PGFed: Personalize Each Client's Global Objective for Federated Learning – Supplementary Material

Jun Luo[1] Matias Mendieta[2] Chen Chen[2] Shandong Wu[1,3,4,5]
[1] Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA
[2] Center for Research in Computer Vision, University of Central Florida, Orlando, FL
[3] Department of Radiology, University of Pittsburgh, Pittsburgh, PA
[4] Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA
[5] Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA
jul117@pitt.edu ma584394@ucf.edu chen.chen@crcv.ucf.edu wus3@upmc.edu

A. Overview

We organize the supplementary material as follows:

- In Appendix B, we report additional results and analyses of PGFed and conduct experiments on two other datasets, OrganAMNIST [2] & Office-home [1] with different FL settings.
- In Appendix C, we compare the local computational speed of the proposed algorithm, PGFed, with the baselines that achieved top performance in the experiments on CIFAR10 and CIFAR100.
- In Appendix D, we further propose PGFed-CE, a variation on top of PGFed to reduce both the communication and the computation cost simultaneously.
- In Appendix E, we report details in hyperparameters regarding our experiments.

B. Additional Experiments and Analyses

B.1. Convergence Behavior

We empirically study the convergence behavior of PGFed and the baselines that achieved high performance on CIFAR10 and CIFAR100. For each method, we plot its mean personalized test accuracy on CIFAR10 for the first 150 rounds of training under 25-, 50- and 100-client settings, as shown in Fig. 1.

From the results, we can see that, while achieving the highest accuracy performance, PGFed is also able to consistently converge faster than most of the baselines that reach high accuracies. Fast as it is under these settings, one limitation of PGFed is that the update of α_{ij} in PGFed only happens when client i and client j are selected in two consecutive rounds (client j is selected exactly one round before client i), which happens by chance. Therefore, this randomness might slightly limit the overall convergence behavior of PGFed, but it is the existence of α_{ij}'s that enables the adaptive personalization of how much each client values other clients' empirical risks, hence the higher accuracy.

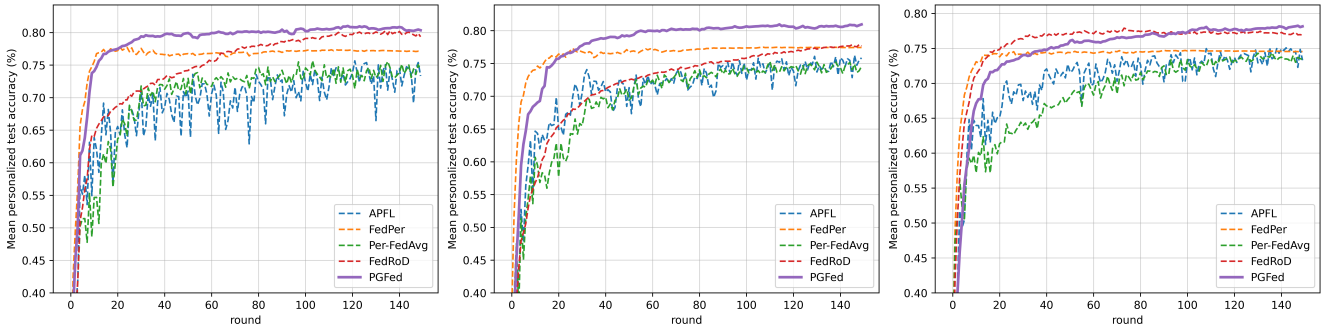
B.2. Experiments on Other Datasets

To further evaluate the effectiveness of PGFed with different types of data and different FL settings, we conduct experiments on two more datasets: OrganAMNIST [2] and Office-home [1]. OrganAMNIST is a medical imaging dataset of abdominal CT images with 11 classes. Office-home [1] contains four domains (Art, Clipart, Product, and Real World) of images depicting 65 classes of objects typically found in Office and Home settings.

	25 clients sample 50% Dir(1.0)	50 clients sample 25% Dir(0.3)	100 clients sample 25% Dir(0.3)
Local	90.45±0.19	90.63±0.07	87.14±0.10
FedAvg	99.11±0.03	98.74±0.04	98.47±0.08
APFL	97.49±0.05	97.53±0.06	96.19±0.11
FedRep	95.06±0.16	94.86±0.07	92.47±0.04
LGFedAvg	90.47±0.18	90.99±0.08	87.52±0.22
FedPer	97.89±0.06	97.55±0.08	95.56±0.33
Per-FedAvg	98.40±0.02	96.80±0.04	95.09±0.07
FedRoD	98.61±0.05	98.14±0.09	97.05±0.06
FedBABU	96.49±0.28	94.33±0.13	91.07±0.23
PGFed	99.20±0.04	**99.17±0.05**	**98.94±0.02**
PGFedMo	**99.21±0.04**	99.17±0.07	98.86±0.06

Table 1. Mean and standard deviation over three trials of the mean personalized test accuracy (%) on OrganAMNIST

For OrganAMNIST, we adopt three settings with different numbers (25, 50, 100) of clients. For the 50- and 100-



(a) 25 clients (b) 50 clients (c) 100 clients

Figure 1. Convergence behavior of the personalized FL approaches with top performance on CIFAR10. While achieving the highest accuracy performance, PGFed is also able to consistently converge faster than several of the baselines that reach high accuracies.

	Art	Clipart	Product	Real World	Mean
Local	17.16 ± 0.85	37.65 ± 0.47	43.83 ± 0.40	24.50 ± 0.21	30.79 ± 0.23
FedAvg	11.68 ± 1.26	41.29 ± 0.85	42.49 ± 1.28	19.14 ± 0.89	28.65 ± 0.49
APFL	19.11 ± 1.55	44.67 ± 0.61	$\mathbf{50.40 \pm 0.56}$	25.85 ± 0.88	35.00 ± 0.41
FedRep	20.24 ± 1.45	38.43 ± 1.02	43.70 ± 1.04	24.02 ± 0.81	31.60 ± 0.05
LGFedAvg	17.54 ± 0.45	38.75 ± 0.13	44.59 ± 0.62	25.79 ± 0.61	31.67 ± 0.21
FedPer	17.83 ± 1.07	38.97 ± 0.35	45.87 ± 0.13	25.01 ± 0.52	31.92 ± 0.24
Per-FedAvg	14.62 ± 0.40	39.94 ± 1.29	44.40 ± 1.32	21.58 ± 0.65	30.13 ± 0.07
FedRoD	19.67 ± 1.23	42.44 ± 0.77	44.34 ± 2.07	24.28 ± 1.69	32.68 ± 0.69
FedBABU	18.18 ± 3.54	42.10 ± 2.31	43.51 ± 0.91	$\mathbf{26.81 \pm 1.86}$	33.38 ± 0.29
PGFed	$\mathbf{22.40 \pm 0.26}$	$\mathbf{46.48 \pm 1.00}$	$\underline{49.86 \pm 2.14}$	26.04 ± 0.80	$\mathbf{36.19 \pm 0.92}$
PGFedMo	$\underline{22.16 \pm 0.45}$	$\underline{45.88 \pm 0.83}$	49.45 ± 0.19	$\underline{26.60 \pm 0.99}$	$\underline{36.02 \pm 0.20}$

Table 2. Mean and standard deviation over three trials of the mean personalized accuracy% of the four domains (5 clients/domain) and the average performance on Office-home dataset. The highest and second-highest accuracies under each setting are in **bold** and underlined, respectively.

client settings, we follow the same setting as in the experiments on CIFAR10/CIFAR100, and use the Dirichlet distribution with $\alpha = 0.3$ (Dir(0.3)) and 25% client sample rate for each round. For the 25-client setting, we reduce the heterogeneity in the dataset via Dir(1.0) distribution, and use a higher client sample rate (50%) to simulate a situation more similar to cross-silo FL settings. For Office-home, we adopt a 20-client setting, where each domain contains 5 clients. The non-IIDness in each domain is achieved by Dir(0.3). The mean personalized test accuracies of each domain and over the whole federation are reported. We compare the proposed PGFed and PGFedMo against Local, FedAvg, and the personalized FL baselines that achieved high performance in previous experiments on CIFAR10 and CIFAR100. The results are shown in Tab. 1 and Tab. 2.

For OrganAMNIST, PGFed and PGFedMo achieve the best performance under all three settings. In addition, the proposed algorithms do not have an obvious drop in the performance from the less heterogeneous 25-client setting to 50-client and 100-client settings. This is not the case for many other personalized FL base-

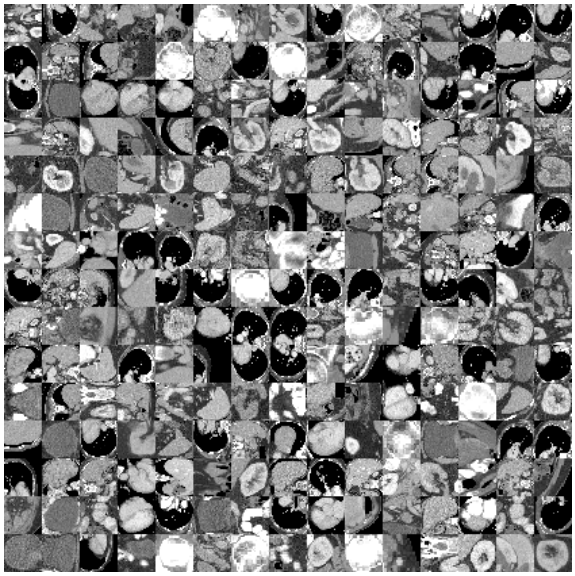


Figure 2. OrganAMNIST [1] image samples.

lines (FedPer, Per-FedAvg, FedRoD, and FedBABU). Moreover, FedAvg achieves excellent performance on OrganAMNIST due to the similarity of clients' images (see Fig. 2). Since the Dirichlet distribution can only differ the clients in $P(y)$, the label distribution, instead of $P(x|y)$, the distribution of the images given the label, a simple averaging might work just fine on OrganAMNIST compared with other datasets. On the contrary, Office-home is a dataset that addresses different $P(x|y)$ since for this dataset, even the images within the same class can be dramatically different if they belong to clients from different domains, which is indicated by the worse performance of FedAvg than Local training. Results on this dataset also show that PGFed and PGFedMo consistently outperform most of the compared methods in each domain, and achieve the highest mean accuracies over all domains, demonstrating their superiority over all the compared methods.

C. Comparison in Local Computational Speed

In this section, we study the local computational speed of PGFed and the baselines that achieved top performance in the experiments. We measure the local computational speed by the number of images each method can process per second, and report the local computational speed of the methods in Tab. 3 on CIFAR10 with 50 clients and a batch size of 128, using an NVIDIA Tesla V100 GPU and an Intel(R) Xeon(R) Gold 6248 CPU.

	Images/s	Relative speed	Accuracy
FedAvg	6917.1	100.00%	64.41±0.66
APFL	3389.8	48.99%	77.36±0.18
Per-FedAvg	3464.5	50.09%	76.27±0.50
FedRoD	6682.4	96.61%	79.61±0.22
PGFed	6120.0	88.48%	81.42±0.31
PGFedMo	6032.8	87.22%	81.48±0.32
PGFed-CE*	6175.5	89.28%	81.16±0.56

* A more communication-efficient variation of PGFed, introduced in Appendix D

Table 3. Computational speed (in terms of "images/s") and accuracy on CIFAR10 with 50 clients

From the results, we can see that PGFed not only reaches high accuracy, but has a relatively high computational speed as well. With a batch size of 128, PGFed reaches a speed equivalent to 88.48% of FedAvg's speeds. PGFedMo is slightly slower than PGFed due to the momentum update of the auxiliary gradient. However, for some of the compared methods that also achieve high accuracy, their computational speed is compromised by around 50% (compared to FedAvg): APFL needs to train a global model and a local adapter, while Per-FedAvg leverages meta-learning which is a bi-level optimization problem. These methods either train two models or conduct twice gradient descent for each iteration. FedRoD trains a global model and a local classifier, which ends up being 8.13% faster than PGFed. For PGFed, the extra local computation (over FedAvg) happens at the addition of the gradients from both the local empirical risk and the auxiliary risk, and at the update of $\boldsymbol{\alpha}_i$ where a dot product of vectorized models is calculated (see Eq.(12) in the main paper).

D. PGFed-CE, a More Communication- and Computation-efficient PGFed

As mentioned in Sec. 6 of the main paper, although PGFed manages to circumvent the seemingly unavoidable $O(N^2)$ communication cost, and achieve asymptotically the same communication cost as FedAvg ($O(N)$), since each client is required to download three and upload two models/gradients per round, on average the communication cost is still high (roughly 2.5 times as much as that of FedAvg).

In this section, we provide a more communication-efficient version of PGFed, dubbed PGFed-CE that downloads one less gradient from the server. In Sec. 4 of the main paper, we mentioned that $g_\alpha^{(2)}$, a portion of the gradient of the local objective in terms of $\boldsymbol{\alpha}_i$ in PGFed, could be computed on the server instead of the client. This is because

$$g_\alpha^{(2)} = \mu \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T \boldsymbol{\theta}_i, \tag{1}$$

and the server has both $\nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)$ from the previous round and the current round global model as the initialization of $\boldsymbol{\theta}_i$. Therefore, it is possible to treat $g_\alpha^{(2)} = \mu \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T \boldsymbol{\theta}_{glob}$ as a constant computed by the server, where the global model is used as an estimation of $\boldsymbol{\theta}_i$ for the whole round. Since α_i should change adaptively according to the change of $\boldsymbol{\theta}_i$ during local training, using the global model as a fixed estimation is not ideal. Nonetheless, this variation saves the communication cost by the size of one gradient ($\bar{\boldsymbol{g}}_{\mathcal{S}_t}$) from clients' round-beginning download, that *was* needed to adaptively compute $\boldsymbol{\alpha}_i$ locally, which also slightly saves the local computation.

We name this more communication- and computation-efficient variant of PGFed as PGFed-CE. In PGFed-CE, each client is now required to download two, instead of three, models/gradients per round. Consequently, on average, the communication cost is reduced from 2.5 to 2 times as much as that of FedAvg. In addition, we follow Appendix C and report the local computational speed of PGFed-CE and its performance on CIFAR10 with 50 clients in Tab. 3. As expected, besides the reduced communication cost, PGFed-CE also simultaneously increases the local computational speed with little drop in the accuracy.

E. Hyperparameters

Besides the hyperparameter tuning of PGFed reported in Sec. 4.1 in the main paper, we further report the hyperparameter tuning of the compared baselines in this section. The learning rate of all baselines is tuned from $\{0.1, 0.01, 0.001, 0.0001\}$. For FedDyn, we tuned the α in $\{0.1, 0.01, 0.001\}$. For APFL, the α is tuned in $\{0.25, 0.50, 0.75\}$. For Per-FedAvg, the two learning rates for each step are selected from $\{0.1, 0.01, 0.001, 0.0001\}$, which is the same for FedBABU's learning rates for federated training and fine-tuning step.

References

[1] 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, 2

[2] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. 1