Is Batch Normalization Necessary for Multi-domain Federated Learning?

Weiming Zhuang, **Lingjuan Lyu**
Sony Research
{weiming.zhuang, lingjuan.lv}@sony.com

Abstract

Federated learning (FL) enhances data privacy with collaborative in-situ training on decentralized clients. However, FL suffers from non-independent and identically distributed (non-i.i.d) data, which can degrade performance and hurt training convergence. Most existing works address the skewed label distribution problem, whereas we focus on an important but less considered problem, multi-domain FL, where data in clients are from different domains that have different feature distribution instead of label distribution. Multi-domain FL is practical in many applications such as autonomous driving where cars in different locations capture images in varying weather conditions. In this work, we propose **Fed**erated learning **W**ith**o**ut **N**ormalizations (FedWon) for multi-domain FL. FedWon is motivated by the insight that batch normalization (BN) struggles to model statistics of multiple domains. It removes all BN layers in FL and reparameterizes the convolution layers with scaled weight standardization. Extensive experiments on four datasets and four models demonstrate that FedWon outperforms FedAvg and the state-of-the-art method (FedBN) on all settings, achieving more than 10% improvement on certain domains. Moreover, FedWon is versatile for both cross-silo and cross-device FL, is applicable to address the skewed label distribution problem, and performs well even on a batch size of 1, making it useful for resource-constrained devices.

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, including healthcare [Li *et al.*, 2019; Bernecker *et al.*, 2022], mobile devices [Hard *et al.*, 2018; Paulik *et al.*, 2021], and autonomous vehicles [Zhang *et al.*, 2021]. However, FL commonly suffers from statistical heterogeneity, where the data distributions across clients are non-independent and identically distributed (non-i.i.d) [Li *et al.*, 2020a]. 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 [Hsieh *et al.*, 2020] or even divergence in training [Zhuang *et al.*, 2020].

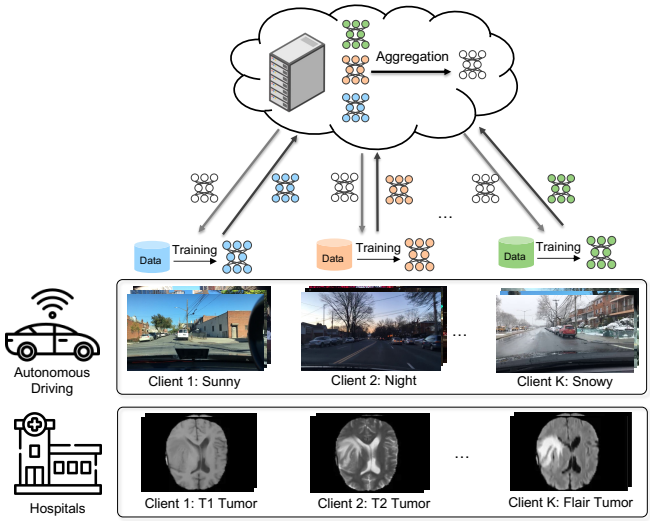
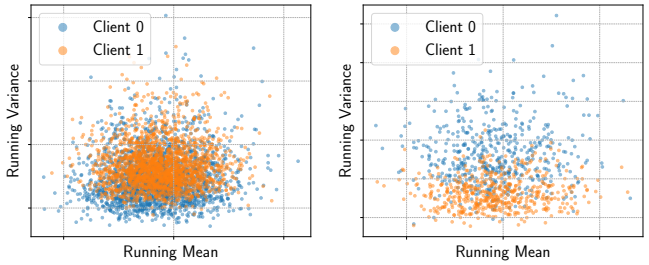


Figure 1: We consider multi-domain federated learning (FL), where each client contains data of one domain. This setting is highly practical and applicable in real-world scenarios. For example, autonomous driving cars in distinct geographical locations capture images in varying weather conditions, and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

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 [Li *et al.*, 2020b; Hsieh *et al.*, 2020; Chen *et al.*, 2022]. However, multi-domain FL, where data in clients are from different domains, has received little attention, despite its practicality in many real-world scenarios. Figure 1 depicts two practical examples of multi-domain FL. For example, multiple autonomous driving cars may collaborate on model training, but their data could originate from different weather conditions or times of day, leading to domain discrepancies in images collected in different clients [Yu *et al.*, 2020]. Similarly, multiple healthcare institutions collaborating on medical imaging analysis may face significant domain gaps due to variations in imaging machines and protocols [Bernecker *et al.*, 2022]. Developing



(a) Statistics of 4-th BN Layer. (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 5-layer CNN, respectively. It highlights different feature statistics of BN layers trained on different domains.

effective solutions for multi-domain FL is a critical research problem with broad implications.

However, all the existing solutions are not able to adequately address the problem of multi-domain FL. FedBN [Li *et al.*, 2021] attempts to solve this problem by keeping batch normalization (BN) [Ioffe and Szegedy, 2015] parameters and statistics locally in client, but it is only suitable for cross-silo FL [Kairouz *et al.*, 2021], where clients are organizations like healthcare institutions, because it requires clients to be stateful [Karimireddy *et al.*, 2020] (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 client participate in training. Besides, BN relies on the assumption that training data are from the same distribution, ensuring the mean and variance of each mini-batch is representative to the entire data distribution [Ioffe and Szegedy, 2015]. Figure 2 shows that the running mean and variance of BNs in two FL clients with multiple domain data can differ significantly.

This paper explores a fundamentally different approach to address multi-domain FL. Given the bottlenecks of BN bottleneck in capturing multi-domain data, we further ask the question: is it possible to learn a general global model without BN layers for multi-domain FL? Recent works have proposed normalization-free ResNets [Brock *et al.*, 2021a] that achieve comparable performance to standard ResNets [He *et al.*, 2016]. We build upon this methodology and investigate its unexplored potential in the context 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 [McMahan *et al.*, 2017] protocols for server aggregation and client training. Unlike existing methods, FedWon removes normalization layers (e.g., BN layers), and reparameterizes the convolution layer with Scaled Weight Standardization [Brock *et al.*, 2021a]. We conducted extensive experiments on four datasets (Digits-Five, Office-Caltech-10 [Gong *et al.*, 2012], DomainNet [Peng *et al.*, 2019], and CIFAR-10 [Krizhevsky *et al.*, 2009]) using four models (5-layer CNN [Li *et al.*, 2021], AlexNet [Krizhevsky *et al.*, 2017], ResNet-18 [He *et al.*, 2016], and MobileNetV2 [Sandler *et al.*, 2018]), where CIFAR-10 dataset is for extended evaluation on skewed label distribution. These experimental results indicate that FedWon outperforms state-of-the-art methods on all datasets and models. The *general global model* trained by FedWon can achieve more than 10% improvement on certain domains compared to the *personalized models* from FedBN [Li *et al.*, 2021]. Moreover, our empirical evaluation demonstrated three key highlights of FedWon: 1) FedWon is a use-case agnostic method that supports both cross-silo and cross-device FL; 2) FedWon achieves competitive performance on small batch size (even with batch size 1), which is particularly useful for edge devices with memory constraints.; 3) FedWon can also be applied to address the imbalanced label distribution problem.

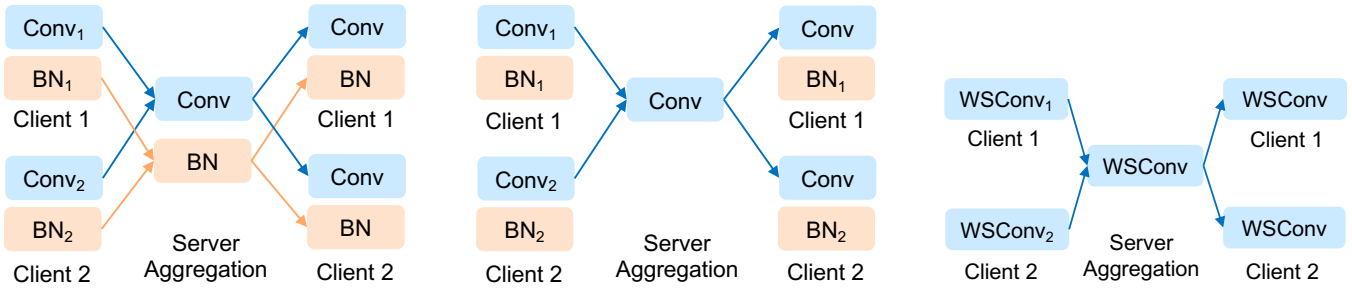
In summary, our contributions are as follows:

- We introduce FedWon, a simple yet effective method for multi-domain FL. By removing all normalization layers and using scaled weight standardization, FedWon is able to learn a general global model from clients with significant domain discrepancies.
- To the best of our knowledge, FedWon is the first method that enables FL without any normalization for both cross-silo and cross-device FL. 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 Related Work

In this section, we provide a literature review of federated learning, normalizations in centralized training and federated learning, and normalization-free networks.

Federated Learning. Federated learning (FL) trains machine learning models collaboratively from decentralized clients, coordinated by a central server [Kairouz *et al.*, 2021]. It enhances data privacy protection by keeping the raw data locally on clients. FedAvg [McMahan *et al.*, 2017] is the most popular FL algorithm, but non-i.i.d data across clients can lead to performance degradation and difficulties in training convergence [Hsieh *et al.*, 2020]. Skewed label distribution, where clients have different label distributions, is a commonly discussed non-i.i.d. problem, and various methods have been proposed to address it [Li *et al.*, 2020b; Chen *et al.*, 2022]. In contrast, multi-domain FL, where the data domains differ across clients, has received less attention. FedBN [Li *et al.*, 2021] addresses this issue by locally keeping the batch normalization layers in clients and aggregating only the other model parameters. FedNorm [Bernecker *et al.*, 2022] extends this idea to medical imaging segmentation by maintaining a BN locally for each data modality. These approaches work well in the cross-silo FL scenario, where clients are stable and allow for statefulness. However, they are not suitable for the cross-device FL scenario, where clients are stateless, and only a fraction of clients participate in each round of training.



(a) FedAvg [McMahan *et al.*, 2017] (b) FedBN [Li *et al.*, 2021] (c) FedWon (Ours)

Figure 3: Illustration of three FL algorithms: (a) FedAvg aggregates both convolution (Conv) layers and batch normalization (BN) layers in server; (b) FedBN keeps BN layers locally in clients and only aggregate Conv layers; (c) Our proposed **Fed**erated learning **W**ith**o**ut **n**ormalizations (FedWon) removes all BN layers and reparameterizes Conv layers with scaled weight standardization (WSConv).

Normalization Layers in Centralized Training. Batch Normalization (BN) has become a fundamental component in modern deep neural networks (DNN) [Ioffe and Szegedy, 2015]. It offers several benefits, including reducing internal covariate shift, stabilizing training, and accelerating convergence [Ioffe and Szegedy, 2015]. Moreover, it is more robust to hyperparameters [Bjorck *et al.*, 2018] and has smoother optimization landscapes [Santurkar *et al.*, 2018]. However, BN also has limitations in various scenarios; it struggles to model statistics of training data from multiple domains, and it may not be suitable for small batch sizes. Researchers have proposed alternative normalizations such as Group Norm [Wu and He, 2018] and Layer Norm [Ba *et al.*, 2016]. Although these methods remove some of the constraints of BN, they come with their own set of limitations. For example, they require additional computation during inference, making them less practical for edge deployment.

Normalization Layers in Federated Learning. Recent studies have shown that BN may not work well in FL under non-i.i.d data [Hsieh *et al.*, 2020], due to external covariate shift [Du *et al.*, 2022] and mismatch between local and global statistics [Wang *et al.*, 2023]. Instead, researchers have adopted alternative normalizations such as GN [Hsieh *et al.*, 2020; Casella *et al.*, 2023] or LN [Du *et al.*, 2022; Casella *et al.*, 2023] that perform better in FL under non-i.i.d data. However, the recent study by Zhong et al. [Zhong *et al.*, 2023] shows that BN and GN have no consistent winner, and they propose FixBN [Zhong *et al.*, 2023] as a solution that only trains BN statistics in the first stage of training and freezes them in the remaining training. Additionally, several works have explored personalized FL that trains personalized model for each client by introducing special operations on BN [Andreux *et al.*, 2020; Li *et al.*, 2021; Bernecker *et al.*, 2022; Lu *et al.*, 2022]. SiloBN [Andreux *et al.*, 2020] keeps BN statistics locally in clients; FedBN [Li *et al.*, 2021] keeps BN parameters and statistics locally.

Normalization-free Networks. Several attempts have been made to remove normalization from DNNs in centralized training using weight initialization methods [Zhang *et al.*, 2019; De and Smith, 2020]. Recently, Brock et al. [Brock *et al.*, 2021a] proposed a normalization-free network via analyzing the signal propagation through the forward pass of the network. Normalization-free network stabilizes training with scaled weight standardization that reparameterizes the convolution layer to prevent mean shift in the hidden activations [Brock *et al.*, 2021a]. This approach achieves competitive performance compared to networks with BN on ResNet [He *et al.*, 2016] and EfficientNet [Tan and Le, 2019]. Building on this work, Brock et al. further introduced an adaptive gradient clipping (AGC) method that enables training normalization-free networks with large batch sizes, further improving overall performance [Brock *et al.*, 2021b].

3 Federated Learning Without Normalization

In this section, we present the problem setup of multi-domain FL, discuss FL algorithms with BN, and propose FL without normalization to address the problem of multi-domain FL.

3.1 Problem Setup

The standard federated learning aims to train a model with parameters θ collaboratively from total $N \in \mathbb{N}$ decentralized clients. The goal is to optimize the following problem:

$$\min_{\theta \in \mathbb{R}^d} f(\theta) := \sum_{k=1}^{K} p_k f_k(\theta) := \sum_{k=1}^{K} p_k \mathbb{E}_{\xi_k \sim \mathcal{D}_k}[f_k(\theta; \xi_k)], \tag{1}$$

where $K \in \mathbb{N}$ is the number of participated clients ($K \leq N$), $f_k(\theta)$ is the loss function of client k, p_k is the weight for model aggregation in the server, and ξ_k is the data sampled from distribution $\mathcal{D}_k$ of client k. FedAvg [McMahan *et al.*, 2017] sets p_k to be proportional to the dataset size of client k. Each client trains for $E \in \mathbb{N}$ local epochs before communicating with the server.

Assume there are N clients in FL and each client k contains $n_k \in \mathbb{N}$ data samples $\{(x_i^k, y_i^k)\}_{i=1}^{n_k}$ data samples. Skewed label distribution refers to the scenario where data in different clients have different label distributions, i.e. the marginal distributions $\mathcal{P}_k(y)$ may differ across clients. In contrast, this work focuses on multi-domain FL, where clients possess data from various domains, and data samples within each client belong to the same domain [Kairouz *et al.*, 2021; Li *et al.*, 2021]. Specifically, the marginal distribution $\mathcal{P}_k(x)$ may vary across clients, while the marginal distribution of data samples within a client is the same, i.e., $\mathcal{P}_k(x_i) =$

$\mathcal{P}_k(x_j)$ for all $i, j \in 1, 2, ..., n_k$. Figure 1 illustrates two practical examples of multi-domain FL. For example, multiple autonomous driving cars in different locations could capture images of different weather conditions.

3.2 Federated Learning with Batch Normalization

Batch normalization (BN), commonly used as a normalization layer, has been a fundamental component of deep neural networks (DNN). The BN operation is defined as follows:

$$BN(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{2}$$

where mean μ and variance σ are computed from a mini-batch of data, and γ and β are two learnable parameters. The term ϵ is a small positive value that is added to the denominator for numerical stability.

BN has shown to be effective for stabilizing training and accelerating convergence [Ioffe and Szegedy, 2015]. However, its effectiveness is based on the assumption that the training data is from the same domain, such that the mean μ and variance σ computed from a mini-batch are representative of the training data. In centralized training, BN has been found to struggle with modeling the statistics from multiple domains, leading to the development of domain-specific BN techniques [Li *et al.*, 2016; Chang *et al.*, 2019].

Similarly, in multi-domain FL, DNN with BN can encounter difficulties in capturing the statistics of multiple domains while training a single global model. To address this issue, methods such as FedBN [Li *et al.*, 2021] have been proposed, which instead learns personalized models with personalized BN for each client. As a result, FedBN requires stateful clients and is only suitable for cross-silo FL. In contrast, we aim to learn a general global model in multi-domain FL that is suitable for both cross-silo and cross-device FL.

3.3 Normalization-Free Federated Learning

Figure 2 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 [Li *et al.*, 2021; Du *et al.*, 2022; Zhong *et al.*, 2023], they come with their own set of limitations, such as additional computation cost during inference. We propose an unconventional approach, **Fed**erated learning **W**ith**o**ut **n**ormalizations (FedWon), which removes all normalization layers in FL.

Particularly, FedWon adheres the server aggregation and client training protocol of FedAvg [McMahan *et al.*, 2017]. However, it completely removes the normalization layers in DNN and reparameterizes the convolutions layer. We employ the Scaled Weight Standardization technique proposed by [Brock *et al.*, 2021a] to reparameterize the convolution layers after removing BN. The reparameterization formula can be expressed as follows:

$$\hat{W}_{i,j} = \gamma \frac{W_{i,j} - \mu_i}{\sigma_i \sqrt{N}}, \tag{3}$$

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 [Huang *et al.*, 2017]. By eliminating the need for normalization layers, the weight standardization technique eliminates batch dependency, resolves discrepancies between training and inference, and is easy to train and free at 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 [McMahan *et al.*, 2017] and FedBN [Li *et al.*, 2021]. FedAvg aggregates both convolution and BN on the server; FedBN only aggregates the convolution layers and keeps BN layers locally in clients. Unlike these two methods, FedWon removes BN, replaces convolution layers with WSConv, and only aggregates these reparameterized convolution layers. It circumvents the limitations of BN and offers unexplored benefits to multi-domain FL. These benefits include suitability for both cross-silo and cross-device FL, as well as compelling performance on small batch sizes, including batch size of 1, which are further demonstrated in Section 4.2.

4 Experiments on Multi-domain FL

In this section, we start by introducing experimental setup for multi-domain FL. We then present empirical results on both cross-silo and cross-device FL, followed by ablation studies.

4.1 Experiment Setup

Datasets. We conduct experiments for multi-domain FL using three datasets: Digits-Five [Li *et al.*, 2021], Office-Caltech-10 [Gong *et al.*, 2012], and DomainNet [Peng *et al.*, 2019]. Digits-Five consists of five datasets of 28x28 digit images, including USPS [Hull, 1994], MNIST [LeCun *et al.*, 1998], SVHN [Netzer *et al.*, 2011], MNIST-M [Ganin and Lempitsky, 2015], and Synthetic Digits [Ganin and Lempitsky, 2015]; each digit dataset represents a domain. Office-Caltech-10 consists of real-world object images from four domains: three domains (WebCam, DSLR, and Amazon) from Office-31 dataset [Saenko *et al.*, 2010] and one domain (Caltech) from Caltech-256 dataset [Griffin *et al.*, 2007]. DomainNet [Peng *et al.*, 2019] contains large-sized 244x244 common object images in six different domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. To mimic the realistic scenarios where clients may not collect a large volume of data, we use a subset of standard digits dataset as adopted in [Li *et al.*, 2021], with 7,438 training samples for each digit dataset instead of tens of thousands. These samples of each domain are further split into 10 clients, simulating 50 clients using Digits-Five datasets. Similarly, we tailor DomainNet dataset to include only 10 classes of 2,000-5,000 images. We simulate multi-domain FL by constructing a client to contain images of one domain using respective datasets.

Implementation Details. We implement the proposed FedWon using PyTorch [Paszke *et al.*, 2017] and run experiments on a cluster of eight NVIDIA T4 GPUs. We evaluate the algorithms using three models: 5-layer convolution neural network (CNN) [Li *et al.*, 2021] for Digits-Five dataset, AlexNet

Table 1: Performance comparison on test accuracy (%) of different methods on three datasets. Our proposed FedWon outperforms the existing methods on most of the domains. In terms of average test accuracy, FedWon achieves the best performance in all datasets. These experiments are the average of five runs; we report these results as mean ± standard deviation.

	Domains	Standalone	FedAvg	FedProx	FedAvg+GN	FedAvg+LN	SiloBN	FixBN	FedBN	Ours
Digit-Five	MNIST	94.4±0.2	96.2±0.2	96.4±0.0	96.4±0.1	96.4±0.1	96.2±0.0	96.3±0.1	96.5±0.1	**96.8±0.2**
	SVHN	67.1±0.7	71.6±0.5	71.0±0.8	76.9±0.1	75.2±0.4	71.3±1.0	71.3±0.9	77.3±0.4	**77.4±0.1**
	USPS	95.4±0.1	96.3±0.3	96.1±0.1	96.6±0.2	96.4±0.4	96.0±0.2	96.1±0.2	96.9±0.2	**97.0±0.1**
	Synth	80.3±0.8	86.0±0.3	85.9±0.2	86.6±0.1	85.6±0.3	86.0±0.3	85.8±0.1	86.8±0.3	**87.6±0.2**
	MNIST-M	77.0±0.9	82.5±0.1	83.1±0.2	83.7±0.5	82.2±0.3	83.1±0.4	83.0±0.8	**84.6±0.2**	84.0±0.2
	Average	83.1±0.4	86.5±0.1	86.5±0.1	88.0±0.1	87.1±0.0	86.5±0.3	86.5±0.0	88.4±0.1	**88.5±0.1**
Caltech-10	Amazon	54.5±1.8	61.8±1.2	59.9±0.5	60.8±1.8	55.0±0.3	60.8±1.3	59.2±1.8	**67.2±0.9**	67.0±0.7
	Caltech	40.2±0.7	44.9±1.2	44.0±1.9	50.8±3.3	41.3±1.2	44.4±1.2	44.0±0.8	45.3±1.3	**50.4±2.8**
	DSLR	81.3±0.0	77.1±1.8	76.0±1.8	88.5±1.8	79.2±1.8	76.0±1.8	79.2±1.8	85.4±1.8	**95.3±2.2**
	Webcam	89.3±1.0	81.4±1.7	80.8±2.6	83.6±5.2	71.8±2.0	81.9±2.0	79.6±2.9	87.5±1.0	**90.7±1.2**
	Average	66.3±0.4	66.3±0.7	65.2±1.0	70.9±2.5	61.8±0.7	65.8±0.2	65.5±0.8	71.4±1.0	**75.6±1.4**
DomainNet	Clipart	42.7±2.7	48.9±2.0	51.1±0.8	45.4±0.5	42.7±0.7	51.8±1.0	49.2±1.8	49.9±0.5	**57.2±0.5**
	Infograph	24.0±1.6	26.5±2.5	24.1±1.6	21.1±1.1	23.6±1.2	25.0±2.1	24.5±0.9	28.1±0.8	**28.1±0.2**
	Painting	34.2±1.6	37.7±3.3	37.3±2.0	35.4±2.0	35.3±0.6	36.4±1.9	38.2±0.7	40.4±0.7	**43.7±1.2**
	Quickdraw	**71.6±0.9**	44.5±3.4	46.1±3.8	57.2±1.0	46.0±1.2	45.9±2.8	46.3±3.9	69.0±0.8	69.2±0.2
	Real	51.2±1.0	46.8±2.3	45.5±0.6	50.7±0.3	43.9±0.7	47.7±0.9	46.2±2.8	55.2±2.6	**56.5±0.4**
	Sketch	33.5±1.1	35.7±0.9	37.5±2.3	36.5±1.8	28.9±1.3	38.0±1.9	37.4±2.0	38.2±6.7	**51.9±1.9**
	Average	42.9±0.5	40.0±1.5	40.2±0.5	41.1±0.0	36.7±0.3	40.8±0.4	40.3±0.3	46.8±1.5	**51.1±0.2**

[Krizhevsky *et al.*, 2017] and ResNet18 [He *et al.*, 2016] for Office-Caltech-10 dataset, and AlexNet [Krizhevsky *et al.*, 2017] for DomainNet dataset. We use cross-entropy loss and stochastic gradient optimization (SGD) as optimizer with learning rates tuned over the range of [0.001, 0.1] for all methods. Based on SGD, we adopt adaptive gradient clipping (AGC) that is specially designed for normalization-free networks [Brock *et al.*, 2021b]. AGC comes with a clipping threshold hyperparameter λ, which is set to $\lambda = 0.1$ if not specified. By default, we conduct experiments with local epochs $E = 1$ and batch size $B = 32$ for all datasets.

4.2 Performance Evaluation

We compare the performance of our proposed FedWon with the following methods: state-of-the-art methods that employ customized approaches on BN, including SiloBN [Andreux *et al.*, 2020], FedBN [Li *et al.*, 2021], and FixBN [Zhong *et al.*, 2023]; baseline algorithms, including FedProx [Li *et al.*, 2020b], FedAvg [McMahan *et al.*, 2017], 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 [Wu and He, 2018] and LN layers [Ba *et al.*, 2016], respectively.

Table 1 presents a comprehensive comparison of the 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 across all datasets. Specifically, FedProx, which adds a proximal term based on FedAvg, performs similarly to FedAvg. These two methods are better than Standalone in Digits-Five dataset, but they may exhibit inferior performance compared to Standalone in certain domains on the other two more challenging datasets. 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. Surprisingly, we discover that simply replacing BN with GN (FedAvg+GN) can 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 on all datasets. Although FedWon falls slightly behind FedBN by less than 1% on two domains across these datasets, it outperforms FedBN by more than 17% on certain domains. These results demonstrate the effectiveness of FedWon under the cross-silo FL scenario. We report the mean (standard deviation) of these methods across three runs of experiments.

Analysis on Different Degrees of Domain Heterogeneity. We evaluate the performance of the proposed FedWon under different degrees of domain heterogeneity. To simulate varying degrees of domain heterogeneity, we follow the approach taken by FedBN [Li *et al.*, 2021] and create different numbers of clients with the same domain on the Digits-Five dataset. We start with 5 clients, each containing data from one domain, and then add 5 clients at a time, with each new client containing one of the Digits-Five dataset, respectively. We evaluate the performance of the algorithms for different numbers of clients from $N = \{5, 10, 15, ..., 50\}$. More clients represent less heterogeneity as more clients have overlapping domains of data. Figure **??** (left) compares the performance of FedWon and FedBN under these settings. The results show that the performances of both FedWon and FedBN increase as the degree of heterogeneity decreases. FedBN outperforms FedAvg in all the settings as evidenced in [Li *et al.*, 2021]. However, FedWon achieves even better performance than FedBN on all domains and all levels of heterogeneity.

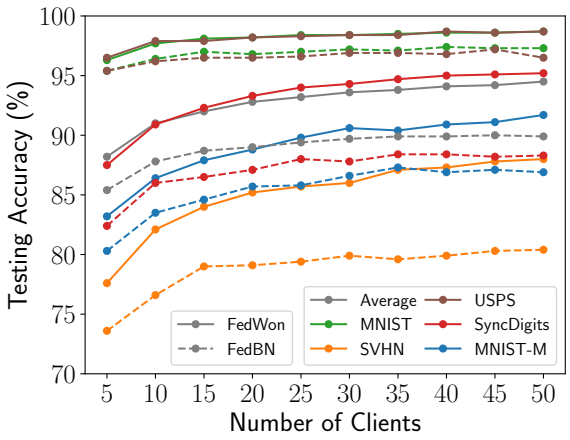


Figure 4: Testing accuracy comparison on different degrees of domain heterogeneity.

Effectiveness on Small Batch Size. Table 2 compares the performance of our proposed FedWon with state-of-the-art methods using small batch sizes $B = \{1, 2, 4\}$ on Office-Caltech-10 dataset. FedWon achieves outstanding performance, with competitive results even at a batch size of 1. While FedAvg+GN and FedAvg+LN also achieve comparable results on batch size $B = 1$, they require additional computational cost during inference to calculate the running mean and variance, whereas our method does not have such constraints and achieves even better performance. The capability of our method to perform well under small batch sizes is particularly important for cross-device FL, as some edge devices may only be capable of training with small batch sizes under constrained resources. We have fine-tuned the learning rates for all methods and report the best ones.

Impact of Selection a Subset of Clients. We assess the impact of randomly selecting a fraction of clients to participate in training in each round, which is common in cross-device FL where not all clients join in training. We conduct experiments with fraction $C = \{0.1, 0.2, 0.4\}$ out of 100 clients on Digits-Five dataset, i.e., $K = \{10, 20, 40\}$ clients are selected to participate in training in each round. Table 3 shows that the performance of our FedWon is better than 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 **??** (right) shows that the performance of FedAvg degrades with batch size $B = 2$, while our proposed FedWon with batch sizes $B = \{1, 2\}$ achieves consistently comparable results to running with larger batch sizes.

Evaluation on An Alternative Backbone. In addition to evaluating the effectiveness of FedWon using AlexNet [Krizhevsky *et al.*, 2017] on the Office-Caltech-10 dataset, Table 4 also compares testing accuracy on a common backbone, ResNet18 [He *et al.*, 2016]. Interestingly, replacing BN with GN or LN is not as effective on ResNet-18 as on AlexNet. FedAvg+GN and FedAvg+LN only achieve similar or even worse performance than FedAvg. FedBN [Li *et al.*, 2021], instead, achieves better performance than the other existing methods. Nevertheless, our proposed FedWon consistently outperforms the state-of-the-art methods even with ResNet18 as the backbone.

Table 2: Performance comparison using small batch sizes $B = \{1, 2, 4\}$ on Office-Caltech-10 dataset. The abbreviations A, C, D, and W respectively represent 4 domains: Amazon, Caltech, DSLR, and WebCam. Our proposed FedWon achieves outstanding performance compared to existing methods.

B	Methods	A	C	D	W	Avg.
1	FedAvg+GN	60.4	52.0	87.5	84.8	71.2
	FedAvg+LN	55.7	43.1	84.4	88.1	67.8
	FedWon	**66.7**	**55.1**	**96.9**	**89.8**	**77.1**
2	FedAvg	64.1	49.3	87.5	89.8	72.7
	FedAvg+GN	63.5	52.0	81.3	84.8	70.4
	FedAvg+LN	58.3	44.9	87.5	86.4	69.3
	FixBN	66.2	50.7	87.5	88.1	73.1
	SiloBN	61.5	47.1	87.5	86.4	70.6
	FedBN	59.4	48.0	96.9	86.4	72.7
	FedWon	**66.2**	**54.7**	**93.8**	**89.8**	**76.1**
4	FedAvg	65.6	46.7	78.1	88.1	69.6
	FedAvg+GN	60.9	52.0	84.4	89.8	71.8
	FedAvg+LN	54.2	44.9	78.1	72.9	62.5
	FixBN	66.2	50.2	78.1	91.5	71.5
	SiloBN	63.5	48.9	78.1	88.1	69.7
	FedBN	67.2	50.7	90.6	91.5	75.0
	FedWon	**68.8**	**54.2**	**96.9**	**91.5**	**77.8**

Table 3: Testing accuracy comparison on randomly selecting a fraction $C = \{0.1, 0.2, 0.4\}$ out of total 50 clients to participate in training each round. FedWon outperforms FedAvg in all domains of Digits-Five dataset.

C	Method	M	S	U	Syn	M-M
0.1	FedAvg	98.2	81.0	97.2	91.6	89.3
	FedWon (Ours)	**98.6**	**85.4**	**98.3**	**93.6**	**90.5**
0.2	FedAvg	97.9	80.2	97.0	91.2	89.3
	FedWon (Ours)	**98.7**	**86.0**	**98.2**	**94.1**	**90.8**
0.4	FedAvg	98.1	80.5	97.0	91.4	89.4
	FedWon (Ours)	**98.8**	**86.4**	**98.4**	**94.2**	**91.0**

4.3 Ablation Studies

We conduct ablation studies to further analyze the impact of reparameterizing convolution layers with scaled weight standardization (WSConv). We evaluated the impact of WSConv on performance at two batch sizes $B = 32$ and $B = 2$ on Office-Caltech-10 dataset. Table 5 demonstrates that replacing convolution layers with WSConv significantly enhances performance when the network removes all normalization layers, for both batch sizes. We only employ AGC for $B = 32$ as it is effective for larger batch size [Brock *et al.*, 2021b]. These experiments use a learning rate of $\eta = 0.08$ for $B = 32$ and $\eta = 0.01$ for $B = 2$.

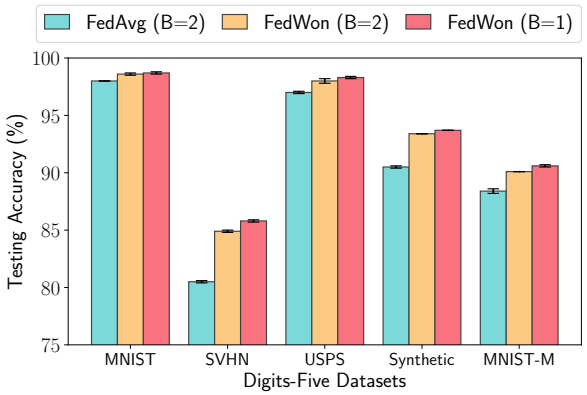


Figure 5: Performance comparison of FedWon and FedAvg using small batch sizes B ={1, 2} on Digits-Five dataset, where a fraction $C = 0.1$ out of 50 clients are randomly selected to train each round.

Table 4: Testing accuracy comparison using ResNet-18 as the backbone on Office-Caltech-10 Dataset, where A, C, D, and W are the abbreviations of four domains.

Methods	A	C	D	W	Avg
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 (Ours)	**63.0**	**46.7**	**90.6**	86.4	**71.7**

Table 5: Ablation studies on the impact of WSConv on Office-Caltech-10 dataset. After removing all BN layers, employing WSConv significantly boosts performance.

Batch Size	WSConv	A	C	D	W	Avg
B = 32	✓	**63.7**	**51.0**	**96.3**	**91.2**	**75.6**
		46.4	37.3	68.8	71.2	55.9
B = 2	✓	**67.2**	**55.6**	**96.9**	**93.2**	**78.2**
		54.7	44.0	84.4	78.0	65.3

5 Experiments on Skewed Label Distribution

This section extends evaluation from multi-domain FL to skewed label distribution. We demonstrate that our proposed FedWon, though not specially designed for skewed label distribution, also can address this problem.

Dataset and Implementation. We simulate skewed label distribution using CIFAR-10 dataset [Krizhevsky *et al.*, 2009], which comprises 50,000 training samples and 10,000 testing samples. We split the training samples into 100 clients and construct i.i.d data and three different levels of label skewness. In particular, we use Dirichlet process Dir(α) to sample non-i.i.d label skewness, where different values $\alpha = \{0.1, 0.5, 1\}$ present varying label skewness (Dir(0.1) is the most heterogeneous setting). We conduct the experiments using MobileNetV2 [Sandler *et al.*, 2018] and run experiments with a fraction $C = 0.1$ out of 100 clients (i.e., $K = 10$ clients) randomly selected clients to participate in training each round. All experiments are run with local epoch $E = 5$ for 300 rounds. We use SGD as the optimizer and tune the learning in range of [0.001, 0.1] for different algorithms.

Table 6: Testing accuracy comparison using MobileNetV2 as backbone on CIFAR-10 dataset with different levels of label skewness, where Dir(0.1) represents the most skewed label distribution 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**

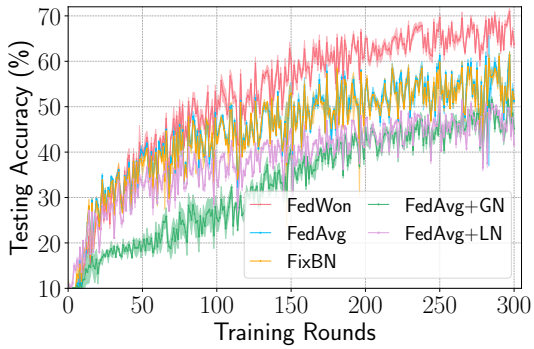


Figure 6: Changes of testing accuracy over the course of training on CIFAR-10 with Dir(0.5) using MobileNetV2 as backbone.

Performance Comparison. Table 6 compares our proposed FedWon with FedAvg, FedAvg+GN, FedAvg+LN, and FixBN. FedWon achieves similar performance as FedAvg and FixBN on i.i.d setting, but outperforms all methods on different degrees of label skewness. We do not compare FedBN and SiloBN as they are not suitable for cross-device FL. Figure 6 shows changes of testing accuracy over the course of training under Dir(0.5) setting. FedWon converges to a better position than the other methods. These experiments indicate the possibility of employing our proposed FL without normalization to solve the skewed label distribution problem.

6 Conclusion

In conclusion, we propose FedWon, a new approach for multi-domain FL by removing BN layers from DNNs 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 on 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 edge devices with memory constraints in cross-device FL. We believe that this work sheds light on the possibility of training models in FL without normalizations.

References

[Andreux *et al.*, 2020] Mathieu Andreux, Jean Ogier du Terrail, Constance Beguier, and Eric W. Tramel. Siloed federated learning for multi-centric histopathology datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pages 129–139. Springer International Publishing, 2020.

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

[Bernecker *et al.*, 2022] Tobias Bernecker, Annette Peters, Christopher L Schlett, Fabian Bamberg, Fabian Theis, Daniel Rueckert, Jakob Weiß, and Shadi Albarqouni. Fednorm: Modality-based normalization in federated learning for multi-modal liver segmentation. *arXiv preprint arXiv:2205.11096*, 2022.

[Bjorck *et al.*, 2018] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.

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

[Brock *et al.*, 2021b] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021.

[Casella *et al.*, 2023] Bruno Casella, Roberto Esposito, Antonio Sciarappa, Carlo Cavazzoni, and Marco Aldinucci. Experimenting with normalization layers in federated learning on non-iid scenarios. *arXiv preprint arXiv:2303.10630*, 2023.

[Chang *et al.*, 2019] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.

[Chen *et al.*, 2022] Chen Chen, Yuchen Liu, Xingjun Ma, and Lingjuan Lyu. Calfat: Calibrated federated adversarial training with label skewness. *Advances in Neural Information Processing Systems*, 2022.

[De and Smith, 2020] Soham De and Sam Smith. Batch normalization biases residual blocks towards the identity function in deep networks. *Advances in Neural Information Processing Systems*, 33:19964–19975, 2020.

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

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

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

[Griffin *et al.*, 2007] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[Hard *et al.*, 2018] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

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

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

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

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

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

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

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

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

[Krizhevsky *et al.*, 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep

convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2016] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[Li *et al.*, 2019] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 133–141. Springer, 2019.

[Li *et al.*, 2020a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37:50–60, 2020.

[Li *et al.*, 2020b] 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.

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

[Lu *et al.*, 2022] Wang Lu, Jindong Wang, Yiqiang Chen, Xin Qin, Renjun Xu, Dimitrios Dimitriadis, and Tao Qin. Personalized federated learning with adaptive batchnorm for healthcare. *IEEE Transactions on Big Data*, 2022.

[McMahan *et al.*, 2017] 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.

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

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

[Paulik *et al.*, 2021] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*, 2021.

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

[Saenko *et al.*, 2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010.

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

[Santurkar *et al.*, 2018] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[Tan and Le, 2019] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[Wang *et al.*, 2023] Yanmeng Wang, Qingjiang Shi, and Tsung-Hui Chang. Why batch normalization damage federated learning on non-iid data? *arXiv preprint arXiv:2301.02982*, 2023.

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

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

[Zhang *et al.*, 2019] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

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

[Zhong *et al.*, 2023] Jike Zhong, Hong-You Chen, and Wei-Lun Chao. Making batch normalization great in federated deep learning. *arXiv preprint arXiv:2303.06530*, 2023.

[Zhuang *et al.*, 2020] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 955–963, 2020.