ICLR 2023| Personalized Federated Learning with Feature Alignment and Classifier Collaboration

丁千惠

January 8, 2025

Abstract

In the field of federated learning, the issue of data heterogeneity is particularly challenging, prompting researchers to explore various strategies for training client-specific models. A common approach in deep learning tasks involves sharing feature representations while customizing the classifier heads for each client. However, previous studies have failed to integrate global information into local representation learning and have not emphasized the fine-grained collaboration between local classifier heads, which limits the generalization performance of the models. This study proposes a method for explicitly aligning local and global features using global semantic information to achieve superior representations. We also formulate an optimization problem to compute the benefits of classifier combinations for each client, quantifying it as a function of the combination weights. Extensive evaluations on benchmark datasets across various heterogeneous data scenarios have confirmed the effectiveness of our approach.

Keywords: Federated Learning, Personalization, Collaboration, General Machine Learning.

1 Introduction

Federated learning (FL) has emerged as a significant advancement in machine learning, addressing the growing concerns over data privacy and security. Traditional machine learning methods often require centralized data storage, which can lead to privacy issues and data silos [28]. FL overcomes these challenges by enabling multiple clients to collaboratively train a shared model without sharing their raw data. This approach was first proposed by Google in 2016 to allow Android users to update models locally while keeping their data private.FL operates by distributing the training process across various devices or servers, where each client trains a local model using its own data. These local models are then aggregated to update a global model, with only model parameters being exchanged between the clients and the central server. This method not only protects user privacy but also reduces the computational and storage burden on the server.Despite its benefits, FL faces several challenges, including privacy and security threats, heterogeneous data distributions, and high communication overhead. Research is ongoing to develop more efficient algorithms and techniques to address these issues and enhance the practicality of FL in various applications such as healthcare, smart cities, and edge computing [27]. This paper is dedicated to studying heterogeneity issues, particularly in scenarios characterized by label distribution shifts.

Drawing inspiration from the advancements in multi-task learning, the author posits that the development of an effective global feature representation and the exploration of inter-task relationships are instrumental in enhancing personalized federated learning [5]. In the context of federated learning, the inherent heterogeneity among client datasets can be conceptualized as a multi-task learning problem, where each client is engaged in a distinct yet related task.

In this research, we examine the situation where label distribution shifts occur, maintaining the same number of classes across clients while the sample sizes per class vary considerably. This results in distinct label distributions for each client's local tasks. We approach federated learning through the lens of multi-task learning, utilizing both shared representations and collaborative classifiers across clients. Specifically, we employ the global feature centroid for each class to regulate local training, which serves as a form of explicit feature alignment. This strategy minimizes the variability in representations produced by local feature extractors, thus enhancing the effectiveness of global aggregation. We also facilitate classifier collaboration using client-specific linear combinations, which promotes collaboration among clients with similar characteristics and mitigates negative transfer from dissimilar clients. To calculate the optimal combining weights, we leverage local feature statistics and data distribution information. This is achieved by solving a quadratic programming problem aimed at minimizing the expected testing loss for each client. Moreover, with simple adjustments, our framework remains effective in scenarios involving concept shifts, where the same label can have different interpretations across clients.

Our contributions are manifested in several aspects. We focus on classification tasks in the field of deep learning and have developed an innovative federated learning (FL) framework that integrates personalized classifier aggregation (FedPAC) and feature alignment techniques. This framework is designed to enhance the performance of client-specific tasks. We have tested the proposed framework on benchmark datasets with varying levels of data heterogeneity to verify its effectiveness in improving model performance. The results indicate that our method can increase the average model accuracy by 2-5%. In summary, our key contributions are as follows:

- We quantify the testing loss for each client under the classifier combination by characterizing the discrepancy between the learned model and the target data distribution, thereby revealing a new bias-variance trade-off.

- We have designed a novel personalized federated learning framework that combines feature representation alignment with optimal classifier combination, facilitating rapid convergence and high efficiency of the model.

- Through extensive evaluation on real datasets with different levels of data heterogeneity, we have demonstrated the high adaptability and robustness of FedPAC.

The advantages of FedPAC technology over current personalized federated learning (FL) technologies are mainly reflected in two aspects: (i) Frequent local representation learning updates. This technology employs a feature alignment strategy, effectively controlling the drift phenomenon in the local representation learning process. This strategy allows each client to make only a small number of local-global parameter adjustments during each communication process, enabling multiple local updates. As a result, it achieves higher-quality representations in a communication-efficient manner. (ii) Benefits of classifier collaboration. This study utilizes a theoretically supported optimal weighted averaging method to integrate information from similar clients. This approach not only enhances the generalization performance of data-scarce clients but also avoids the negative impact of information transfer from unrelated clients.

2 Related works

2.1 FEDERATED LEARNING WITH NON-IID DATA

Many efforts have been devoted to improving the global model learning of FL with non-IID data. A variety of works focus on optimizing local learning algorithms by leveraging well-designed objective regularization and local bias correction. For example, FedProx [16] adds a proximal term to the local training objective to keep updated parameter close to the original downloaded model, SCAFFOLD [13] introduces control variates to correct the drift in local updates, and MOON [14] adopts the contrastive loss to improve the representation learning. Class-balanced data re-sampling and loss re-weighting methods can improve the training performance when clients have imbalanced local data [11]. Besides, data sharing mechanisms and data augmentation methods are also investigated to mitigate the non-IID data challenges [31]. From the model aggregation perspective, selecting clients with more contribution to global model performance can also speed up the convergence and mitigate the influence of non-IID data [26]. With the availability of public data, it is possible to employ knowledge distillation techniques to obtain a global model despite of the data heterogeneity [18]. The prototype-based methods are also utilized in some FL works, such as [21]proposes a prototype-based weight attention mechanism during global aggregation and [32]utilize the prototypes to enhance the local model training. Different from above methods, this paper aims at learning a customized model for each client.

2.2 MODEL PERSONALIZATION IN FL

In the literature, popular personalized FL methods include additive model mixture that performs linear combination of local and global modes, such as L2CD [10] and APFL [8]; multi-task learning with model dissimilarity penalization, including FedMTL [25], pFedMe [9] and Ditto [15]; meta-learning based local adaption [1]; parameter decoupling of feature extractor and classifier, such as FedPer [3], LG-FedAvg [17] and FedRep [7]. A special type of personalized FL approach is clustered FL to group together similar clients and learn multiple intra-group global models [23]. Client-specific model aggregations are also investigated for fine-grained federation, such as FedFomo [30] and FedAMP [4], which have similar spirit to our approach. Nevertheless, existing client-specific FL methods are usually developed by evaluating model similarity or validation accuracy in a heuristic way, and these techniques need to strike a good balance between communication/-computation overhead and the effectiveness of personalization. FedGP based on Gaussian processes [2] and selective knowledge transfer based solutions [29] are also developed, however those methods inevitably rely on public shared data set or inducing points set. Besides, pFedHN enabled by a server-side hypernetwork [24] and FedEM that learns a mixture of multiple global models [19] are also investigated for generating customized model for each client. However, pFedHN requires each client to communicate multiple times for learning a representative embedding and FedEM significantly increases both communication and computation/storage overhead. Recently, Fed-RoD [6] proposes to use the balanced softmax for learning generic model and vanilla softmax for personalized heads. FedBABU [22] proposes to keep the global classifier unchanged during the feature representation learning and perform local adoption by fine-tuning. kNN-Per [20] applies the ensemble of global model and local kNN classifiers for better personalized performance. Our work shares the most similar learning procedure with FedRep [7], but differs in the sense that we employ global knowledge for guiding local representation learning and also perform theoretically-guaranteed classifier heads combination for each client.

3 Method

3.1 PROBLEM SETUP

We consider a setup involving m clients and a central server, where all clients communicate with the server to collaboratively train personalized models without sharing their original private data.Each client i follows its own data distribution $P^{(i)}_{(XY)}$ on $X \times Y$ where X is the input space and Y is the label space with K categories in total.We posit that the data distributions across any two clients are distinct.Let $l : X \times Y \rightarrow R_{+}$ denote the loss function given local model ω_i and data point sampled from $P^{(i)}_{(XY)}$, e.g., cross-entropy loss, then the underlying optimization goal of PFL can be formalized as follows:

$$\min_{W}\{F(W) := \frac{1}{m}\sum_{i=1}^{m} E_{(x,y)\sim P^{(i)}_{XY}}[l(\omega_i; x, y)]\} \tag{1}$$

where $W = (\omega_1, \omega_2, ..., \omega_m)$ denotes the collection of all local models. However, the true underlying distribution is inaccessible and the goal is usually achieved by empirical risk minimization(ERM). Assume each client has access to n_i IID data points sampled from $P^{(i)}_{(XY)}$, and we assume the empirical marginal distribution $\hat{P}^{(i)}_{Y}$ is identical to the true $P^{(i)}_{Y}$. Then the training objective is

$$w^{*} = \arg\min_{w} \frac{1}{m}\sum_{i=1}^{m}[L_i(\omega_i) + R_i(\omega_i; \Omega)] \tag{2}$$

where $L_i(\omega_i) = \frac{1}{n_i}\sum_{l=1}^{n_i} l(\omega_i; x^{(i)}_{l}, y^{(i)}_{l})$ is the local average loss over personal training data, e.g., empirical risk; Ω is some kind of global information introduced to relate clients, and the $R_i(\cdot)$ is a predefined regularization term for preventing ω_i from over-fitting.

3.2 SHARING FEATURE REPRESENTATION

Without loss of generality, we decouple the deep neural network into the representation layer and the final decision layer, where the former is also known as the feature extractor, and the latter refers to the classifier head in classification tasks. The feature embedding function $f : x \rightarrow R^{d}$ is a learnable network parameterized by θ_f and d is the dimension of feature embedding. Given a data point x, a prediction for extracted feature vector $z = f(x)$ can be generated by a linear function $g(z)$ parameterized by ϕ_g. In the rest of paper, we omit the subscripts of θ_f and ϕ_g for simplicity, i.e., $\omega = \{\theta, \phi\}$. Due to the insufficiency of client data, the feature representation learned locally tends to overfit, resulting in poor generalization. A viable solution is to leverage data from other clients by sharing the feature representation layer. However, frequent local updates of private data can lead to local overfitting among clients and an increase in parameter diversity, causing the aggregated model to deviate from the optimal representation. To address this issue, we introduce a new regularization term for the learning of local feature representations.

During the update of local models, clients need to consider both the loss function of supervised learning and the generalization error. To this end, this study adopts the concept of global feature centroids and introduces a new regularization term into the local training objective function, aiming to enable local representation learning to benefit from the global dataset. The local regularization term is given by

$$R_i(\theta_i; c) = \frac{\lambda}{n_i} \sum_{l=1}^{n_i} \frac{1}{d} \| f_{\theta_i}(x_l) - c_{y_l} \|_2^2 \tag{3}$$

where $f_{\theta_i}(x_j)$ is the local feature embedding of given data point x_j, and c_{y_j} is the corresponding global feature centroid of class y_j, λ is hyper-parameter to balance supervised loss and regularization loss. By leveraging global semantic feature information, the regularization term significantly enhances the performance of each client. In short, it enables each client to grasp the invariant representation of the task through explicit feature distribution alignment. As a result, the diversity of local feature extractors $\{\theta_i\}_{i=1}^m$ could also be regularized while minimizing local classification error.

3.3 CLASSIFIER COLLABORATION

We firmly believe that, in addition to optimizing the feature extractor through the shared representation layer, merging classifiers from other clients with similar data distributions can also significantly enhance performance. Unlike previous studies [3] [7], we not only retain the locally trained classifiers but also implement a weighted average for the clients to optimize the performance of local classifiers. Intuitively, when the local data volume is insufficient, the locally learned classifier may face a high variance issue. Therefore, clients with similar data distributions can actually train personalized classifiers through knowledge transfer across clients. However, assessing the similarity between clients and the transferability of knowledge remains a challenge. To this end, we conduct a linear combination of those received classifiers for each client i to reduce the local testing loss:

$$\hat{\phi}_i^{(t+1)} = \sum_{j=1}^{m} \alpha_{ij} \phi_j^{(t+1)}, s.t. \sum_{j=1}^{m} \alpha_{ij} = 1 \tag{4}$$

with each coefficient $\alpha_{ij} \geq 0$ determined by minimizing local expected testing loss, which can be formulated as the following optimization problem:

$$\alpha_i^* = \arg\min_{\alpha_i} E_{(x,y)\sim P_{XY}^{(i)}} [l(\theta, \sum_{j=1}^{m} \alpha_{ij} \phi_j; x, y]) \tag{5}$$

For a better collaboration, we need to update the coefficients α_i adaptively during the training process.

3.4 ALGORITHM DESIGN

3.4.1 LOCAL TRAINING PROCEDURE

For local model training at round t, we first replace the local representation layers $\theta_i^{(t)}$ by the received global aggregate $\tilde{\theta}^{(t)}$ and update private classifier analogously. Then, we perform stochastic gradient decent steps to train the two parts of model parameters as follows:

- Step 1: Fix θ_i, Updateϕ_i

$$\phi_i^{(t)} \leftarrow \phi_i^{(t)} - \eta_g \nabla_\phi l(\theta_i^{(t)}, \phi_i^{(t)}; \xi_i) \tag{6}$$

  where ξ_i denotes the mini-batch of data, η_g is the learning rate for updating classifier.

- Step 2: Fix New ϕ_i, Update θ_i

$$\theta_i^{(t)} \leftarrow \theta_i^{(t)} - \eta_f \nabla_\theta [l(\theta_i^{(t)}, \phi_i^{(t+1)}; \xi_i) + R_i(\theta_i^{(t)}; c^{(t)})] \tag{7}$$

  where η_f is the learning rate for updating representation layers, $c^{(t)} \in R^{K \times d}$ is the collection of global feature centriod vector for each class, and $K = |Y|$ is the total number of classes.

Before local feature extractor updating, each client should extract the local feature statistics $\mu_i^{(t)}$ and $V_i^{(t)}$ through a single pass on the local dataset, which will be utilized to estimate the optimal classifier combination weights for each client. Moreover, after updating the local feature extractor, we compute the local feature centroid for each class as follows:

$$\hat{c}_{i,k}^{(t+1)} = \frac{\sum_{l=1}^{n_i} 1(y_l^{(i)} = k) f_{\theta_i^{(t+1)}}(x_l^{(i)})}{\sum_{l=1}^{n_i} 1(y_l^{(i)} = k)}, \forall k \in [K] \tag{8}$$

3.4.2 GLOBAL AGGREGATION

- **Global Feature Representation.** Like the common algorithms, the server performs weighted averaging of local representation layers with each coefficient determined by the local data size.

$$\tilde{\theta}^{(t+1)} = \sum_{i=1}^{m} \beta_i \theta_i^{(t)}, \beta_i = \frac{n_i}{\sum_{i=1}^{m} n_i} \tag{9}$$

- **Classifier Combination.** The server uses received feature statistics to updates the combination weights vector α_i by solving equation 10 and conducts classifier combination for each client i.

$$\alpha_i^* := \arg\min_{\alpha_i} R_i(\alpha_i),\ s.t. \sum_{j=1}^{m} \alpha_{ij} = 1\ and\ \alpha_{ij} \geq 0, \forall j \tag{10}$$

- **Update Global Feature Centroids.** After receiving the local feature centroids, the following centroid aggregating operation is conducted to generate an estimated global centroid c_k for each class k.

$$c_k^{(t+1)} = \frac{1}{\sum_{i=1}^{m} n_{i,k}} \sum_{i=1}^{m} n_{i,k} \hat{c}_{i,k}^{(t+1)},\ \forall k \in [K] \tag{11}$$

4 Implementation details

4.1 Comparing with the released source codes

Training Feature Representation. The code for training the representation layer is based on the open-source code provided in the article, with no modifications or improvements made. This means that the implementation of the feature extraction process remains consistent with the original work, leveraging the established methodologies and algorithms presented in the source code. By using the unaltered code, we ensure that the feature representation aligns with the foundational concepts and techniques outlined in the article, allowing for a reliable and accurate extraction of relevant features from the data.

Classifier Collaboration. The calculation of client aggregation weights presents two drawbacks.

- High Computational Complexity. (i) Complex Optimization Process: FEDPAC calculates the client aggregation weights by optimizing a complex formula. This optimization process involves multiple variables and constraints. For instance, it needs to consider the differences in data distribution among clients, the similarity of model parameters, and other factors, all of which require complex mathematical operations. (ii) Resource Consumption: Due to the need for frequent data exchange between clients and the server, as well as extensive computations, this results in high consumption of computational resources. Especially in large-scale federated learning environments where there are numerous clients, each with varying data volumes and model complexities, the computational burden is further increased.

- Lack of Intuitive Reflection of Client Similarity. (i) Dependence on Optimization Results: The calculation of aggregation weights in FEDPAC mainly relies on the results of the optimization formula, rather than directly on the similarity of client data or models. This means that the weight allocation may not intuitively reflect the similarity between clients, as the optimization process may focus more on enhancing overall performance rather than the specific relationships between clients. (ii) Difficulty in Capturing Subtle Differences: Due to the lack of direct measurement of client data distribution similarity, FEDPAC may not effectively capture subtle differences in data distribution among clients. This can lead to suboptimal aggregation results in certain situations, especially when there are significant differences in client data distribution, making it difficult to achieve the best aggregation effect.

Therefore, this paper improves the calculation of aggregation weights by proposing a decision-layer aggregation method based on class prototypes. To more clearly explain how to calculate the aggregation weights, let's consider a scenario with $m + 1$ clients $A, A_1, A_2, ..., A_M$, focusing on the perspective of client A. During the decision-layer aggregation on the server, the class prototypes from clients $A_1, A_2, ..., A_M$ are input into the decision layer of client A, resulting in m loss values. More generally, the loss value obtained by inputting the class prototype of client i into the decision layer of client j is given by:

$$f_{i,j}^{(t)} = l(\hat{c}_i^{(t)}; \phi_j^{(t)}) \tag{12}$$

The weights are calculated based on these m loss values, as follows:

$$\alpha_{i,j}^{(t)} = \frac{e^{-f_{i,j}^{(t)}}}{\sum_{k=1}^{m} e^{-f_{k,j}^{(t)}}},\ s.t. \sum_{j=1}^{m} \alpha_{i,j} = 1\ and\ \alpha_{i,j} \geq 0, \forall j \tag{13}$$

It should be noted that when $f_{i,j}^{(t)}$ is less than 0, $\alpha_{i,j}^{(t)}$ equals 0.

This approach allows for a more nuanced understanding of the relationships between clients, as it directly incorporates the compatibility of their class prototypes into the aggregation process. By leveraging the loss values derived from these prototypes, the method can effectively capture the similarities and differences among clients, leading to a more accurate and robust aggregation of their models. This enhanced aggregation strategy not only improves the overall performance of the federated learning system but also provides a clearer insight into the contributions of each client, thereby facilitating better collaboration and knowledge sharing among them.

4.2 Experimental environment setup

Datasets and Models. We consider image classification tasks and evaluate our method on two popular datasets: Fashion-MNIST with 10 categories of clothes, CIFAR-10 with 10 categories of color images. We construct two different CNN models for Fashion-MNIST and CIFAR-10, respectively.

Data Partitioning. Based on the research findings of [13] [30] [12], this study ensures that all participating clients have equal data scales. Specifically, this study proportionally extracts $s\%$ of the data from all categories (typically set at 20%), while the remaining $(100 - s)\%$ of the data is selected from each client's specific primary category. This study deliberately divides clients into different groups to ensure that clients within each group share the same primary category. Additionally, to accommodate the characteristics of federated learning (FL), this study intentionally maintains the scale of local training data at a relatively low level. At the same time, the distribution of test data for each client is consistent with its training data.

Compared Methods. We compare the following baselines: Local-only, where each client trains its model locally; FedAvg that learns a single global model and its locally fine-tuned version (FedAvg-FT); multi-task learning methods, including APFL, pFedMe and Ditto; parameter decoupling methods, including LG-FedAvg, FedPer, FedRep and FedBABU; Fed-RoD and kNN-Per that learn an extra local classifier based on the global feature extractor and use model ensemble for prediction; FedFomo that conducts inter-client linear combination and pFedHN enabled by a server-side hypernetwork.

Training Settings. We employ the mini-batch SGD as a local optimizer for all approaches, and the number of local training epochs is set to E = 5 unless explicitly specified. The number of global communication rounds is set to 200 for all datasets, where all FL approaches have little or no accuracy gain with more communications. We report the average test accuracy across clients.

5 Results and analysis

Performance Comparison. We conducted experiments under two settings, with the number of clients being 20 and 100, respectively. For the latter, we adopted a random client selection with a sampling rate of $C = 0.3$ and fully participated in the last round. For all datasets, the size of the training data for each client was set to 600. The main results are shown in Table 1. It is evident that, for the original results, FedPAC performs well in both small-scale and large-scale FL systems. For all datasets, FedPAC leads other methods in terms of average test accuracy, which proves the effectiveness and benefits of global feature alignment and classifier collaboration among clients. The replicate results on the Fashion-MNIST dataset are highly consistent with the original study results. However, there is some deviation in the replicate results on the CIFAR-10 dataset compared to the original study. A detailed analysis of the reasons mainly attributes to the limitation of training epochs and possible subtle differences in the replicate process. After improving the FedPAC algorithm, this study achieved performance on par with or even slightly better than the replicate results on both datasets, thereby verifying the effectiveness of the proposed improvements.

Method	Fashion-MNIST		CIFAR-10	
	20 clients	100 clients	20 clients	100 clients
Local-only	85.68	86.37	65.43	64.68
FedAvg	85.28	86.89	70.05	73.82
FedAvg-FT	90.47	91.53	76.56	79.34
FedPer	87.43	87.88	68.37	72.26
LG-FedAvg	85.66	86.28	65.19	65.38
FedRep	88.25	88.54	71.36	72.58
FedBABU	89.68	91.38	76.46	78.39
Fed-RoD	90.11	91.61	77.16	81.97
kNN-Per	90.02	90.93	76.73	80.47
APFL	89.45	90.88	74.39	77.32
pFedMe	89.76	89.85	68.81	69.85
Ditto	90.43	91.08	77.02	80.09
FedFomo	88.95	90.19	73.33	76.21
pFedHN	88.36	87.93	76.95	79.39
FedPAC	91.83	92.72	81.13	83.36
Replicate	91.95	92.47	76.50	78.53
Modify	92.03	92.60	76.37	78.55

Table 1. The comparison of final test accuracy ($\%$) on different datasets. We apply full participation for FL system with 20 clients, and apply client sampling with rate 0.3 for FL system with 100 clients.

Effects of Data Heterogeneity. We vary the values of s to simulate different levels of data heterogeneity, while $s = 0\%$ indicates a highly heterogeneous case (pathological non-IID) and$s = 80\%$ means data across clients are more homogeneous. We evaluate the CIFAR-10 dataset and the results of different methods are reported in Figure 1. It can be found that our method consistently outperforms other baselines, which demonstrates its adaptability and robustness in a variety of heterogeneous data scenarios. The results of the replicate are shown in Figure 2.

Effects of Data Size. We test our FedPAC and FedAvg with varying local data sizes, recording the resulted model accuracy as in Figure 3. The results indicate that clients with different data sizes can consistently benefit from participating in FL and our method achieves higher performance gain.The results of the replicate are shown in Figure 4.

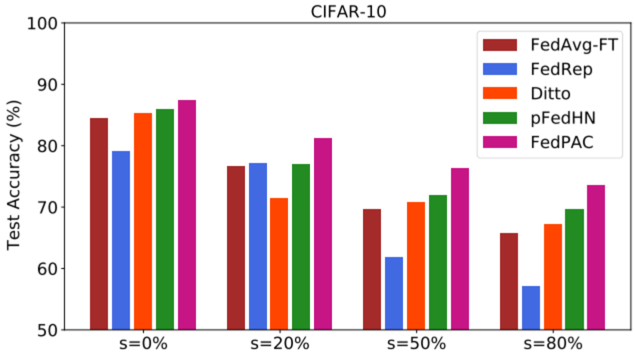

Figure 1. Effects of Data Heterogeneity

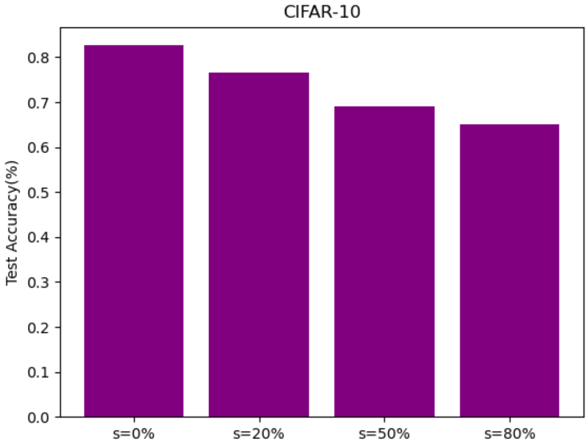

Figure 2. Effects of Data Heterogeneity

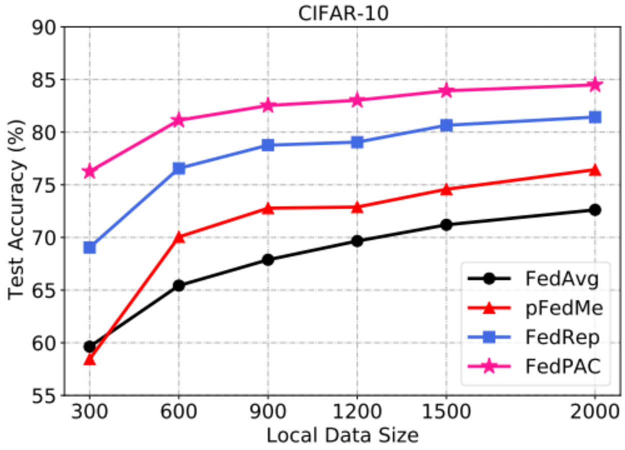

Figure 3. Effects of Data Size

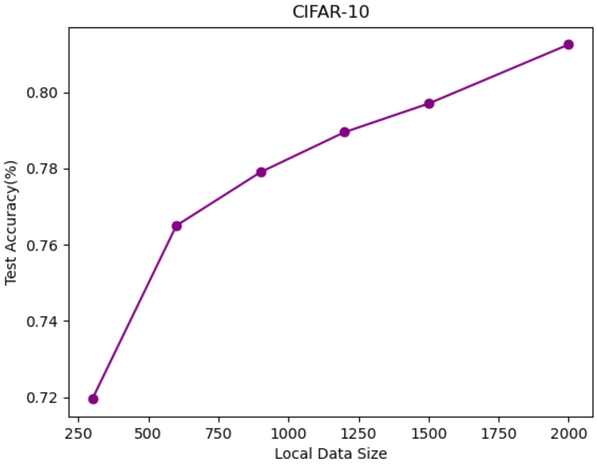

Figure 4. Effects of Data Size

6 Conclusion and future work

This paper introduces global feature alignment for enhanced representation learning and a novel classifier ensemble algorithm for building personalized classifiers in federated learning (FL), providing both theoretical and empirical evidence for their practicality in heterogeneous environments. Additionally, this paper improves the calculation of classifier aggregation weights based on FedPAC. Future work includes analyzing optimal model personalization in more complex environments, such as decentralized systems or clients with dynamic data distributions, and studying the optimal aggregation of local feature extractors.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Debiasing model updates for improving personalized federated training. In *International Conference on Machine Learning*, volume 139, pages 21–31, 2021.

[2] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. In *Proc. Conf. on Neural Information Processing Systems*, volume 34, pages 8392–8406, 2021.

[3] Manoj Ghuhan Arivazhagan, V. Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv*, 2019.

[4] Martin Beaussart, Felix Grimberg, Mary-Anne Hartley, and Martin Jaggi. Waffle: Weighted averaging for personalized federated learning. *arXiv*, 2021.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 35(8):1798–1828, 2013.

[6] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *Proc. Int. Conf. on Learning Representations*, 2022.

[7] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, volume 139, pages 2089–2099, 2021.

[8] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. In *Proc. Int. Conf. on Learning Representations*, 2021.

[9] Canh T. Dinh, Nguyen Hoang Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Proc. Conf. on Neural Information Processing Systems*, volume 33, pages 21394–21405, 2020.

[10] Filip Hanzely and Peter Richtarik. Federated learning of a mixture of global and local models. In *Proc. Int. Conf. on Learning Representations*, 2021.

[11] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Proc. Euro. Conf. on Computer Vision*, number 17, page 76–92, 2020.

[12] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. *Proc. AAAI Conf. on Artificial Intelligence*, 35(9):7865–7873, 2021.

[13] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, volume 119, 2020.

[14] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 10708–10717, 2021.

[15] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, volume 139, pages 6357–6368, 2021.

[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Machine Learning and Systems*, 2:429–450, 2020.

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

[18] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Proc. Conf. on Neural Information Processing Systems*, 2020.

[19] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Proc. Conf. on Neural Information Processing Systems*, volume 34, pages 15434–15447, 2021.

[20] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, volume 162, pages 15070–15092, 2022.

[21] Umberto Michieli and Mete Ozay. Prototype guided federated learning of visual feature representations. *arXiv*, 2021.

[22] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. In *Proc. Int. Conf. on Learning Representations*, 2022.

[23] 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, 2021.

[24] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, volume 139, pages 9489–9502, 2021.

[25] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Proc. Conf. on Neural Information Processing Systems*, number 11, page 4427–4437, 2017.

[26] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. *IEEE Conference on Computer Communications*, pages 1698–1707, 2020.

[27] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *INTERNATIONAL JOURNAL OF MACHINE LEARNING AND CYBERNETICS*, 14(2):513–535, 2023.

[28] Mengna Yang, Yejun He, and Jian Qiao. Federated learning-based privacy-preserving and security: Survey. In *Computing, Communications and IoT Applications (ComComAp)*, pages 312–317, 2021.

[29] J. Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wencao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *Proc. Conf. on Neural Information Processing Systems*, volume 34, pages 10092–10104, 2021.

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

[31] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv*, 2018.

[32] Tailin Zhou, Jun Zhang, and Danny H. K. Tsang. Fedfa: Federated learning with feature anchors to align features and classifiers for heterogeneous data. *IEEE Transactions on Mobile Computing*, 23:6731–6742, 2022.