Is Normalization Indispensable for Multi-domain Federated Learning?

Weiming Zhuang
Sony AI
weiming.zhuang@sony.com

Lingjuan Lyu
Sony AI
lingjuan.lv@sony.com

ABSTRACT

Federated learning (FL) enhances data privacy with collaborative in-situ training on decentralized clients. Nevertheless, FL encounters challenges due to non-independent and identically distributed (non-i.i.d) data, leading to potential performance degradation and hindered convergence. While prior studies predominantly addressed the issue of skewed label distribution, our research addresses a crucial yet frequently overlooked problem known as multi-domain FL. In this scenario, clients' data originate from diverse domains with distinct feature distributions, as opposed to label distributions. To address the multi-domain problem in FL, we propose a novel method called **Fed**erated learning **W**ith**o**ut **n**ormalizations (FedWon). FedWon draws inspiration from the observation that batch normalization (BN) faces challenges in effectively modeling the statistics of multiple domains, while alternative normalizations possess their own limitations. In order to address these issues, FedWon eliminates all normalizations in FL and reparameterizes convolution layers with scaled weight standardization. Through comprehensive experimentation on four datasets and four models, our results demonstrate that FedWon surpasses both FedAvg and the current state-of-the-art method (FedBN) across all settings, achieving notable improvements of over 10% in certain domains. Furthermore, FedWon is versatile for both cross-silo and cross-device FL, exhibiting strong performance even with a batch size as small as 1, thereby catering to resource-constrained devices. Additionally, FedWon effectively tackles the challenge of skewed label distribution.

CCS CONCEPTS

• **Computing methodologies → Distributed artificial intelligence**; **Distributed computing methodologies**.

KEYWORDS

federated learning, multi-domain, normalization-free

ACM Reference Format:
Weiming Zhuang, Lingjuan Lyu. 2023. Is Normalization Indispensable for Multi-domain Federated Learning?. In *KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA.*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

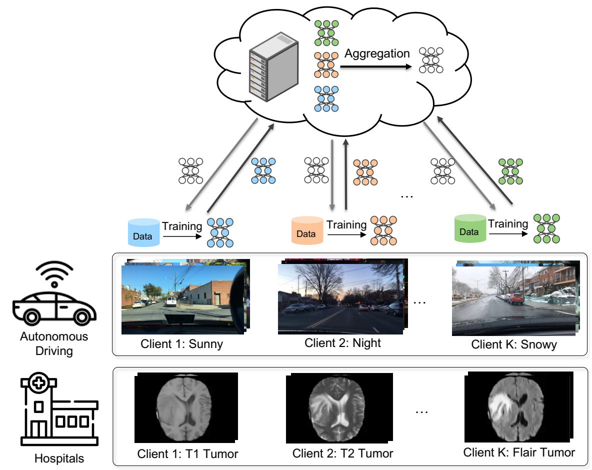
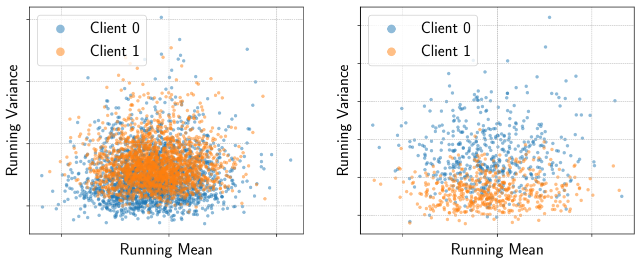


Figure 1: We consider multi-domain federated learning, where each client contains data of one domain. This setting is highly practical and applicable in reality. For example, autonomous cars in distinct locations capture images in varying weather conditions, and healthcare institutions collect medical images with different machines and protocols.

1 INTRODUCTION

Federated learning (FL) has emerged as a promising method for distributed machine learning, enabling in-situ model training on decentralized client data. It has been widely adopted in diverse applications, such as healthcare [3] and autonomous cars [29]. However, FL commonly suffers from non-independent and identically distributed (non-i.i.d) data across clients [18]. This is due to the fact that data generated from different clients is highly likely to have different data distributions, which can cause performance degradation [9] even divergence in training [26, 31].

The majority of studies that address the problem of non-i.i.d data mainly focus on the issue of skewed label distribution, where clients have different label distributions [9, 19]. However, multi-domain FL, where data in clients are from different domains, has received little attention, despite its practicality in reality. Figure 1 depicts two practical examples of multi-domain FL. For example, autonomous cars may collaborate on model training, but their data could originate from different weather conditions or times of day, leading to domain discrepancies in collected images [28]. Similarly, multiple healthcare institutions collaborating on medical imaging analysis may face significant domain gaps due to variations in imaging machines and protocols [3]. Hence, developing solutions for multi-domain FL is a critical research problem with broad implications.



(a) Statistics of 4-th BN Layers. **(b) Statistics of 5-th BN Layer.**

Figure 2: Visualization of batch normalization (BN) channel-wise statistics from two clients, each with data of a single domain. (a) and (b) are the results from 4-th and 5-th BN layer of a 6-layer CNN, respectively. It highlights different feature statistics of BN layers trained on different domains.

However, the existing solutions are unable to adequately address the problem of multi-domain FL. FedBN [20] attempts to solve this problem by keeping batch normalization (BN) [12] parameters and statistics locally in client, but it is only suitable for cross-silo FL [13], where clients are organizations like healthcare institutions, because it requires clients to be stateful [14] (i.e. keeping states of BN information) and participate training every round. As a result, FedBN is not suitable for cross-device FL, where the clients are stateless and only a fraction of clients could be available each round. Besides, BN relies on the assumption that training data are from the same distribution, ensuring the mean and variance of each mini-batch are representative of the entire data distribution [12]. Figure 2a shows that the running mean and variance of BNs in two FL clients from different domains can differ significantly. Alternative normalizations like Layer Norm [2] and Group Norm [27] have not been studied for multi-domain FL, but they have limitations like requiring extra computation in inference.

This paper explores a fundamentally different approach to address multi-domain FL. Given that BN struggles to capture multi-domain data and other normalizations come with their own limitations, we further ask the question: is normalization indispensable to learning a general global model for multi-domain FL? In recent studies, normalization-free ResNets [4] demonstrates comparable performance to standard ResNets[8]. Inspired by these findings, we build upon this methodology and explore its untapped potential within the realm of multi-domain FL.

We introduce **Fed**erated learning **W**ith**o**ut **n**ormalizations (FedWon) to address the domain discrepancies among clients in multi-domain FL. FedWon follows FedAvg [21] protocols for server aggregation and client training. Unlike existing methods, FedWon removes normalization layers and reparameterizes convolution layers with Scaled Weight Standardization [4]. Extensive experiments on four datasets and four models indicate that FedWon outperforms state-of-the-art methods under all settings. The *general global model* trained by FedWon can achieve more than 10% improvement on certain domains compared to the *personalized models* from FedBN [20]. Moreover, our empirical evaluation demonstrated three key benefits of FedWon: 1) FedWon is versatile to support both cross-silo and cross-device FL; 2) FedWon achieves competitive performance on small batch sizes (even on a batch size of 1), which is particularly useful for resource-constrained devices; 3) FedWon can also be applied to address the skewed label distribution problem.

In summary, our contributions are as follows:

- We introduce FedWon, a simple yet effective method for multi-domain FL. By removing normalizations and employing scaled weight standardization, FedWon learns a general global model from clients with significant domain gaps.
- To the best of our knowledge, FedWon is the first method that enables both cross-silo and cross-device FL without relying on any form of normalization. Our study also reveals the unexplored benefits of this method, particularly in the context of multi-domain FL.
- Extensive experiments demonstrate that FedWon outperforms state-of-the-art methods on all datasets and models, and is suitable for training with small batch sizes, which is especially beneficial for cross-device FL.

2 METHOD

This section presents the problem setup of multi-domain FL and introduces FL without normalization to address the problem.

2.1 Problem Setup

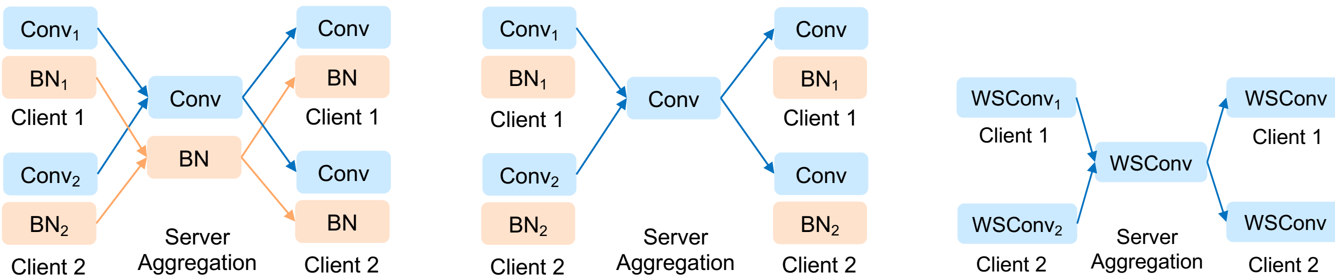
Assume there are N clients in FL and each client k contains $n_k \in \mathbb{N}$ data samples $\{(x_i^k, y_i^k)\}_{i=1}^{n_k}$. Skewed label distribution refers to the scenario where data in clients have different label distributions, i.e. the marginal distributions $\mathcal{P}_k(y)$ may differ across clients ($\mathcal{P}_k(y) \nsim \mathcal{P}_{k'}(y)$ for different clients k and k'). In contrast, this work focuses on multi-domain FL, where clients possess data from various domains, and data samples within a client belong to the same domain [13, 20]. Specifically, the marginal distribution $\mathcal{P}_k(x)$ may vary across clients ($\mathcal{P}_k(x) \nsim \mathcal{P}_{k'}(x)$ for clients k and k'), while the marginal distribution of data samples within a client is the same, i.e., $\mathcal{P}_k(x_i) \sim \mathcal{P}_k(x_j)$ for all $i, j \in 1, 2, ..., n_k$. Figure 1 illustrates practical examples of multi-domain FL. For example, autonomous cars in different locations could capture images under different weather.

2.2 Normalization-Free Federated Learning

Figure 2a demonstrates that the BN statistics of clients with data from distinct domains are considerably dissimilar in multi-domain FL. Although various existing approaches have attempted to address this challenge by manipulating or replacing the BN layer with other normalization layers [5, 20, 30], they come with their own set of limitations, such as additional computation cost during inference. Unlike all the existing approaches, we instead propose a novel approach called **Fed**erated learning **W**ith**o**ut **n**ormalizations (FedWon), which removes all normalization layers in FL.

Compared with FedAvg [21], FedWon completely removes normalization layers in DNNs and reparameterizes convolution layers. We employ the Scaled Weight Standardization technique proposed by [4] to reparameterize the convolution layers after removing BN. The reparameterization formula can be expressed as follows:

$$\hat{W}_{i,j} = \gamma \frac{W_{i,j} - \mu_i}{\sigma_i \sqrt{N}}, \tag{1}$$



(a) FedAvg [21] **(b) FedBN [20]** **(c) FedWon (Ours)**

Figure 3: Illustration of three FL algorithms: (a) FedAvg aggregates both convolution (Conv) layers and batch normalization (BN) layers; (b) FedBN keeps BN layers in clients and only aggregates Conv layers; (c) Our proposed Federated learning Without normalizations (FedWon) removes all BN layers and reparameterizes Conv layers with scaled weight standardization (WSConv).

where $W_{i,j}$ is the weight matrix of a convolution layer, γ is a constant number, N is the fan-in of convolution layer, $\mu_i = (1/N)\sum_j W_{i,j}$ and $\sigma_i^2 = (1/N)\sum_j (W_{i,j} - \mu_i)$ are the mean and variance of the i-th row of $W_{i,j}$, respectively. This weight standardization technique is closely linked to Centered Weight Normalization [10]. By removing normalization layers, FedWon eliminates batch dependency, resolves discrepancies between training and inference, and does not require computation for normalization statistics in inference. We refer to this newly parameterized convolution as WSConv.

Figure 3 highlights the algorithmic differences between our proposed FedWon and the other two FL algorithms: FedAvg [21] and FedBN [20]. FedAvg aggregates both convolution and BN layers on the server; FedBN only aggregates the convolution layers and keeps BN layers locally in clients. Unlike these two methods, FedWon removes BN layers, replaces convolution layers with WSConv, and only aggregates these reparameterized convolution layers. Prior work theoretically shows that BN slows down and biases the FL convergence [26]. FedWon circumvents these issues by removing BN and offers unexplored benefits to multi-domain FL. These benefits include versatility for both cross-silo and cross-device FL, as well as compelling performance on batch sizes as small as 1.

3 EXPERIMENTS

3.1 Experiment Setup

Datasets. We run experiments for multi-domain FL using three datasets: Digits-Five [20], Office-Caltech-10 [7], and DomainNet [24]. Digits-Five consists of five sets of 28x28 digit images, including MNIST [17], SVHN [22], USPS [11], SynthDigits [6], MNIST-M [6]; each digit dataset represents a domain. Office-Caltech-10 consists of real-world object images from four domains: Amazon, Caltech, DSLR, and Webcam. DomainNet [24] contains 244x244 object images in six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. We follow [20] to preprocess these datasets. Besides, we evenly split samples of Digits-Five into 20 clients for cross-device FL with total 100 clients. To simulate multi-domain FL, we construct a client to contain images from a single domain.

Additionally, we simulate skewed label distribution using CIFAR-10 dataset [15]. We split training samples into 100 clients and construct i.i.d data and three levels of label skewness using Dirichlet process Dir(α) with $\alpha = \{0.1, 0.5, 1\}$.

Implementation Details. We implement FedWon using PyTorch [23] and evaluate the algorithms with three architectures in multi-domain FL: 6-layer convolution neural network (CNN) [20] for Digits-Five dataset, AlexNet [16] and ResNet-18 [8] for Office-Caltech-10 dataset, and AlexNet [16] for DomainNet dataset. Besides, we use MobileNetV2 [25] for experiments on skewed label distribution. We use cross-entropy loss and stochastic gradient optimization (SGD) as optimizer with learning rates tuned in the range of [0.001, 0.1] for all methods. By default, we conduct experiments with local epochs $E = 1$ and batch size $B = 32$.
Compared Methods. We compare FedWon with the following methods: state-of-the-art methods that use customized approaches on BN, including SiloBN [1], FedBN [20], and FixBN [30]; baseline algorithms, including FedProx [19], FedAvg [21], and Standalone training (i.e. training a model independently in each client); alternative normalization methods, including FedAvg+GN and FedAvg+LN that replace BN layers with GN [27] and LN layers [2], respectively.

3.2 Experiments on Multi-domain FL

Table 1 presents a comprehensive comparison of the aforementioned methods under cross-silo FL on Digits-Five, Office-Caltech-10, and DomainNet datasets. Our proposed FedWon outperforms the state-of-the-art methods on most of the domains. Specifically, FedProx has similar performance as FedAvg and both methods may exhibit inferior performance compared to Standalone in certain domains on DomainNet dataset. SiloBN and FixBN perform similarly to FedAvg in terms of average testing accuracy. However, they tend to underperform FedBN in multi-domain FL, where FedBN is specifically designed to excel. Interestingly, we discover that simply replacing BN with GN (FedAvg+GN) could boost the performance of FedAvg in multi-domain FL. Furthermore, our proposed FedWon surpasses both FedAvg+GN and FedBN in terms of the average testing accuracy. Although FedWon falls slightly behind FedBN by less than 1% in one domain on Digits-Five dataset, it outperforms FedBN by more than 10% on certain domains. These results demonstrate the effectiveness of FedWon under the cross-silo FL scenario. We report the mean of these methods across three runs of experiments.
Effectiveness on Small Batch Size. Table 2 compares the performance of FedWon with state-of-the-art methods using small batch sizes $B = \{1, 2\}$ on Office-Caltech-10 dataset. FedWon achieves outstanding performance, with competitive results even at a batch

Table 1: Testing accuracy (%) comparison on Digits-Five, Office-Caltech-10, and DomainNet datasets. For Digits-Five, M, S, U, Syn, and M-M are abbreviations for MNIST, SVHN, USPS, SynthDigits, and MNIST-M. For Office-Caltech-10, A, C, D, and W are abbreviations for Amazon, Caltech, DSLR, and WebCam. For DomainNet, C, I, P, O, R, and S are abbreviations for Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

Methods	Digit-Five						Office-Caltech-10					DomainNet						
	M	S	U	Syn	M-M	Avg.	A	C	D	W	Avg.	C	I	P	Q	R	S	Avg.
Standalone	94.4	67.1	95.4	80.3	77.0	83.1	54.5	40.2	81.3	89.3	66.3	42.7	24.0	34.2	**71.6**	51.2	33.5	42.9
FedAvg	96.2	71.6	96.3	86.0	82.5	86.5	61.8	44.9	77.1	81.4	66.3	48.9	26.5	37.7	44.5	46.8	35.7	40.0
FedProx	96.4	71.0	96.1	85.9	83.1	86.5	59.9	44.0	76.0	80.8	65.2	51.1	24.1	37.3	46.1	45.5	37.5	40.2
FedAvg+GN	96.4	76.9	96.6	86.6	83.7	88.0	60.8	50.8	88.5	83.6	70.9	45.4	21.1	35.4	57.2	50.7	36.5	41.1
FedAvg+LN	96.4	75.2	96.4	85.6	82.2	87.1	55.0	41.3	79.2	71.8	61.8	42.7	23.6	35.3	46.0	43.9	28.9	36.7
SiloBN	96.2	71.3	96.0	86.0	83.1	86.5	60.8	44.4	76.0	81.9	65.8	51.8	25.0	36.4	45.9	47.7	38.0	40.8
FixBN	96.3	71.3	96.1	85.8	83.0	86.5	59.2	44.0	79.2	79.6	65.5	49.2	24.5	38.2	46.3	46.2	37.4	40.3
FedBN	96.5	77.3	96.9	86.8	**84.6**	88.4	**67.2**	45.3	85.4	87.5	71.4	49.9	28.1	40.4	69.0	55.2	38.2	46.8
FedWon (Ours)	**96.8**	**77.4**	**97.0**	**87.6**	84.0	**88.5**	67.0	**50.4**	**95.3**	**90.7**	**75.6**	**57.2**	**28.1**	**43.7**	69.2	**56.5**	**51.9**	**51.1**

Table 2: Testing accuracy (%) comparison using small batch sizes $B = \{1, 2\}$ on Office-Caltech-10 dataset.

B	Methods	Amazon	Caltech	DSLR	Webcam
1	FedAvg+GN	60.4	52.0	87.5	84.8
	FedAvg+LN	55.7	43.1	84.4	88.1
	FedWon	**66.7**	**55.1**	**96.9**	**89.8**
2	FedAvg	64.1	49.3	87.5	89.8
	FedAvg+GN	63.5	52.0	81.3	84.8
	FedAvg+LN	58.3	44.9	87.5	86.4
	FixBN	66.2	50.7	87.5	88.1
	SiloBN	61.5	47.1	87.5	86.4
	FedBN	59.4	48.0	96.9	86.4
	FedWon	**66.2**	**54.7**	**93.8**	**89.8**

Table 3: Evaluation on randomly selecting $C = \{10\%, 40\%\}$ out of total 100 clients to train each round with batch size $B = 4$. M, S, U, Syn, and M-M are abbreviations of five domains.

C	Method	M	S	U	Syn	M-M	Avg.
10%	FedAvg	98.2	81.0	97.2	91.6	89.3	91.5
	FedWon	**98.6**	**85.4**	**98.3**	**93.6**	**90.5**	**93.3**
40%	FedAvg	98.1	80.5	97.0	91.4	89.4	91.3
	FedWon	**98.8**	**86.4**	**98.4**	**94.2**	**91.0**	**93.7**

size of 1. Although FedAvg+GN also achieves comparable results on batch size $B = 1$, it requires additional computational cost during inference to calculate the running mean and variance. The capability of our method to perform well with small batch sizes is particularly important for cross-device FL, as some devices may only be capable of training with small batch sizes under constrained resources. We tune the best learning rates for methods in these experiments.

Impact of Selection a Subset of Clients. We assess the impact of randomly selecting a fraction of clients to participate in each training round, which is common in cross-device FL. We conduct experiments with fraction $C = \{10\%, 40\%\}$ out of 100 clients on Digits-Five dataset, i.e., $K = \{10, 40\}$ clients are selected to participate in training in each round. Table 3 shows that FedWon outperforms FedAvg under all client fractions. FedBN is not compared as it is not applicable in cross-device FL. We also evaluate small batch sizes in cross-device FL, with $K = 10$ clients selected per round. Figure 4 illustrates that our proposed FedWon consistently surpasses FedAvg with batch sizes $B = \{1, 2\}$ and it achieves consistently comparable results to training with larger batch sizes.

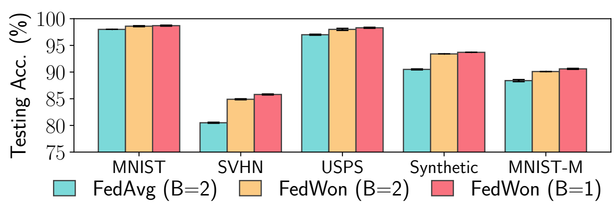


Figure 4: Performance comparison of FedWon and FedAvg using small batch sizes B = {1, 2} on Digits-Five dataset, where $C = 10\%$ out of 100 clients are randomly selected each round.

Evaluation on Alternative Backbones. In addition to evaluating the effectiveness of FedWon using AlexNet [16], Table 4 compares testing accuracies with ResNet18 [8] on the Office-Caltech-10 dataset. Our proposed FedWon generally has better performance than the existing methods even using ResNet-18 as the backbone.

3.3 Experiments on Skewed Label Distribution

We run experiments on skewed label distribution with a fraction $C = 10\%$ randomly selected clients (i.e., $K = 10$) out of total 100 clients in each round. Table 5 compares our proposed FedWon with FedAvg, FedAvg+GN, FedAvg+LN, and FixBN. FedWon achieves similar performance as FedAvg and FixBN on the i.i.d setting, but outperforms all methods on different degrees of label skewness. We do not compare with FedBN and SiloBN as they are not suitable for cross-device FL. All experiments are run with local epoch $E = 5$ for

Table 4: Testing accuracy (%) comparison using ResNet-18 on Office-Caltech-10 Dataset.

Methods	Amazon	Caltech	DSLR	Webcam	Average
FedAvg	45.3	36.4	68.8	76.3	56.7
FedAvg+GN	44.3	31.1	71.9	74.6	55.5
FedAvg+LN	34.4	26.2	59.4	44.1	41.0
FixBN	34.9	33.8	62.5	78.0	52.3
SiloBN	40.6	29.3	59.4	81.4	52.7
FedBN	57.3	37.3	90.6	**89.8**	68.8
FedWon	**63.0**	**46.7**	**90.6**	86.4	**71.7**

Table 5: Testing accuracy comparison on different levels of label skewness using MobileNetV2 as backbone on CIFAR-10 dataset, where Dir (0.1) is the most skewed setting.

Methods	i.i.d	Dir (1)	Dir (0.5)	Dir (0.1)
FedAvg	75.0	64.5	61.1	36.0
FedAvg+GN	65.3	58.8	51.8	21.5
FedAvg+LN	69.2	61.8	57.9	23.3
FixBN	75.4	64.1	61.2	34.7
FedWon (Ours)	**75.7**	**72.8**	**70.7**	**41.9**

300 rounds. We use SGD as the optimizer and tune the learning in the range of [0.001, 0.1] for different algorithms. These experiments indicate the possibility of employing our proposed FedWon to solve the skewed label distribution problem.

4 CONCLUSION

In conclusion, we propose FedWon, a new method for multi-domain FL by removing all normalizations and reparameterizing convolution layers with weight scaled convolution. Extensive experiments across four datasets and models demonstrate that this simple yet effective method outperforms state-of-the-art methods in a wide range of settings. Notably, FedWon is versatile for both cross-silo and cross-device FL. Its ability to train on small batch sizes is particularly useful for resource-constrained devices.

REFERENCES

[1] Mathieu Andreux, Jean Ogier du Terrail, Constance Beguier, and Eric W. Tramel. 2020. Siloed Federated Learning for Multi-centric Histopathology Datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer International Publishing, 129–139.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Tobias Bernecker, Annette Peters, Christopher L Schlett, Fabian Bamberg, Fabian Theis, Daniel Rueckert, Jakob Weiß, and Shadi Albarqouni. 2022. FedNorm: Modality-Based Normalization in Federated Learning for Multi-Modal Liver Segmentation. *arXiv preprint arXiv:2205.11096* (2022).

[4] Andrew Brock, Soham De, and Samuel L Smith. 2021. Characterizing signal propagation to close the performance gap in unnormalized resnets. *International Conference on Learning Representations* (2021).

[5] Zhixu Du, Jingwei Sun, Ang Li, Pin-Yu Chen, Jianyi Zhang, Hai" Helen" Li, and Yiran Chen. 2022. Rethinking normalization methods in federated learning. In *Proceedings of the 3rd International Workshop on Distributed Machine Learning*. 16–22.

[6] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[7] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2066–2073.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. 2020. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*. PMLR, 4387–4398.

[10] Lei Huang, Xianglong Liu, Yang Liu, Bo Lang, and Dacheng Tao. 2017. Centered weight normalization in accelerating training of deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2803–2811.

[11] Jonathan J. Hull. 1994. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence* 16, 5 (1994), 550–554.

[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, 448–456.

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*. PMLR, 5132–5143.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37 (2020), 50–60.

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

[20] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623* (2021).

[21] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[24] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1406–1415.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[26] Yanmeng Wang, Qingjiang Shi, and Tsung-Hui Chang. 2023. Why Batch Normalization Damage Federated Learning on Non-IID Data? *arXiv preprint arXiv:2301.02982* (2023).

[27] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[28] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2636–2645.

[29] Hongyi Zhang, Jan Bosch, and Helena Holmström Olsson. 2021. End-to-end federated learning for autonomous driving vehicles. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[30] Jike Zhong, Hong-You Chen, and Wei-Lun Chao. 2023. Making Batch Normalization Great in Federated Deep Learning. *arXiv preprint arXiv:2303.06530* (2023).

[31] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. 2020. Performance Optimization of Federated Person Re-identification via Benchmark Analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*. 955–963.