Supplementary

The supplementary is organized in the following sections:

- Section A: More details about the experiments.
- Section B: Connection and comparison between APFL [2] and FedPerfix.
- Section C: More hyperparameter ablations of FedPerfix.

A. Experiment Details

A.1. Visualization of Data Partitioning

We partition the data in each dataset with Dirichlet distributions, which is a common setting in previous work [9, 6]. The details of the distributions are visualized in Figure 1. Specifically, the number of clients N, number of classes C, and the parameter of Dirichlet distribution α for each dataset are as follows:

- CIFAR-100 [5]: $N = 64, C = 100, \alpha = 0.1$.
- OrganAMNIST [12]: $N = 64, C = 11, \alpha = 0.5$.
- Office-Home [11]: $N = 16, C = 65, \alpha = 1.0$.

A.2. Hyperparameters

We perform experiments based on the implementation from an existing federated learning platform. For each method, we tune the hyperparameters in a range and report the result with the optimal hyperparameters. The range and optimal value of the hyperparameters are as follows:

FedAVG [8]: Learning rate (lr) is searched from $\{0.001, 0.01, 0.1\}$. The optimal value is 0.01.

Local: lr is searched from $\{0.001, 0.01, 0.1\}$. The optimal value is 0.01.

APFL [2]: lr and the initial mixture coefficient α are searched from $\{0.001, 0.01, 0.1\}$ and $\{0.25, 0.5, 0.75\}$, the optimal values are $lr = 0.01$ and $\alpha = 0.25$.

Per-FedAVG [3]: lr and β are searched from $\{0.001, 0.01, 0.1\}$ and $\{0.001, 0.01, 0.1\}$, the optimal values are $lr = 0.001$ and $\beta = 0.001$.

FedBN [7]: lr is searched from $\{0.001, 0.01, 0.1\}$. The optimal value is 0.01.

FedRep [1]: lr is searched from $\{0.001, 0.01, 0.1\}$. The optimal value is 0.01. The classification head is defined as the last layer.

FedBABU [10]: lr is searched from $\{0.001, 0.01, 0.1\}$. The optimal value is 0.01. One local step is done for fine-tuning the classification head. The classification head is defined as the last layer.

FedPerfix: lr is searched from $\{0.001, 0.01, 0.1\}$, and the optimal value is 0.01. The hidden state dimension is set as 256. The scale s is set as 1.5. The classification head is defined as the last two layers.

B. Connection between APFL and FedPerfix

In this section, we provide a detailed comparison between the APFL and FedPerfix to explain *why they both have the leading performance* compared with other methods and show that our method has *additional advantages in storage and computation resource requirements*.

B.1. Why APFL and FedPerfix lead the performance?

First, we briefly introduce the idea of APFL. APFL keeps a separate personalized model for each client. In each communication round, it will first train the global and local models separately and obtain their gradients, then update the personalized model with the gradient mixed from these two models. Now if we only consider the personalized model, its update can be written as

$$\begin{aligned} h_{per} &\leftarrow h_{per} - \eta(\alpha \nabla h_{per} + (1-\alpha)\nabla h_{global}) \\ &= h_{per} - \eta \nabla(\alpha h_{per} + (1-\alpha) h_{global}), \end{aligned} \tag{1}$$

where h_{per} is the personalized model, h_{global} is the global model, η is the learning rate, and α is the mixture coefficient.

Therefore, the updating of the personalized model is equivalent to updating a model $\bar{h}$ that takes the mixture of the personalized and global models as the output. Further, we formulate the output of $\bar{h}$ as

$$\boldsymbol{O}^{(L)} = \alpha h_{per}^{(L)}(\boldsymbol{Z}^{(L-1)}) + (1-\alpha) h_{global}^{(L)}(\boldsymbol{Z}^{(L-1)}), \tag{2}$$

where L is the number of layers of the model, $\boldsymbol{Z}^{(L-1)}$ is the hidden state from the last layer.

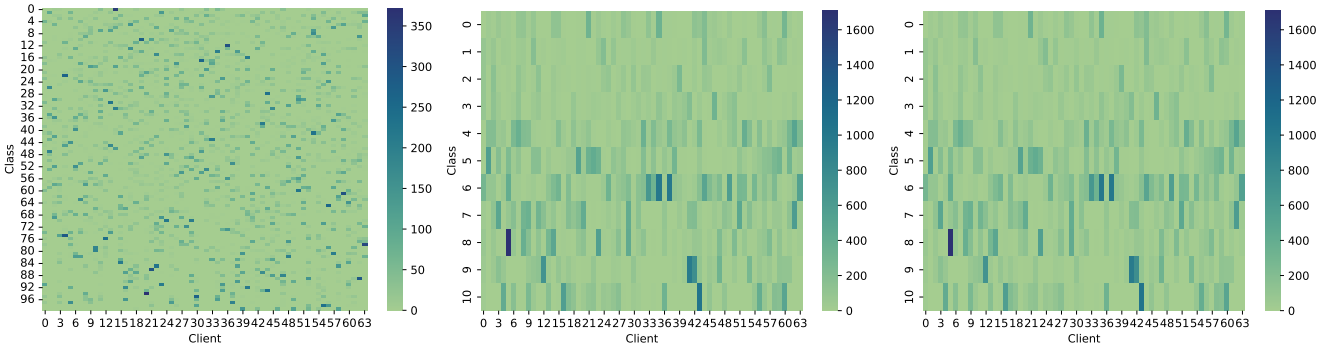
For FedPerfix, the output of one head of the self-attention layer can be formulated and rewritten [4] as

$$\begin{aligned} head(\boldsymbol{Z}) =& Attn(\boldsymbol{Z}\boldsymbol{W}_q, \boldsymbol{Z}[\boldsymbol{P}_k, \boldsymbol{W}_k], \boldsymbol{Z}[\boldsymbol{P}_v, \boldsymbol{W}_v]) \\ =& \text{softmax}(\boldsymbol{Z}\boldsymbol{W}_q[\boldsymbol{P}_k, \boldsymbol{W}_k]^\top) \begin{bmatrix} \boldsymbol{P}_v \\ \boldsymbol{Z}\boldsymbol{W}_v \end{bmatrix} \\ =& (1-\lambda(\boldsymbol{Z}))\text{softmax}(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{W}_k^\top\boldsymbol{Z}^\top)\boldsymbol{Z}\boldsymbol{W}_v \\ &+ \lambda(\boldsymbol{Z})\text{softmax}(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{P}_k^\top)\boldsymbol{P}_v \\ =& (1-\lambda(\boldsymbol{Z}))Attn(\boldsymbol{Z}\boldsymbol{W}_q, x\boldsymbol{W}_k, x\boldsymbol{W}_v) \\ &+ \lambda(\boldsymbol{Z})Attn(\boldsymbol{Z}\boldsymbol{W}_q, \boldsymbol{P}_k, \boldsymbol{P}_v) \end{aligned} \tag{3}$$

where $Attn$ is the attention operation, Z is the hidden state from the last layer, and $\lambda(\boldsymbol{Z})$ is the mixture coefficient defined as

$$\lambda(\boldsymbol{Z}) = \frac{\sum_i \exp(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{P}_k^\top)_i}{\sum_i \exp(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{P}_k^\top)_i + \sum_j \exp(\boldsymbol{Z}\boldsymbol{W}_q\boldsymbol{W}_k^\top\boldsymbol{Z}^\top)_j}. \tag{4}$$

If only taking the self-attention layer into consideration, Equation 2 and Equation 3 share a similar formulation,



(a) CIFAR-100 (b) OrganAMNIST (c) Office-Home

Figure 1. Data Partitioning of each dataset.

which can be interpreted as an information transfer between the global and local models. Therefore, both approaches have a leading performance, indicating that *the underlying idea to balance local and global information is effective and crucial in personalized federated learning*.

B.2. Advantages of FedPerfix

Although APFL and FedPerfix share a connected underlying idea, FedPerfix can outperform APFL in a consistent margin across different settings. Besides, FedPerfix is much more efficient than APFL from several perspectives:

- Parameter size of the Prefixes in FedPerfix (**289.0K**) is fewer than a separate self-attention layer (**577.5K**) in APFL, leading to fewer computational costs for the self-attention layer.
- APFL needs to perform the mixture between the global and local models with additional computational costs for every layer, while FedPerfix only needs to perform it for the self-attention layers.
- APFL needs to store a separate personalized model (**21.03M**) on each client, while FedPerfix only needs (**3.39M**) additional space.

In conclusion, APFL needs $\mathbf{70}\times$ additional FLOPs and $\mathbf{6.2}\times$ additional parameters to store than FedPerfix, using FedAVG as a baseline. Therefore, when compared with APFL, FedPerfix not only achieves superior performance but also enjoys a more efficient implementation by specifying and focusing only on the sensitive parts of the ViT. This targeted approach leads to a more effective and efficient approach for personalized federated learning.

C. More Ablation Study

To reduce the exhausting hyperparameter-searching in practice, we report the result under a unified default setting for all tasks in the main paper. *The performance under such a unified default setting still achieves state-of-the-art performance*, demonstrating the robustness of our proposed method. However, we still want to demonstrate the potential to further improve the performance by tuning the hyperparameters. In this section, we will show the results and analyze the impact of three key hyperparameters in FedPerfix: hidden dimension, scale, and the depths of prefixes.

C.1. Impact of Hidden Dimension and Scale

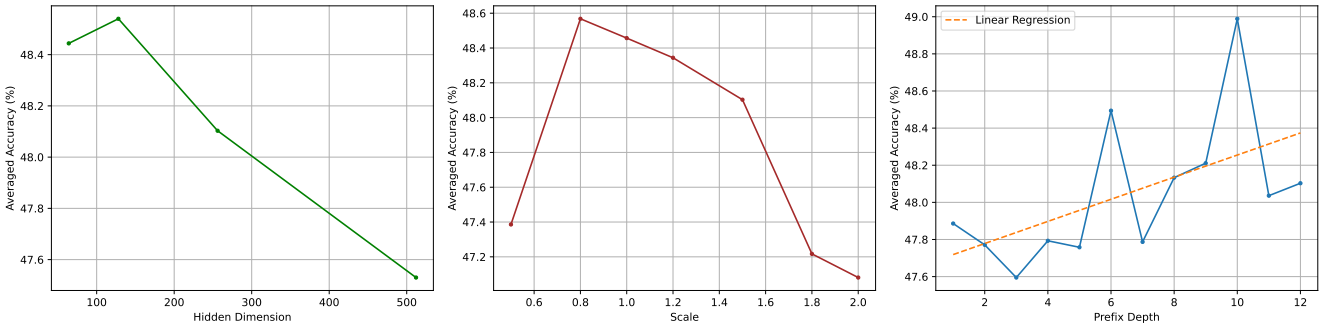
The hidden dimension is the dimension of the hidden state of the adapter to generate the Prefixes, *i.e.*, the common dimension shared by $\boldsymbol{W}_{down}$ and $\boldsymbol{W}_{up}$, and the scale s is scalar to control the impact of the Prefixes. In our default setting, the hidden dimension is set as 256, and the scale is set as 1.5. FedPerfix, under the default setting, can outperform all the compared methods in all datasets. Here, we demonstrate the potential to achieve better performance on CIFAR-100 by tuning the hyperparameters. As shown in Figure 2 (a) and (b), increasing the hidden dimension and scale will not always lead to an increase in performance. Therefore, in practice, further tuning the hidden dimension and the scale can lead to a better result, even though without such tuning can still maintain a high performance.

C.2. Impact of the Depths of Prefixes

In our default setting, we add the Prefixes to every self-attention layer. To investigate the impact of the depths of Prefixes, we further conduct experiments only to add Prefixes to the last several self-attention layers, *i.e.*, $depth = 1$ means only adding Prefixes to the last self-attention layer. The result is shown in Figure 2 (c). In general, with the increase in the depths of the Prefixes, the overall performance increases. However, such an increase is not strict, indicating the potential to further improve the performance and reduce storage and computational costs.

References

[1] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting Shared Representations for Personalized Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2089–2099. PMLR, July 2021. ISSN: 2640-3498. 1



(a) Hidden Dimension vs. Performance (b) Scale vs. Performance (c) Prefix Depth vs. Performance

Figure 2. Impact of each hyperparameter on CIFAR-100.

[2] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive Personalized Federated Learning, Nov. 2020. arXiv:2003.13461 [cs, stat]. 1

[3] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized Federated Learning: A Meta-Learning Approach, Oct. 2020. arXiv:2002.07948 [cs, math, stat]. 1

[4] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a Unified View of Parameter-Efficient Transfer Learning. Feb. 2022. 1

[5] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 2012. 1

[6] Qinbin Li, Bingsheng He, and Dawn Song. Model-Contrastive Federated Learning, Mar. 2021. arXiv:2103.16257 [cs]. 1

[7] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization, May 2021. arXiv:2102.07623 [cs]. 1

[8] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, Feb. 2017. arXiv:1602.05629 [cs]. 1

[9] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning, Apr. 2022. arXiv:2111.14213 [cs]. 1

[10] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. FedBABU: Towards Enhanced Representation for Federated Image Classification, Mar. 2022. arXiv:2106.06042 [cs]. 1

[11] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1

[12] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023. Publisher: Nature Publishing Group UK London. 1