Supplementary Material for Efficiently Assemble Normalization Layers and Regularization for Federated Domain Generalization

Khiem Le[1], Long Ho[2], Cuong Do[2], Danh Le-Phuoc[3], Kok-Seng Wong[2,*]
[1] Department of Computer Science and Engineering, University of Notre Dame, IN, USA
[2] College of Engineering and Computer Science, VinUniversity, Hanoi, Vietnam
[3] Open Distributed Systems, Technical University Berlin, Berlin, Germany
kle3@nd.edu, danh.lephuoc@tu-berlin.de, {long.ht, cuong.dd, wong.ks}@vinuni.edu.vn

A. Code Availability

Our source code for running experiments was built originally using PyTorch [4] and Flower [1] frameworks and is attached along with supplementary material.

B. Additional Details for Section 4.2

As we mentioned in the paper, the hyper-parameter of our regularization term λ is searched in the range [0, 1] with a step of 0.25. Here we provide the results of different λ on the PACS dataset. As shown in Table 1, we sequentially set the value of λ as 0.00, 0.25, 0.50, 0.75, 1.00 and observed that the average performance can be improved consistently with different λ. The best results are found at λ is 0.50. Note that when λ is set as 0.00, the guiding regularizer is not used during training, which means that gPerXAN becomes PerXAN accordingly.

λ in gPerXAN	PACS				
	P	A	C	S	Avg
0.00	96.35	86.28	83.49	82.01	86.91
0.25	96.47	86.13	83.79	83.00	87.35
0.50	97.27	86.52	**84.68**	83.28	**87.94**
0.75	**97.28**	**86.60**	84.37	**83.47**	87.93
1.00	96.45	85.89	83.31	82.41	87.02

Table 1. Impact of λ on the PACS dataset.

C. Additional Ablation Study

To better understand the proposed regularization term's contribution, we now conduct an additional comparison between our regularizer and the one from FedProx [2] on enhancing the proposed normalization scheme PerXAN. As shown in Table 2, FedProx [2] seems to not enhance our PerXAN. This might be due to the difference in the design purpose of regularizers. Also, we argue that our two components, the normalization scheme, and the regularizer, have a solid connection and support to each other during training, hence, gPerXAN shows better results.

Method	PACS				
	P	A	C	S	Avg
PerXAN	96.35	86.28	83.49	82.01	86.91
PerXAN w/ FedProx	95.12	85.74	83.29	82.76	86.73
gPerXAN (Ours)	**97.27**	**86.52**	**84.68**	**83.28**	**87.94**

Table 2. A comparison of two regularizers on the PACS dataset.

D. Additional Analysis

Regarding efficiency, we provide a more detailed analysis, which is even more solid than supporting experiments, to clarify our improvements. Here we denote N, C, and d as the number of clients, number of classes, and dimension of the final feature vector in the model. Notably, we use the number of parameters as a metric for evaluating memory consumption, communication, and computation costs. Assuming training a ResNet-50 model with R parameters, the table below compares the efficiency of COPA [5], FedDG-GA [6], and **gPerXAN (Ours)**. The provided table clearly demonstrates our method has comparable efficiency with FedAvg [3] and is much better compared to others.

*Corresponding author: wong.ks@vinuni.edu.vn

Method	Memory	Additional Cost	
		Communication	Computation
FedAvg	R	R	R
COPA	$R + N(N-1)Cd$	$R + N(N-1)Cd$	$R + N(N-1)Cd$
FedDG-GA	$R \times 2$	R	$R \times 2$
gPerXAN (Ours)	R	R	$R + NCd$

References

[1] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390*, 2020. 1

[2] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 1

[3] 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*, pages 1273–1282. PMLR, 2017. 1

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[5] Guile Wu and Shaogang Gong. Collaborative Optimization and Aggregation for Decentralized Domain Generalization and Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6484–6493, 2021. 1

[6] Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated Domain Generalization with Generalization Adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3954–3963, 2023. 1