Personalized Federated Learning via Deviation Tracking Representation Learning

Jaewon Jang and Bong Jun Choi
Computer Science and Engineering
Soongsil University
Seoul, South Korea
{jwon0524, davidchoi}@soongsil.ac.kr

***Abstract*—Federated learning preserves privacy by decentralized training of individual client devices, ensuring only model weights are shared centrally. However, the data heterogeneity across clients presents challenges. This paper focuses on representation learning, a variant of personalized federated learning. According to various studies, the representation learning model can be divided into two: the base layer, shared and updated to the server, and the head layer, localized to individual clients. The novel approach exclusively utilizes the base layer for both local and global training, arguing that the head layer might introduce noise due to data heterogeneity. This can potentially affect accuracy, and the head layer is used only for fine-tuning after training to capture unique client data characteristics. Here, we observed that prolonged base training can diminish accuracy in the post-fine-tuning. As a countermeasure, we proposed a method to determine the best round for fine-tuning based on monitoring the standard deviation of test accuracy across clients. This strategy aims to generalize the global model for all the clients before fine-tuning. The study highlights the downside of excessive base training on fine-tuning accuracy and introduces a novel approach to pinpoint optimal fine-tuning moments, thereby minimizing computational and communication overheads. Similarly, we achieved a better accuracy of 53.6% than other approaches while there's a trade-off of minute communication round.**

***Index Terms*—federated learning(FL), data heterogeneity, personalized federated learning(PFL), representation learning, meta-learning**

I. Introduction

In the era of digitalization and big data, concerns over privacy data protection have taken center stage in global discussions. Legal regulations, such as GDPR [1] in Europe, restricted the collection and sharing of client data for cooperation. As a result, it is essential to safeguard personal information while also learning from heterogeneous data distributed across various devices, like mobile phones. Federated learning [2] provides some basic privacy to the user by avoiding the sharing of the client's data directly to a central server. In this approach, model training occurs on each edge device (e.g., smartphones), and only the model weights are shared with the central server.

In real-world scenarios, clients are heterogeneous, and their data distributions can vary significantly. Such diversity across user data has led to exploring various methods in federated learning, such as FedProx [3], SCAFFOLD [4] etc., to address the client drift problem. However, these approaches do not capture the unique local characteristics of heterogeneous clients and do not fit the client's personalized data better. This led to the introduction of personalized federated learning (PFL), which aims to generalize a global model and better fit on the client's local data. One of the approaches of PFL that has been used in this study is representation learning. In representation learning, specific layers of deep neural networks extract personalized patterns from the client's input data. Similarly, the shallower layers detect low-level features from images, representing common input data characteristics. In contrast, deeper layers discern high-level features or intricate patterns more tailored to a specific task.

Representation learning within a federated learning framework isn't fundamentally different from its role in conventional deep learning. In this context, the deep layers of each client's model are trained to reflect the unique characteristics of their local data. These locally learned representations can then be transmitted to the central server to inform the training of the global model. To delve deeper, as detailed in [8], [9], [10], [11], [12], the model is divided into two parts: the base and the head. The base is shared among all clients and updated on the server, while the head remains with each client, managing specific data traits or characteristics. Functionally, the base part extracts features, whereas the head handles classification. Communication predominantly takes place via the base part, thus optimizing communication efficiency. By maintaining the head part locally, the system tailors personalized models for individual users and ensures enhanced data privacy.

In FedBABU [12], only the base layer is used for both training the local model and aggregation of weights for the global model. Due to significant client data heterogeneity, the head layer might introduce noise during training, negatively impacting accuracy. As a result, the head layer has designated the role of fine-tuning at the last global training round. Notably, given that training primarily uses the model's base, the variation in evaluation metrics is relatively minimal. This lack of variance complicates the task of pinpointing the optimal fine-tuning moment, and we often have to rely on experience to make this determination. Furthermore, our experiments indicate that over-extending the training of the

base layer can lead to decreased accuracy after fine-tuning. Our proposed work determines the best global round for fine-tuning by monitoring the standard deviation of test accuracy across all clients after each round. We used an approach similar to meta-learning [5], [6], but our objective is to have the global model, trained solely on the base layer, generalize across all clients before fine-tuning begins. Thus, fine-tuning should start at the optimal global round; having minimal accuracy disparity among all clients indicates better generalization.

The contributions of this paper are as follows:

- We identified a new problem: excessive base training negatively impacts accuracy after fine-tuning. The phenomenon is demonstrated using experimental results.
- We presented an approach based on the standard deviation method to identify the optimal fine-tuning point during training, reducing unnecessary computation and communication costs.

The remainder of the paper is organized as follows. Section II presents the related work in personalized federated learning and representation learning. Section III presents the representation-based personalized federated learning model. Section IV provides the details of our proposed algorithm. Section V demonstrates the performance evaluation of the proposed algorithm compared to other existing personalized federated algorithms. Finally, Section VI provides the conclusion and future research directions.

II. Related Work

A. Federated Learning

FedAvg [2] introduced the concept of federated learning as a new method to train machine learning models directly on devices while keeping data localized. The primary objective was to address privacy concerns while using user data on edge devices like smartphones.

B. Personalized Federated Learning

1) Meta Learning: Meta-learning, or learning to learn, has gained significant attention in the machine learning (ML) community. Its main objective is to train models on multiple tasks, so that they can quickly adapt to new, unseen tasks. MAML [5] aims to find a generalizable model initialization that can be fine-tuned for a specific task with minimal data. It is designed to learn a set of parameters from which a few gradient steps can lead to effective task-specific fine-tuning. Reptile, proposed later as a simpler alternative to MAML [6], also focuses on meta-learning but with a less computationally intensive update mechanism. Unlike MAML's bi-level optimization, Reptile employs a form of moving average of the task-specific parameters. Therefore, Reptile uses less computation and memory than MAML.

2) Representation Learning: In real-world scenarios, client data distributions are heterogeneous(non-iid). Representation learning has been used to mitigate such statistical heterogeneity in which the model is split into two architectural parts: the base (common shared) and the head (kept locally) [8]. The base layer shares similar common features across clients, while the head layer utilizes each client's unique features for personalization. For instance, in a CNN model, the base can be the convolutional layer, and the head layer can be the fully connected layer because the fully connected layer (classifier) represents specific features of the model. LG-FedAVG [9] demonstrated that Representation Learning generalizes better to new devices than other learning techniques. Additionally, fair representations that obscure protected attributes were effectively learned through adversarial training. FedRep [10] leverages the distributed computational power among clients to perform many local updates involving low-dimensional local parameters for each representation update and demonstrates fast convergence in linear regression problems. FedROD [11] considered both local model accuracy and global model accuracy to address the discrepancy in validation methods between general FL algorithms and personalized FL algorithms. FedBABU [12] showed that when client data is heterogeneous, the head can negatively impact personalization during training. Thus, a fixed random head was used for the learning process.

However, all related studies have utilized a fixed number of global rounds in their experimental setups. Specifically, in [12], where fine-tuning is conducted at the final point of the global round, it is challenging to find the optimal number of global rounds empirically. Therefore, we propose a method to halt training by using the standard deviation of the test accuracy across clients to find the optimal fine-tuning point.

III. System Model

TABLE I
List of Symbols Used

Symbol	Description
N	Total number of clients
C_i	Client i, $i \in \{1, 2, \ldots, N\}$
$\|D_i\|$	Number of data points for client i
$\|D\|$	Total number of data points
T	Total global rounds
F	Fine tune rounds
R	The optimal fine tune round
θ_i	Local parameter for client i
θ_G	Global model parameter
$\theta_{i,b}$	Base parameter shared among all clients
$\theta_{i,p}$	Head parameter of client i
a_i	Accuracy of client i
$\bar{a}$	Average of each client's accuracy
σ	The standard deviation of each client's accuracy
$\bar{a}_w$	Average of each client's accuracy in recent window
σ_w	Standard deviation of each client's accuracy in recent window

1) Federated Learning: We assume that each client C_i possess data $\mathbf{D}_i = (x_i, y_i) \in R^d$, where $i \in 1, 2, \ldots, N$ represents i-th client out of a total of N clients and d represents the input dimension. Each client i updates its local model parameter θ_i based on its data D_i and the global model parameter θ_G as

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta \nabla_{\theta_i} \mathcal{L}(D_i, \theta_i^{(t)}), \tag{1}$$

where η is the learning rate, ∇L is the gradient of the loss function L, and $t \in (1, 2, \ldots, T)$ denotes the number of

rounds. The central server updates the global parameter θ_G based on all clients' local parameter updates as

$$\theta_G^{(t+1)} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \theta_i^{(t+1)}, \tag{2}$$

where $|D_i|$ is the number of data points for client i, and $|D|$ is the total number of data points.

2) Representation Learning: The model parameters are divided into two components $\theta_i = (\theta_{i,b}, \theta_{i,h})$. Here, $\theta_{i,b}$ represents the base parameter shared among all clients and $\theta_{i,h}$ represents the head parameter of the i-th client. The model parameters from Eq. (1) is modified as follows:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta \nabla_{\theta_i} \mathcal{L}(D_i, \theta_{i,b}^{(t)}, \theta_{i,h}^{(0)}). \tag{3}$$

During learning, the gradient of the head parameter is stopped, and a global model is learned using only the base as

$$\theta_G^{(t+1)} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \theta_{i,b}^{(t+1)}. \tag{4}$$

IV. Proposed Algorithm

Given a set of client's accuracy $a_i \in a_1, a_2, \ldots, a_N$, the average of each client's accuracy $\bar{a}$ is computed as

$$\bar{a} = \frac{1}{N} \sum_{i=1}^{N} a_i. \tag{5}$$

The standard deviation σ of each client's accuracy is calculated as

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (a_i - \bar{a})^2}. \tag{6}$$

We interpreted the deviation of the test accuracy of the client as time series data by dividing it by window size. Given a window size denoted as w, the average accuracy of each client $\bar{a}$ for the last w rounds, $\bar{a}_w$, is computed as

$$\bar{a_w} = \frac{1}{N \times w} \sum_{i=1}^{N} \sum_{j=1}^{w} a_{i,j}, \tag{7}$$

where $a_{i,j}$ represents the accuracy of the i^{th} client at the j^{th} round within the window. The standard deviation for the recent window w rounds, σ_w, is given by

$$\sigma_w = \sqrt{\frac{1}{N \times w} \sum_{i=1}^{N} \sum_{j=1}^{w} (a_{i,j} - \bar{a}_w)^2}. \tag{8}$$

The optimal fine tune round R is determined as follows: Observe the rounds where σ_w converges to a value σ_w^*. Identify the round $R \leq T$ such that

$$\sigma_w'(R) < 0 \text{ and } |\sigma_w''(R)| \approx \epsilon, \tag{9}$$

where ϵ is a small positive threshold. During the optimal fine-tuning round R, fine-tuning occurs over F rounds, encompassing both the base and head as

$$\theta_i^{(R+F)} = \theta_i^{(R+F-1)} - \eta \nabla_{\theta_i} \mathcal{L}(D_i, \theta_{i,b}^{(R+F-1)}, \theta_{i,h}^{(R+F-1)}). \tag{10}$$

The optimal round, denoted as R, is fewer than the total global rounds T. At this optimal round R, fine-tuning is performed using the base and head layers for as many rounds as F.

Algorithm 1 Federated Learning with Base and Personal Layer Fine-tuning

1: **Input:** Total global rounds T, Total clients N, participate ratio r, Fine tuning rounds F, Learning rate η, window size w, epsilon ϵ
2: **Initialize:** Base parameter $\theta_{i,b}^{(0)}$ and head parameter $\theta_{i,h}^{(0)}$
3: **for** each round $t = 1, 2, \ldots, T$ **do**
4: select clients with participate ratio $M = r \times N$
5: **for** selected client $i = 1, 2, \ldots, M$ **do**
6: $\theta_i^{(t)} = \theta_i^{(t-1)} - \eta \nabla_{\theta_i} \mathcal{L}(D_i, \theta_{i,b}^{(t-1)}, \theta_{i,h}^{(0)})$
7: **end for**
8: $\theta_G^{(t)} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \theta_i^{(t)}$
9: $\bar{a}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} a_i^{(t)}$
10: Compute standard deviation for all clients during recent window rounds
11: $\sigma_w^{(t)} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (a_i^{(t)} - \bar{a}_w^{(t)})^2}$
12: **if** $\sigma_w'(t) < 0$ and $|\sigma_w''(t)| \approx \epsilon$ **then**
13: **break**
14: **end if**
15: **end for**
16: **for** in R round, fine tune round $f = 1$ to F **do**
17: **for** For all clients $i = 1, 2, \ldots, N$ **do**
18: Fine-tune with base and personal layers:
19: $\theta_i^{(R+f)} = \theta_i^{(R+f-1)} - \eta \nabla_{\theta_i} \mathcal{L}(D_i, \theta_{i,b}^{(R+f-1)}, \theta_{i,h}^{(R+f-1)})$
20: **end for**
21: $\theta_{i,G}^{(R+f)} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \theta_i^{(R+f)}$
22: **end for**

Algorithm 1 outlines the training procedure. Line 6 represents the local update, Line 8 indicates the global model aggregation, and Lines 9–11 compute the standard deviation of the clients' accuracy during the evaluation step. The training is terminated at round R which satisfies the condition in Line 12. Lastly, Lines 16–20 conduct fine-tuning using each client's head and base layers.

V. Experiments

Experiments were conducted using the MNIST, CIFAR-10, and CIFAR-100 datasets but mainly foucsed on CIFAR-100. For the baseline, we employed FedAvg [2], FedPer [8], LG-FedAvg [9], FedRep [10], FedROD [11], and FedBABU [12]. We utilized a shallow CNN model consisting of two convolutional layers and two fully connected layers, which is simpler than the baseline models. To generate Non-IID data, we sampled each client's data using a Dirichlet distribution and set the Dirichlet parameter $\alpha = 0.1$ to ensure a highly skewed

distribution. For ease of visualization of the client's data distribution, we displayed the results from sampling 20 clients from the CIFAR-10 dataset using the Dirichlet distribution ($\alpha = 0.1$) in **Fig. 1**.

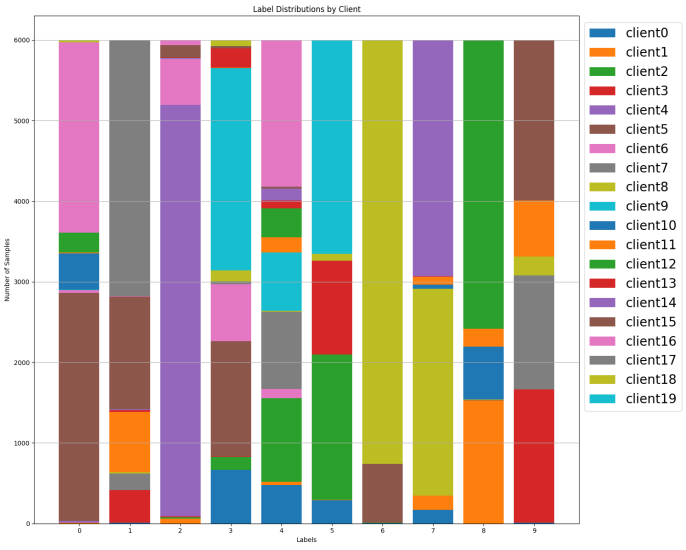

Fig. 1. This illustrates the distribution of the CIFAR-10 dataset for 20 clients generated randomly from a Dirichlet distribution ($\alpha = 0.1$). Both the class and data distribution are notably heterogeneous.

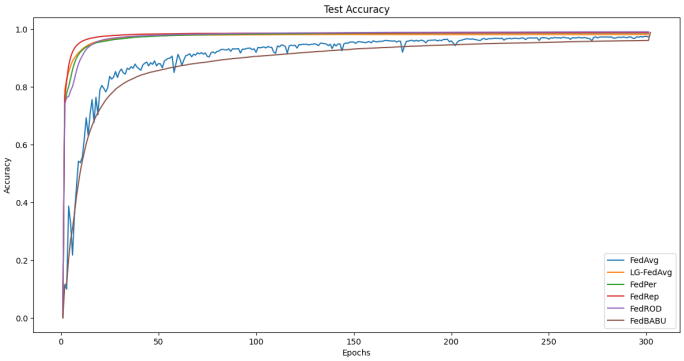

Fig. 2. Baseline Accuracies in MNIST. While the data heterogeneity was sufficient, due to the nature of the data, a lack of class heterogeneity was observed. This led to a convergence of all baseline accuracies to approximately 100% even before personalization.

For MNIST, CIFAR-10, and CIFAR-100, the total number of clients N was set to 100. We used a batch size of 10, a learning rate of 0.005, and a client join ratio of 0.1, ensuring that $k = 10$ clients participated in each round. The total global rounds T were set to 300 for MNIST and CIFAR-10, while it was set to 1000 for CIFAR-100. In the MNIST (Fig. 2) and CIFAR-10 (Fig. 3) results, all personalized Federated Learning algorithms performed relatively well. Thus, due to the significant accuracy difference in the CIFAR-100 (Fig. 4), we focus primarily on experiments with CIFAR-100 dataset.

In Fig. 4, given the highly heterogeneous dataset, the accuracy of FedBABU, which did not employ a head, was

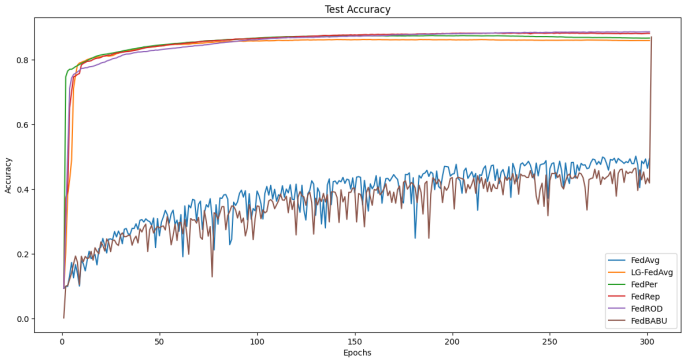

Fig. 3. Baseline Accuracies in CIFAR-10. In the case of the CIFAR-10 dataset, although data heterogeneity was sufficient, the similarity in class characteristics resulted in similar accuracies across all baselines, except for FedAvg.

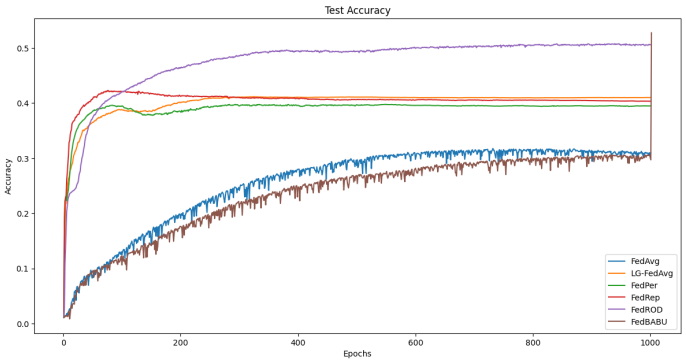

Fig. 4. Baseline Accuracies in CIFAR-100. The CIFAR-100 dataset's diverse data and class heterogeneity led to varied accuracies post-personalization. FedROD [11] improved accuracy with a new loss function for class heterogeneity. FedBABU [12] used only the base layer for updates in heterogeneous environments, excluding the head layer to reduce noise.

observed to be the highest. However, existing studies also empirically used a fixed round, the optimality of which as the learning termination point remains uncertain.

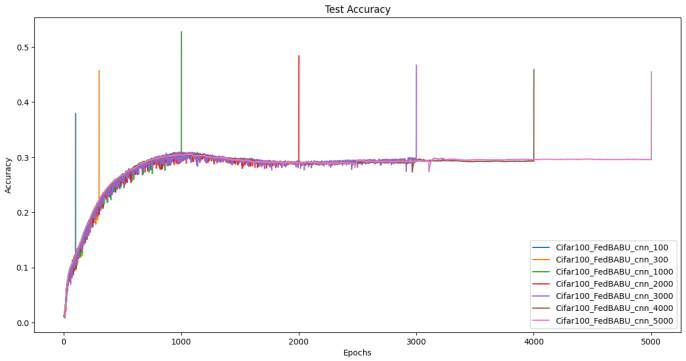

Fig. 5. In the CIFAR-100 dataset, as the accuracy of FedBABU is observed, the accuracy begins to decline after a certain number of rounds.

As observed in Fig. 5, the accuracy tends to increase with training progress but eventually decreases. This underscores the importance of identifying the optimal fine-tuning point.

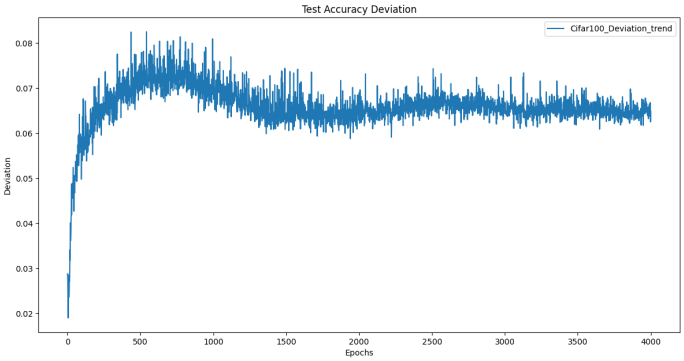

Fig. 6. Deviation Trend in CIFAR-100. To determine an appropriate termination point, upon examining the trend, we found that the point where $\sigma' > 0$ and $\sigma'' \approx \epsilon$ corresponds to a global round between 1000 and 1500.

Fig. 6 illustrates the standard deviation of test accuracy in the CIFAR-100 dataset. The deviation (σ) first increases ($\sigma' > 0$), then decreases ($\sigma' < 0$), and subsequently stabilizes with minimal fluctuations ($|\sigma''| \approx \epsilon$). Based on this observation, we used the condition in Eq. (9) to determine the optimal fine-tuning round.

TABLE II
COMPARISON OF ACCURACY

	MNIST		CIFAR-10		CIFAR-100	
	Acc. (%)	T	Acc. (%)	T	Acc. (%)	T
FedAvg [2]	91.48	100	36.87	100	29.04	500
	96.32	200	42.00	200	30.95	1000
	97.48	300	49.36	300	30.34	1500
FedPer [8]	98.30	100	86.16	100	40.22	500
	98.63	200	87.34	200	41.03	1000
	98.69	300	86.66	300	40.05	1500
LG-FedAvg [9]	97.95	100	85.75	100	41.06	500
	98.17	200	86.10	200	41.01	1000
	98.13	300	85.91	300	40.80	1500
FedRep [10]	98.56	100	86.58	100	40.44	500
	98.69	200	88.12	200	41.24	1000
	98.75	300	88.11	300	40.66	1000
FedROD [11]	98.61	100	86.24	100	49.50	500
	98.95	200	88.01	200	48.70	1000
	99.06	300	**88.59**	300	50.56	1500
FedBABU [12]	98.19	100	85.77	100	49.28	500
	98.77	200	86.16	200	52.75	1000
	98.84	300	87.07	300	51.82	1500
Ours	98.49($\pm$0.11)	135($\pm$16)	86.82($\pm$0.16)	219($\pm$18)	**53.60**($\pm$0.36)	1121($\pm$81)

VI. CONCLUSION

In this study, we propose a technique that tracks standard deviation in representation learning, a method of personalized federated learning designed to address data heterogeneity, to find the optimal fine-tuning point. We empirically determined that excessive training of the base layer results in decreased accuracy after fine-tuning. Noting that the minimal changes in accuracy and loss during the base layer's training can obscure the optimal fine-tuning point, we suggest identifying the fine-tuning point by using the standard deviation of client accuracy. Moreover, we allocated highly heterogeneous data to each client and visually illustrated the sharp changes in accuracy after fine-tuning, which was not addressed in prior research.

VII. FUTURE WORK

In the current paper, the focus has been on data heterogeneity. However, based on the experimental results, it can be observed that in datasets with a limited number of classes, there is only a marginal difference in accuracy after personalization. Therefore, it appears crucial to consider both data and class heterogeneity in future work. This paper has utilized widely used image classification datasets such as MNIST, CIFAR-10, and CIFAR-100, where datasets with only 10 classes exhibit minimal class heterogeneity and limited accuracy variation post-personalization. As a result, in future research, we plan to compare datasets with more than 100 classes, including CIFAR-100, Imagenet, and Tiny-Imagenet, to account for the diversity of classes.

ACKNOWLEDGMENT

This research was supported by the MSIT Korea under the NRF Korea (NRF-2022R1A2C4001270) and the Information Technology Research Center (ITRC) support program (IITP-2022-2020-0-01602) supervised by the IITP.

REFERENCES

[1] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[2] McMahan, Brendan, "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.
[3] Li, Tian, et al. "Federated optimization in heterogeneous networks." Proceedings of Machine learning and systems 2 (2020): 429-450.
[4] Karimireddy, Sai Praneeth, et al. "Scaffold: Stochastic controlled averaging for federated learning." International conference on machine learning. PMLR, 2020.
[5] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." International conference on machine learning. PMLR, 2017.
[6] Nichol, Alex, Joshua Achiam, and John Schulman. "On first-order meta-learning algorithms." arXiv preprint arXiv:1803.02999 (2018).
[7] Fallah, Alireza, Aryan Mokhtari, and Asuman Ozdaglar. "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach." Advances in Neural Information Processing Systems 33 (2020): 3557-3568.
[8] Arivazhagan, Manoj Ghuhan, et al. "Federated learning with personalization layers." arXiv preprint arXiv:1912.00818 (2019).
[9] Liang, Paul Pu, et al. "Think locally, act globally: Federated learning with local and global representations." arXiv preprint arXiv:2001.01523 (2020).
[10] Collins, Liam, et al. "Exploiting shared representations for personalized federated learning." International conference on machine learning. PMLR, 2021.
[11] Chen, Hong-You, and Wei-Lun Chao. "On bridging generic and personalized federated learning for image classification." arXiv preprint arXiv:2107.00778 (2021).
[12] Oh, Jaehoon, Sangmook Kim, and Se-Young Yun. "Fedbabu: Towards enhanced representation for federated image classification." arXiv preprint arXiv:2106.06042 (2021).