arXiv:2403.19499v1 [cs.LG] 28 Mar 2024

CLIENT-SUPERVISED FEDERATED LEARNING: TOWARDS ONE-MODEL-FOR-ALL PERSONALIZATION

Peng Yan, Guodong Long

ABSTRACT

Personalized Federated Learning (PerFL) is a new machine learning paradigm that delivers personalized models for diverse clients under federated learning settings. Most PerFL methods require extra learning processes on a client to adapt a globally shared model to the client-specific personalized model using its own local data. However, the model adaptation process in PerFL is still an open challenge in the stage of model deployment and test time. This work tackles the challenge by proposing a novel federated learning framework to learn only one robust global model to achieve competitive performance to those personalized models on unseen/test clients in the FL system. Specifically, we design a new **Client-Supervised Federated Learning (FedCS)** to unravel clients' bias on instances' latent representations so that the global model can learn both client-specific and client-agnostic knowledge. Experimental study shows that the FedCS can learn a robust FL global model for the changing data distributions of unseen/test clients. The FedCS's global model can be directly deployed to the test clients while achieving comparable performance to other personalized FL methods that require model adaptation.

Index Terms— Federated Learning

1. INTRODUCTION

Modern machine learning relies on massive data to train models like deep neural networks (DNNs), but collecting data is becoming more sensitive with increasing attention to privacy protection. Then, federated learning (FL) [1] emerged and became a popular learning paradigm in recent years. The FL aims to coordinate a set of clients (i.e., devices) to train a global model while preserving their data locally and privately. Further, personalized FL (PerFL) [2, 3, 4] balances cross-client training and personalization. Clients will collaborate as in vanilla FL and leverage client-specific properties to learn personalized models demonstrating superior performance on non-IID data.

Existing personalized federated learning research usually aims to learn many client-specific personalized models to tackle the non-IID data in federated learning systems. Specifically, a global model will be learned to grasp shared knowledge, and then many personalized models will be learned or fine-tuned on the client by leveraging the global model and local dataset. Although this client-specific personalised model strategy has the potential to catch non-IID in fine-grained, the on-device learning and fine-tuning process is usually difficult to control in practice, especially in the stage of model deployment and test time.

This paper aims to rethink the personalized federated learning problem by proposing a one-model-for-all strategy to embody personalization in federated settings. One motivation is that model personalization in most PerFL methods relies on the bias of instances on the same client, and little supervised information describing the client is introduced. Then, the global model trained on the same data will attain the same performance if it can recognize the bias of a client. Inspired by this, we propose to use a one-model-for-all strategy to learn a unified model that can be shared across clients in the FL system. Moreover, the client-specific information will be encoded in a unified representation space and then be fed into a decision module along with the client-agnostic knowledge to make the final prediction.

Based on the thought above, this work proposes a novel **Client-Supervised Federated Learning (FedCS)** that is to learn a unified global model with the below functions, including 1) to learn a unified representation space that can encode the bias of a client, 2) to share client-agnostic knowledge as vanilla FL methods, and 3) to make personalized predictions by leveraging both pieces of information. Moreover, the **Representation Alignment (RA)** mechanism in FedCS could become a plug-in component to be integrated with any federated learning methods. It enables a vanilla FL model to output personalized results without on-device fine-tuning steps. An illustration of the FedCS is in Fig.1.

To learn the above objectives in a federated optimization framework, we designed a novel **Client-Supervised Federated Optimization Framework** to align the objective representation space while being consistent with the federated optimization framework. It exploits the bias that instances on the same client are influenced by identical client properties and formulates the representation aligning task into an optimization problem that clients can solve collaboratively.

Through qualitative and quantitative experiments, we illustrate how FedCS is integrated into a black-box model to achieve compared performance of other personalized federated learning methods. Moreover, we demonstrate that FedCS

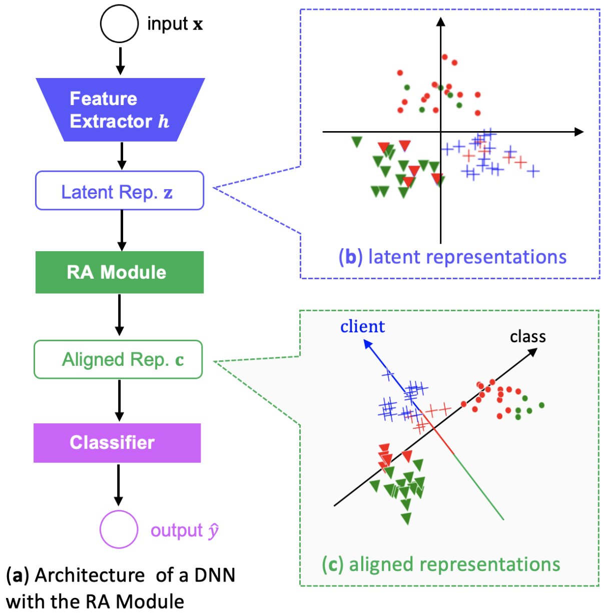

input $\mathbf{x}$
Feature Extractor h
Latent Rep. $\mathbf{z}$
RA Module
Aligned Rep. $\mathbf{c}$
Classifier
output $\hat{y}$
(a) Architecture of a DNN with the RA Module
(b) latent representations
client
class
(c) aligned representations

Fig. 1. Illustration to the FedCS. (b) and (c) describe distributions of instances' latent representations before and after the Representation Alignment module (RA Module). Types of markers denote classes, and colors indicate clients they are observed. The vanilla feature extractor is unable to recognize clients' bias. RA module in FedCS will align the hidden space so that latent representations can indicate biases of clients.

is compatible with most FL models. Vanilla models with an RA module can achieve competitive performance compared to those ad hoc PerFL models while not need extra fine-tuning steps or personal parameters.

The main contributions are summarized as follows:

- We propose a novel one-model-for-all personalized FL framework that won't require an extra fine-tuning process at the stage of model deployment and test time. The personalization in the FL system is carried on representations indicating client bias rather than models.
- We designed a novel **representation alignment** mechanism to project instances' representation into a space indicating clients' biases. The following decision layers in the neural architecture can automatically learn to make personalized predictions by leveraging the client's bias.
- A **client-supervised optimization framework** is designed to fit the proposed framework. It formulates the representation-aligning problem into a unified optimization framework that clients can solve collaboratively under FL settings.
- Contrast with baseline methods shows that, by integrating FedCS into vanilla FL models, they can achieve competitive personalization performance without requiring extra fine-tuning steps or personal parameters.

2. RELATED WORK

Federated Learning is an emerging machine learning paradigm where many clients collaborate to train a global model by sharing and aggregating model parameters rather than user data [1]. Further work [2, 3, 4] shows that the global model performs better after tuning toward client-specific properties and proposes **Personalized Federated Learning**.

A popular way to personalize is to fine-tune the global model on a client's local data [1, 5, 6]; meta-learning strategies will help improve the tuning process [7, 8]. Meanwhile, [9] trains a local model for each client while constraining the distributed training procedures with a shared regularizer. Recent work [10] studies partial personalization, splitting an FL model into global and local parts. Personalization is fulfilled by individually training the local part of a model on clients. [11] is a simple but efficient case of these methods. It trains a model through the vanilla FedAvg [1] except for preserving batch-normalization modules locally. [12] introduces a method to learn a shared data representation across clients and unique classification heads for each client. [13] utilizes a global and a local encoder to learn different representations for cross-client collaboration and personalization. [14] and [15] propose to learn unique representations for client properties to improve personalization on new clients. However, these methods require extra training data to capture those clients' properties when deployed on them, which is only sometimes feasible.

3. PROBLEM FORMULATION

Federated Learning (FL) assumes K clients participate in a learning process with $\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, ..., \mathbf{x}_{N_i}^{(i)}\} \in \mathcal{X}^{(i)}$ denoting instances on the i-th client, and $\{y_1^{(i)}, y_2^{(i)}, ..., y_{N_i}^{(i)}\} \in \mathcal{Y}^{(i)}$ are their labels. The FL task is to find the optimal parameters ω^* for a global model $f(\mathbf{x};\omega)$ by minimising the total loss of all clients as the following optimization problem:

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{K} \alpha_i \mathcal{L}_i(\omega) \tag{1}$$

where $\mathcal{L}_i(\omega) = (1/N_i) \sum_{j=1}^{N_i} l(f(\mathbf{x}_j^{(i)};\omega), y_j^{(i)})$ is the supervised loss on the i-th client, and α_i is its weight. In particular, in the standard FL framework [1], α_i is the fraction of the size of the client's training data, i.e., $\alpha_i = N_i / \sum_{i'=1}^{K} N_{i'}$. Each client will update ω privately by minimizing $\mathcal{L}_i(\omega)$, and a server will orchestrate the distributed training process by collecting locally updated ω and synchronizing the averaged one.

Personalized Federated Learning (PerFL) leverages cross-client collaboration but learns a personalized model $f(\mathbf{x}; \omega, \mu_i)$ for each client, where ω denotes parameters shared, and μ_i denotes parameters for the i-th client. The learning task can be formulated into a unified optimization problem [10]:

$$\omega^*, \{\mu_i^*\}_{i=1}^K = \arg \min_{\omega, \{\mu_i\}_{i=1}^K} \sum_{i=1}^K \alpha_i \mathcal{L}_i(\omega, \mu_i) \tag{2}$$

There are different types of personalization according to how to define ω and μ_i. 1) **full personalization**: a local model is fully parameterized by μ_i, and the global model guides the local training process, e.g., regularizing μ_i by minimizing $\|\omega - \mu_i\|_2$ [9] or initializing local parameters with $\mu_i = \omega - \nabla \mathcal{L}_i$ [8]; 2) **partial personalization**: ω and μ_i constitute the global and personal parts of a local model, where ω is shared through the fundamental FL method in 1, and μ_i is trained individually on each client. e.g., ω could be parameters of a shared backbone model, and μ_i is a classification head for the i-th client [12].

4. METHODOLOGY

Looking inside the latent space of a DNN $f(\mathbf{x}; \omega) = g(h(\mathbf{x}; \omega_h); \omega_g)$, it consists of two parts: $h(\mathbf{x}; \omega_h)$ is a feature extractor learning the latent representation $\mathbf{z} \in \mathbb{R}^d$, and $g(\mathbf{z}; \omega_g)$ is a classification head making predictions based on $\mathbf{z}$. Our goal is to align the latent representation space for $\mathbf{z}$, such that 1) it is able to embed client-specific information, i.e., instances from similar clients will have similar values and vary significantly otherwise; 2) it is able to share client-agnostic knowledge as vanilla FL methods, and 3) downstream modules can make personalized predictions by leveraging both information.

Formally, FedCS looks for a projection $\mathbf{c} = \mathbf{P}^T \mathbf{z}$, where $\mathbf{P}_{d \times r} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_r]$ is the orthonormal basis of the objective representation space. It searches for the optimal directions $\mathbf{P}^*$ according to an inductive bias that representation $\mathbf{c}$ on the same client shall be similar, and those from different clients are on the contrary.

4.1. Representation Alignment

Specifically, let $\bar{\mathbf{c}}^{(i)}$ denote the mean of representations on the i-th client, and $\bar{\mathbf{c}}$ be the global mean among clients.

$$\Sigma_W = \frac{1}{\sum_{i=1}^K N_i} \sum_{i=1}^K \sum_{j=1}^{N_i} (\mathbf{c}_j^{(i)} - \bar{\mathbf{c}}^{(i)})(\mathbf{c}_j^{(i)} - \bar{\mathbf{c}}^{(i)})^T \tag{3}$$

Eq.3 is the within-client scatter matrix that measures the scatter of latent representations within each client and

$$\Sigma_B = \frac{1}{\sum_{i=1}^K N_i} \sum_{i=1}^K N_i (\bar{\mathbf{c}}^{(i)} - \bar{\mathbf{c}})(\bar{\mathbf{c}}^{(i)} - \bar{\mathbf{c}})^T \tag{4}$$

Eq.4 is the between-client scatter matrix that measures the scatters of the mean across clients. To find the $\mathbf{P}^*$ is to find the directions that minimize Σ_W and maximize Σ_B. For example, it can be formulated as the Linear Discriminate Analysis (LDA) problem below

$$\mathbf{P}^* = \arg \max_{\mathbf{P}} J(\mathbf{P}) = \arg \max_{\mathbf{P}} \mathrm{Tr}(\Sigma_W^{-1} \Sigma_B) \tag{5}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of the matrix.

Then, bring Eq.5 into the FL framework, the overall learning task is formulated as a bi-level optimization problem

$$\begin{aligned} \omega_h^*, \omega_g^* &= \arg \min_{\omega_h, \omega_g} \sum_{i=1}^K \alpha_i \mathcal{L}_i(\omega_h, \omega_g) \\ s.t.\ \mathbf{P}^* &= \arg \max_{\mathbf{P}} J(\mathbf{P}) \end{aligned} \tag{6}$$

where

$$\mathcal{L}_i = \sum_{j=1}^{N_i} l(g(\mathbf{P}^T h(\mathbf{x}_j^{(i)}; \omega_h); \omega_g), y_j^{(i)}) \tag{7}$$

In the next section, we introduce a client-supervised method to optimize the Eq.6 under the FL setting.

4.2. Client Supervised Optimization

Theoretically, the optima of Eq.5 are eigenvectors of $\Sigma_W^{-1} \Sigma_B$ associated with the r largest eigenvalues [16]. Then, the classification and the alignment tasks in Eq.6 can be optimized alternatively under the conventional FL framework [9, 10]. However, decomposing $\Sigma_W^{-1} \Sigma_B$ is computationally expensive, and involves collecting the local mean $\bar{\mathbf{c}}^{(i)}$ which is privacy sensitive. To this end, we propose a client-supervised method that decomposes the learning task in Eq.5 into subtasks so that clients can optimize it collaboratively.

Concretely, previous works [17] show that maximizing Eq.5 is equivalent to maximizing $\Sigma_W^{-1/2} \Phi \Sigma_W^{-1/2}$, where Φ is an approximation to the eigen system of the global correlation matrix $\Sigma_W + \Sigma_B$, and both $\Sigma_W^{-1/2}$ and Φ can be updated incrementally through the following equations

$$\Sigma_W^{-1/2} = \Sigma_W^{-1/2} + \eta * (I - \Sigma_W^{-1/2} \Sigma_W \Sigma_W^{-1/2}) \tag{8}$$

and

$$\Phi = \Phi + \lambda * (\bar{\mathbf{u}} \bar{\mathbf{u}}^T \Phi - \Phi \tau (\Phi \bar{\mathbf{u}} \bar{\mathbf{u}}^T \Phi) \tag{9}$$

where $\mathbf{u} = \Sigma_W^{-1/2} \bar{\mathbf{c}}$, and $\tau(\cdot)$ is an operator that sets all the elements below the main diagonal of the matrix to zero. In this process, Σ_W summarizes the correlation of instances on each client and hence will work as supervised information to encode the bias of a client into $\mathbf{u}$. Details of the derivations of Eq.8 and Eq.9 are discussed in Appendix. Then, the representation alignment process is described in Algorithm.1, and the overall FL process with FedCS is described in Algorithm.2.

Algorithm 1 Representation Alignment

```
Input: a batch of latent representations $\mathbf{z}$, global mean $\mathbf{z}_g$,
client's local correlation $\Sigma_W^{-1/2}$ and $\Phi$
begin:
1. if $\Sigma_W^{-1/2}$ and $\Phi$ are empty, initialize $\Sigma_W^{-1/2}$ and $\Phi$
2. calculate local mean: $\mathbf{z}_l = mean(\mathbf{z})$
3. update global mean: $\mathbf{z}_g = \mathbf{z}_g + \frac{1}{|\mathbf{z}|}(\mathbf{z} - \mathbf{z}_g)$
4. calculate local correlations: $\Sigma_W = (\mathbf{z} - \mathbf{z}_l)(\mathbf{z} - \mathbf{z}_l)^T$
5. update $\Sigma_W^{-1/2}$ by Eq.8
6. update $\Phi$ by Eq.9
7. $\mathbf{P} = \Sigma_W^{-1/2}\Phi$
8. return $\mathbf{P}$
end
```

Algorithm 2 Client-Supervised Federate Learning

```
Input: training data $(\mathcal{X}^{(i)}, \mathcal{Y}^{(i)})$ distributed on $K$ clients,
$t$ the round to update $P$.
begin:
1. Initialize the FL model $f(\mathbf{x}; \omega) = g(h(\mathbf{x}; \omega_h); \omega_g)$.
2. Select a set of clients $\mathbb{S}$
for the $i$-th client in $\mathbb{S}$ parallel do
  update $\omega_h$ and $\omega_g$ locally
  if round%$t$ == 0:
  update $\mathbf{P}$ locally by Algorithm.2
end for
3. Aggregate local updates of $\omega_h$, $\omega_g$ and $\mathbf{P}$ by averaging
end
```

5. EXPERIMENTS

In this section, we demonstrate the advantages of FedCS in learning from clients with non-i.i.d. data. The FedCS can learn a robust FL global model for the changing data distributions of unseen/test clients. The FedCS's global model can be directly deployed to the test clients while achieving comparable performance to other personalized FL methods that require model adaptation.

5.1. Client Settings

We simulate FL environments by allocating instances from benchmark datasets to 50 clients, and two types of heterogeneity are applied (Details of client settings are introduced in the Appendix).

- **Label-shift**: We experiment, respectively, on the MNIST and the CIFAR-10 datasets. We allocate instances of each label individually according to a posterior of the Dirichlet distribution[18], which divides clients into ten groups with different label distributions. Eight groups of clients will participate in the collaborative training process, and the rest will be held for test.
- **Feature-shift**: We experiment on the Digit-5 dataset to evaluate FedCS's performance on feature-shift data. The Digit-5 consists of digits from five domains (MNIST, MNIST-M, SVHN, USPS and Synth Digits). We assign instances of each domain to nine clients, where eight clients will train the global model and one for the test. In addition, we randomly draw instances from all domains to compose five mixed datasets for the test.

5.2. Models and Hyperparameters

We apply convolution neural networks (CNN) as fundamental models and integrate our proposed RA module into fully connected layers (FCs) to align their hidden layers (see Appendix for details of model architectures). By default, in each communication round, we sample ten clients to update the global model and evaluate the global model's performance on all clients. One epoch of fine-tuning steps will be applied for benchmark methods where the global model must be adapted to a client's local data before testing. The learning rate of a client's local training step is initialized as 0.005, and it will decay at the rate of 0.8 every 50 communication rounds. The RA module will be updated every five communication rounds by the sampled ten clients, and the learning rate is fixed at 0.001. (see Appendix for details of other hyperparameters).

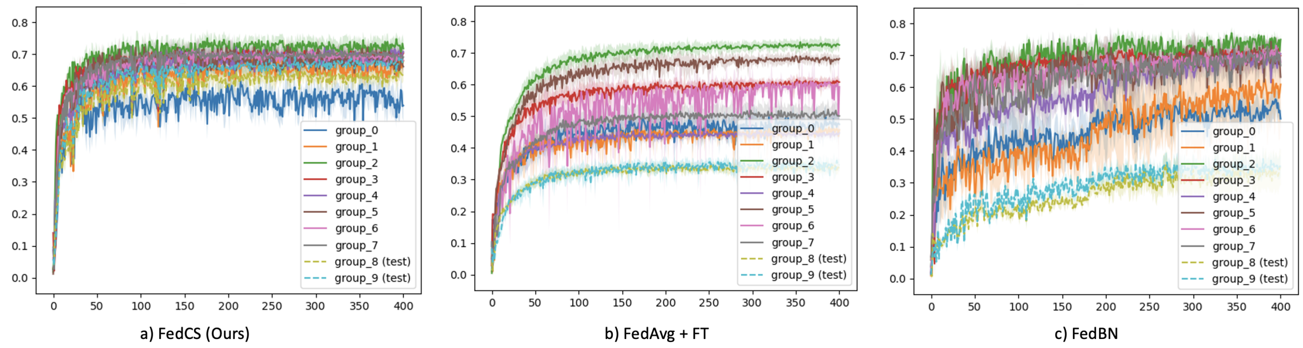
5.3. Performance

We first demonstrate averaged model performance on all clients, which shows that a global model learned with FedCS will achieve comparable performance to other personalized FL methods that require model adaptation. Then, we look inside the group-wised metrics to evaluate a model's performance on different distributions. The global model learned with FedCS is more robust to different distributions. It can be directly deployed on test clients without extra adaptation. Several FL strategies are compared as baselines: 1) FedAvg+FT [5]; 2) FedAvg+BN; 3) FedBN [11]. 4) FedRep [12]; 5) PerFL [10]; 6) In addition, we demonstrate the performance of models those trained on each client locally (Local Only).

5.4. Label-shift Settings

5.4.1. Overall Performance

For label-shift settings, the weighted AUC score and the weighted F1 score are applied to evaluate model performance on data with unbalanced label distributions. Table.1 demonstrates models' performance on the MNIST dataset. We can find that a model with FedCS layers achieves the best performance under the label-shift setting. It outperforms those locally fine-tuned global models (FedAvg+FT) and models with client-specific parameters (FedBN). FedRep has the worst



a) FedCS (Ours)

b) FedAvg + FT

c) FedBN

Fig. 2. Part of experiments on CIFAR-10. The horizontal axis denotes communication rounds, and the vertical axis denotes F1 Scores. We demonstrate the averaged weighted F1-scores within each client group and marked them with different colours. We can find the performance on different distributions (client groups) vary significantly through FedAvg+FT and FedBN. FedCS has the most robust performance even on unseen clients (group 8 and 9)

Table 1. The averaged performance on the MNIST dataset. The standard deviation of the metric between clients is reported in parentheses. w. is the abbreviation of 'weighted', and the ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted.

	w. AUC (10^{-2}) ↑	w. F1 (10^{-2}) ↑
Local Only	85.18(4.62)	47.80(10.12)
FedAvg+FT	96.97(2.54)	80.81(10.14)
FedAvg+BN	99.69(0.14)	93.38(1.76)
FedBN	99.32(0.75)	83.85(18.82)
FedRep	75.72(15.56)	38.53(20.52)
PerFL	99.48(0.20)	91.40(1.95)
FedCS-FC1(ours)	**99.72(0.15)**	**93.72(1.71)**
FedCS-FC2(ours)	99.71(0.16)	93.43(1.78)
FedCS-FC3(ours)	99.72(0.26)	93.64(1.56)

Table 2. The averaged performance on the CIFAR-10 dataset. The standard deviation of the metric between clients is reported in parentheses. w. is the abbreviation of 'weighted', and the ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted.

	w. AUC (10^{-2})↑	w. F1 (10^{-2})↑
Local Only	70.93(9.23)	34.86(10.71)
FedAvg+FT	86.89(4.65)	51.75(12.62)
FedAvg+BN	91.57(4.30)	57.87(14.95)
FedBN	91.64(4.56)	59.41(14.70)
FedRep	70.70(9.80)	33.01(13.88)
PerFL	89.18(3.36)	58.62(9.17)
FedCS-FC1(ours)	**93.72(1.71)**	69.48(4.69)
FedCS-FC2(ours)	93.33(2.33)	69.06(5.97)
FedCS-FC3(ours)	93.49(2.09)	**69.69(4.96)**

performance, which may result from the lack of training data and the unbalanced label distribution of each client.

Table.2 demonstrates models' performance on the CIFAR-10 dataset. FedCS has the best performance under this setting. Other models are less effective than FedCS, and their performances vary significantly among clients (higher standard deviations). We will show that the gap results from the generalization error on test clients, and there is no such problem for the proposed FedCS.

5.4.2. *Group-wised Performance*

Fig.2 demonstrates averaged weighted F1-scores within each client group[1]. We can find that the global model with FedCR is more robust among different clients, even if they are from test groups (groups 8-9). Fine-tuned models (FedAvg+FT) and models with personalized parameters (FedBN) have significant performance gaps when deployed on training clients (group 0-7) and test clients. They achieve higher F1 scores in training groups but could be less effective in test groups.

5.5. Feature-shift Settings

In this section, we demonstrate evaluations of feature-shifted data. According to Table.3, we show that FedCS achieves the highest weighted AUC and weighted-F1 score. The averaged accuracy within each data domain is shown in the Appendix. It also validates our claims that FedCS performs more robustly on all domains while other methods degenerate significantly on test clients.

[1] Full version of results are shown in Appendix.

Table 3. The averaged accuracy on the Digit-5 dataset. The standard deviation of the metric between clients is reported in parentheses. w. is the abbreviation of 'weighted', and the ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted.

	w. AUC (10^{-2})↑	w. F1 (10^{-2})↑
Local Only	84.03(9.75)	49.73(20.38)
FedAvg+FT	96.11(1.98)	75.64(4.94)
FedAvg+BN	97.82(3.11)	83.40(12.96)
FedBN	95.71(3.79)	74.07(10.90)
FedRep	82.94(11.54)	50.14(20.88)
PerFL	96.10(2.47)	74.63(6.71)
FedCS-FC1(ours)	**98.74(0.95)**	**87.89(4.76)**
FedCS-FC2(ours)	98.60(1.09)	87.66(5.15)
FedCS-FC3(ours)	98.57(1.09)	87.66(5.19)

6. CONCLUSION

This paper is the first to propose using a one-model-for-all strategy to implement personalized federated learning. We believe the one-model-for-all personalization can form a new topic to advance existing personalized federated learning research. It is foreseeing more discussion and exploration can be conducted in this new one-model-for-all personalized federated setting.

7. REFERENCES

[1] 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*. PMLR, 2017, pp. 1273–1282.

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

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

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

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

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai, "Fedavg with fine tuning: Local updates lead to representation learning," *arXiv preprint arXiv:2205.13692*, 2022.

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

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

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

[10] Krishna Pillutla, Kshitiz Malik, Abdelrahman Mohamed, Michael Rabbat, Maziar Sanjabi, and Lin Xiao, "Federated learning with partial model personalization," *arXiv preprint arXiv:2204.03809*, 2022.

[11] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[12] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai, "Exploiting shared representations for personalized federated learning," *arXiv preprint arXiv:2102.07078*, 2021.

[13] Zhengquan Luo, Yunlong Wang, Zilei Wang, Zhenan Sun, and Tieniu Tan, "Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring," *arXiv preprint arXiv:2206.06818*, 2022.

[14] Tiandi Ye, Cen Chen, Yinggui Wang, Xiang Li, and Ming Gao, "Upfl: Unsupervised personalized federated learning towards new clients," 2023.

[15] Yuwei Sun, Ng Chong, and Hideya Ochiai, "Feature distribution matching for federated domain generalization," 2022.

[16] Richard O Duda, Peter E Hart, et al., *Pattern classification*, John Wiley & Sons, 2006.

[17] Youness Aliyari Ghassabeh, Frank Rudzicz, and Hamid Abrishami Moghaddam, "Fast incremental lda feature extraction," *Pattern Recognition*, vol. 48, no. 6, pp. 1999–2012, 2015.

[18] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019.

[19] C. Chatterjee and V.P. Roychowdhury, "On self-organizing algorithms and networks for class-separability features," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 663–678, 1997.

A. THEORY BACKGROUNDS

We introduce an inductive bias to align the hidden layers of a DNN so that it is able to learn client bias to achieve personalization without on-device fine-tuning. We assume data on the same client are influenced by the same client properties so that data from the similar clients will have similar representations. We formulate the representation alignment problem into an optimization described in Eq.5.

Eq.5 is equivalent to the objective function of Linear Discriminant Analysis (LDA) whose solution is the enginvector of corresponding to the largest enginvalue of $\Sigma_W^{-1}\Sigma_B$. However, the computation cost of the decomposing $\Sigma_W^{-1}\Sigma_B$ would be high, and aggregating local representations to a server is infeasible in FL. Therefore, we divide the matrix decomposition process into a set of clients' local updating steps and integrate it into standard FL framework.

A.1. Client Supervised Optimization

Let Σ_G denote the correlation matrix of latent representations on all clients, there is $\Sigma_G = \Sigma_W + \Sigma_B$. Then the learning problem can be formulate as solving the following eigenvalue problem

$$\Sigma_W^{-1}\Sigma_G P^* = P^*\Lambda \tag{10}$$

where Λ is the diagonal eigenvalue matrix of $\Sigma_W^{-1}\Sigma_G$. [17] shows solving Eq.10 can be simplified into solving the following symmetric eigenvalue problem:

$$\Sigma_W^{-1/2}\Sigma_G\Sigma_W^{-1/2}\Phi = \Phi\Lambda \tag{11}$$

where Φ denotes eignevectors of $\Sigma_W^{-1/2}\Sigma_G\Sigma_W^{-1/2}$, and there is $P^* = \Sigma_W^{-1/2}\Phi$. To find the optimum P^* is to find Φ and $\Sigma_W^{-1/2}$.

[19] introduced an incremental algorithm to optimize Φ and $\Sigma_W^{-1/2}$ through which we can distribute the optimizing steps to clients. Concretely, it proves that, 1)

$$\Phi_{k+1} = \Phi_k + \lambda(\mathbf{z}_k\mathbf{z}_k^T\Phi_k - \Phi_k\tau[\Phi_k^T\mathbf{z}_k\mathbf{z}_k^T\Phi_k]) \tag{12}$$

will converge to the enginvector matrix Φ when there are sufficient instance $\mathbf{z}_k$ sampled from the data distribution; 2). let $\mathbf{S}$ denotes $\Sigma_W^{-1/2}$, then

$$S_{k+1} = S_k + \eta * (I - S_k\Sigma_W S_k) \tag{13}$$

where S_{k+1} will converge to the inverted square root of Σ_W when 1) S_0 is initialized as a symmetric positive define matrix, and 2) there are sufficient instance $\mathbf{z}_k$ sampled from the data distribution.

We apply the above methods in Eq.9 and Eq.8 to update P on each clients individually and aggregate local updates to align axis on different clients.

A.2. Discussion on Privacy Protection

According to the section above, clients requires to share local correlations to update the matrix Σ_W. However, Σ_W is a global statistic where a client's local bias would be eliminated, privacy-protection methods like differential privacy are feasible to avoid privacy leakage.

B. EXPERIMENTS

B.1. Heterogeneity Settings

Three benchmark datasets are applied to evaluate FedCR's performance.

B.1.1. Label-shift

We allocate instances of each label individually according to a posterior of the Dirichlet distribution[18], which divides clients into ten groups with different label distributions. Eight groups of clients will participate in the collaborative training process, and the rest will be held for testing. An example of the client setting is in Fig.3. Distributions of the number of instances of each class are show in Fig.4

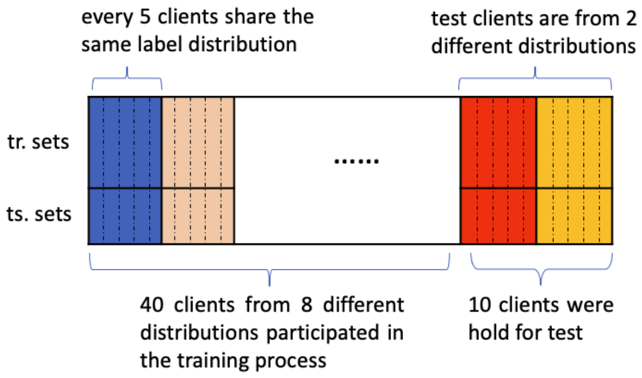

every 5 clients share the same label distribution
test clients are from 2 different distributions
tr. sets
ts. sets
......
40 clients from 8 different distributions participated in the training process
10 clients were hold for test

Fig. 3. client settings for label shift experiments

B.1.2. Feature-shift

We experiment on the Digit-5 dataset to evaluate FedCRR's performance on feature-shift data. The Digit-5 consists of digits from five different domains (MNIST, MNIST-M, SVHN, USPS and Synth Digits). We assign instances of each domain to nine clients, where eight clients will train the global model and one for the test. In addition, we randomly draw instances from all domains to compose five mixed datasets for the rest clients for the test. The distributions on clients are shown in Fig.5

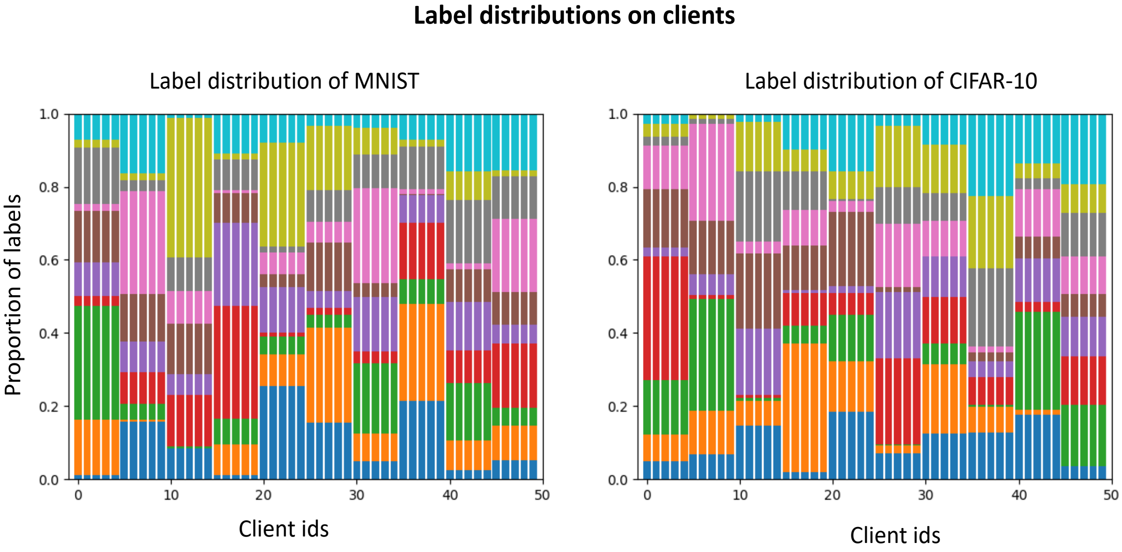


Fig. 4. Proportion of instances of different classes. Different classes are marked with different colors. Horizontal axix denotes client ids, and the vertical axis denotes the proportion of classes

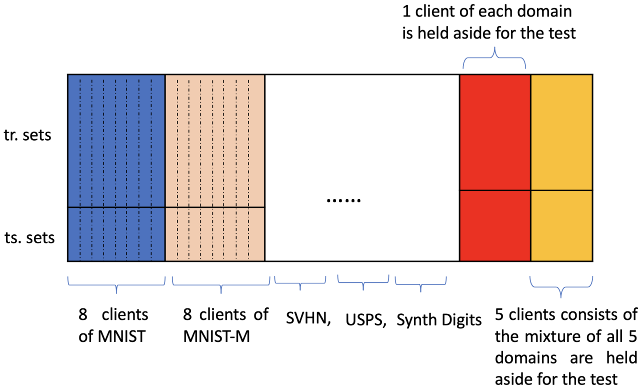

1 client of each domain is held aside for the test
tr. sets
ts. sets
......
8 clients of MNIST
8 clients of MNIST-M
SVHN,
USPS,
Synth Digits
5 clients consists of the mixture of all 5 domains are held aside for the test

Fig. 5. client settings for feature shift experiments

B.2. Metrics

We apply three different metrics to evaluate a model's performance on each clients. They are: accuracy, weighted F1 score and weighted AUC. We apply implementations in scikit-learn[2] in our experiments and utilize Laplacian smooth for clients where some labels are missing.

B.3. Model Architecture

Please refers attached codes for details of our experiments.

B.4. Supplementary Results

In addition to the results in Sec.5, more results are demonstrated as below. We can find our methods achieve better robustness and performance, especially on unseen/test clients.

[2] https://scikit-learn.org/stable/index.html

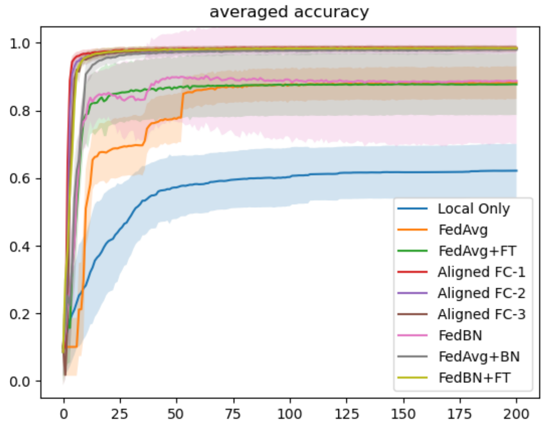


Fig. 6. Accuracy on the MNIST dataset. y-axis denotes the averaged accuracy on all clients and x-axis denotes the communication round. Shades denote the standard deviation of accuracy among clients. Our FedCS (Aligned FC-1 to Aligned FC3) achieves the best and the most robust performance.

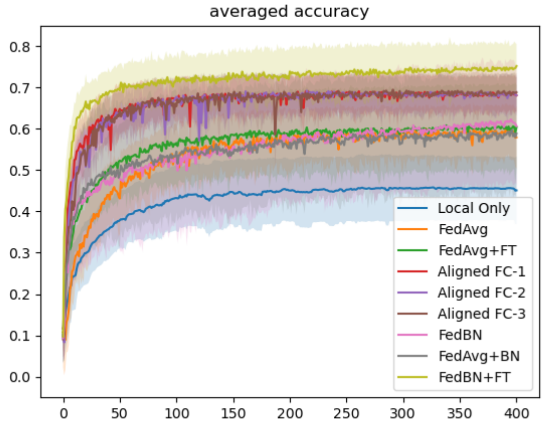


Fig. 7. Accuracy on the CIFAR-10 dataset. y-axis denotes the averaged accuracy on all clients and x-axis denotes the communication round. Shades denote the standard deviation of accuracy among clients. Our FedCS (Aligned FC-1 to Aligned FC3) achieves the best and the most robust performance.

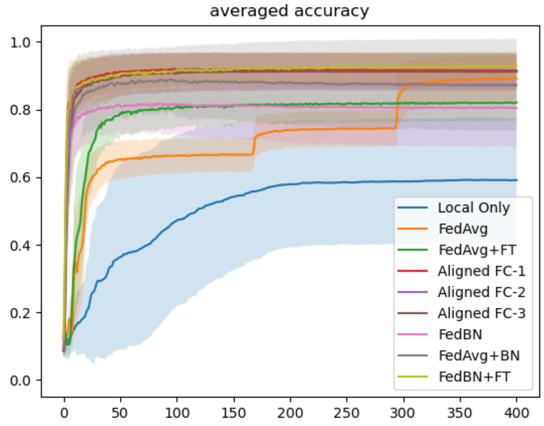


Fig. 8. Accuracy on the Digit-5 dataset. y-axis denotes the averaged accuracy on all clients and x-axis denotes the communication round. Shades denote the standard deviation of accuracy among clients. Our FedCS (Aligned FC-1 to Aligned FC3) achieves the best and the most robust performance.

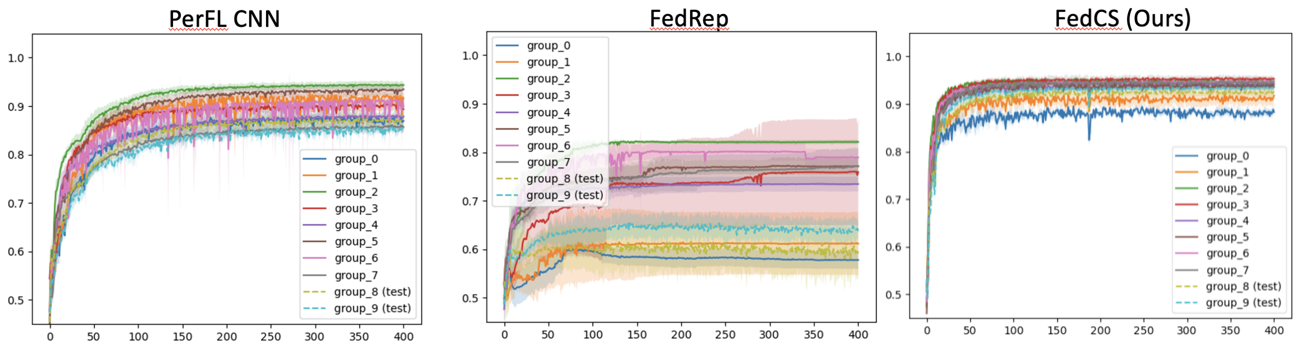


Fig. 9. Part of experiments on CIFAR-10. The horizontal axis denotes communication rounds, and the vertical axis denotes accuracy. We demonstrate the averaged accuracy within each client group and marked them with different colours. We can find the performance on different distributions (client groups) vary significantly through FedAvg+FT and FedBN. FedCS has the most robust performance even on unseen clients (group 8 and 9)

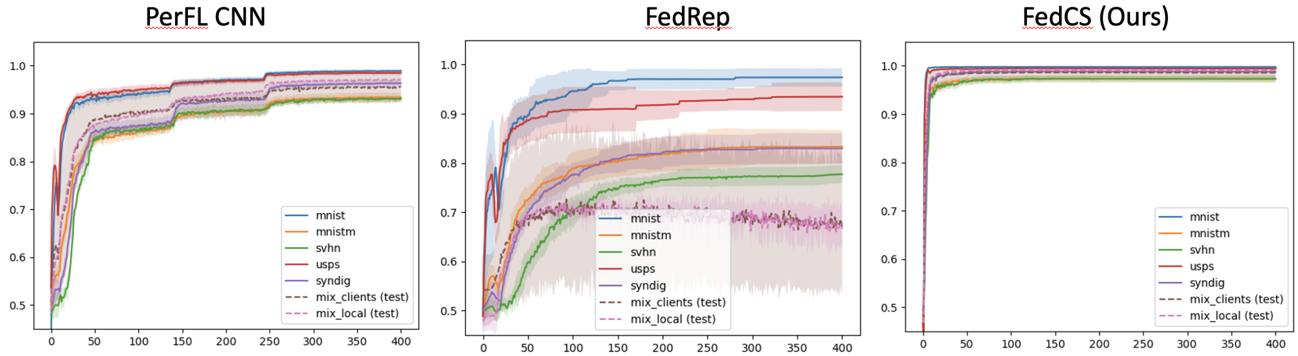


Fig. 10. Part of experiments on Digit5. The horizontal axis denotes communication rounds, and the vertical axis denotes accuracy. We demonstrate the averaged accuracy within each client group and marked them with different colours. We can find the performance on different distributions (client groups) vary significantly through FedAvg+FT and FedBN. FedCS has the most robust performance even on unseen clients (group 8 and 9)