.; Nair, V.; and Hinton, G. 2009a. CIFAR-100 (Canadian Institute for Advanced Research). http://www.cs.toronto.edu/~kriz/cifar.html.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2009b. Learning Multiple Layers of Features from Tiny Images. *CIFAR-10 (Canadian Institute for Advanced Research)*.

Kurach, K.; Lučić, M.; Zhai, X.; Michalski, M.; and Gelly, S. 2019. A large-scale study on regularization and normalization in GANs. In *International conference on machine learning*, 3581–3590. PMLR.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Li, H.; Huang, W.; Wang, J.; and Shi, Y. 2024. Global and Local Prompts Cooperation via Optimal Transport for Federated Learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. 2020a. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.

Li, Q.; He, B.; and Song, D. 2021. Practical One-Shot Federated Learning for Cross-Silo Setting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020b. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3): 50–60.

Lin, T.; Kong, L.; Stich, S. U.; and Jaggi, M. 2020. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363.

Liu, X.; Liu, L.; Ye, F.; Shen, Y.; Li, X.; Jiang, L.; and Li, J. 2023. FedLPA: Personalized One-shot Federated Learning with Layer-Wise Posterior Aggregation. *arXiv preprint arXiv:2310.00339*.

Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34: 5972–5984.

Mai, P.; Yan, R.; and Pang, Y. 2024. Rflpa: A robust federated learning framework against poisoning attacks with secure aggregation. *arXiv preprint arXiv:2405.15182*.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.

Mothukuri, V.; Parizi, R. M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; and Srivastava, G. 2021. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115: 619–640.

Neiswanger, W.; Wang, C.; and Xing, E. 2013. Asymptotically exact, embarrassingly parallel MCMC. *arXiv preprint arXiv:1311.4780*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Nguyen, D. P.; Munoz, J. P.; and Jannesari, A. 2024. Flora: Enhancing vision-language models with parameter-efficient federated learning. *arXiv preprint arXiv:2404.15182*.

Nguyen, J.; Wang, J.; Malik, K.; Sanjabi, M.; and Rabbat, M. 2022. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*.

Rao, B.; Zhang, J.; Wu, D.; Zhu, C.; Sun, X.; and Chen, B. 2024. Privacy inference attack and defense in centralized and federated learning: A comprehensive survey. *IEEE Transactions on Artificial Intelligence*.

Sahu, A. K.; Li, T.; Sanjabi, M.; Zaheer, M.; Talwalkar, A.; and Smith, V. 2019. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Machine Learning and Systems (MLSys) Conference*.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Tresp, V. 2000. A Bayesian committee machine. *Neural computation*, 12(11): 2719–2741.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9446–9454.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.

Wu, X.; Liu, X.; Niu, J.; Wang, H.; Tang, S.; and Zhu, G. 2024. FedLoRA: When Personalized Federated Learning Meets Low-Rank Adaptation.

Xu, J.; Tong, X.; and Huang, S.-L. 2023. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*.

Yang, M.; Su, S.; Li, B.; and Xue, X. 2023. One-Shot Federated Learning with Classifier-Guided Diffusion Models. *arXiv preprint arXiv:2311.08870*.

Yang, M.; Su, S.; Li, B.; and Xue, X. 2024a. Exploring One-Shot Semi-supervised Federated Learning with Pre-trained Diffusion Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16325–16333.

Yang, M.; Su, S.; Li, B.; and Xue, X. 2024b. FedDEO: Description-Enhanced One-Shot Federated Learning with Diffusion Models. *arXiv preprint arXiv:2407.19953*.

Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; and Parizi, R. M. 2024. A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*.

Zhang, J.; Chen, C.; Li, B.; Lyu, L.; Wu, S.; Ding, S.; Shen, C.; and Wu, C. 2022. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35: 21414–21428.

Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; and Alvarez, J. M. 2020. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*.

Zhang, R.; Xu, Q.; Yao, J.; Zhang, Y.; Tian, Q.; and Wang, Y. 2023. Federated domain generalization with generalization adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3954–3963.

Zhou, Y.; Pu, G.; Ma, X.; Li, X.; and Wu, D. 2020. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*.