This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

A Simple Data Augmentation for Feature Distribution Skewed Federated Learning

Yunlu Yan[1], Huazhu Fu[2], Yuexiang Li[3], Jinheng Xie[4], Jun Ma[1,6], Guang Yang[5], Lei Zhu[1,6*]
[1]The Hong Kong University of Science and Technology (Guangzhou) [2]IHPC, A*STAR
[3]Guangxi Medical University [4]National University of Singapore [5]Imperial College London
[6]The Hong Kong University of Science and Technology

Abstract

Federated Learning (FL) facilitates collaborative learning among multiple clients in a distributed manner and ensures the security of privacy. However, its performance inevitably degrades with non-Independent and Identically Distributed (non-IID) data. In this paper, we focus on the feature distribution skewed FL scenario, a common non-IID situation in real-world applications where data from different clients exhibit varying underlying distributions. This variation leads to feature shift, which is a key issue of this scenario. While previous works have made notable progress, few pay attention to the data itself, i.e., the root of this issue. The primary goal of this paper is to mitigate feature shift from the perspective of data. To this end, we propose a simple yet remarkably effective input-level data augmentation method, namely FedRDN, which randomly injects the statistical information of the local distribution from the entire federation into the client's data. This is beneficial to improve the generalization of local feature representations, thereby mitigating feature shift. Moreover, our FedRDN is a plug-and-play component, which can be seamlessly integrated into the data augmentation flow with only a few lines of code. Extensive experiments on several datasets show that the performance of various representative FL methods can be further improved by integrating our FedRDN, demonstrating its effectiveness, strong compatibility and generalizability. Code is available at https://github.com/IAMJackYan/FedRDN.

1. Introduction

Federated Learning [17, 20, 25] (FL) has been a de facto solution for distributed learning and attracted wide attention from various communities [2, 3, 13, 29]. It utilizes a credible server to communicate the privacy irrelevant information, *e.g.*, model parameters, thereby collaboratively training the model on multiple distributed data clients, while the local data of each client is invisible to others. Such a design is simple yet can achieve superior performance and preserve privacy. However, it is inevitable to suffer non-Independent and Identically Distributed (non-IID) data when we deploy FL in real-world applications, which will greatly degrade the performance of the trained model [9, 21, 26].

Since non-IID data is widely common in many real-world cases, many researchers try to address this issue and improve the practicability of FL [6, 9, 21, 31]. However, most of them try to design a general FL method that can handle different non-IID scenarios. In fact, this is suboptimal due to the discrepancy among various non-IID scenarios, such as feature distribution skew and label distribution skew [18]. Hence, recent studies try to design special FL algorithms for different non-IID scenarios [44, 48]. In this work, we focus on solving feature distribution skew (Definition in §3.1), a typical non-IID scenario in FL, as the data of discrete clients are always collected from different devices or environments, incurring different underlying distributions. For instance, different hospitals possess MR images scanned by different devices, and different phones store images in different styles (*e.g.*, cartoon images and natural images). Such data situation leads to inconsistent feature distribution across different clients, namely ***feature shift*** [22, 48], resulting in serious performance degradation. To address this problem, FedBN [22] learns the specific batch normalization layer parameters for each client. HarmoFL [8] utilizes the potential knowledge of medical images in the frequency domain to mitigate feature shift.

While feature distribution skew has been explored in FL, little attention has been paid to the data itself. When it comes to data processing, the common practice [8, 22, 48] is to integrate some traditional data operations in the data augmentation flow (*e.g.*, *transforms.Compose*() in Pytorch) of each client like flipping, cropping and rotating. This overlooks the effectiveness of data augmentation for FL at the input level, *i.e.*, data augmentation flow, which leaves an alternative space to explore. Since the root cause of feature shift lies in the divergence of local data distributions, we ask

*Lei Zhu (leizhu@ust.hk) is the corresponding author.

two questions: 1) ***Why not proactively resolve it by directly processing the data?*** 2) ***Can we design an FL-specific data augmentation technology that can be integrated into the data augmentation flow?*** To answer these questions, we try to design a plug-and-play input-level data augmentation technique.

Although it may appear straightforward, effectively implementing data augmentation in FL poses a significant challenge, due to the lack of direct access to the external data of other clients. Therefore, how to inject global information into augmented samples and thereby mitigate the distribution bias between different local datasets is the core of the challenge. In this regard, FedMix [40] extends Mixup to FL for label distribution skew by incorporating the mixing of averaged data across multiple clients. However, it is only suitable for the classification task and its effectiveness for feature distribution skew has not been demonstrated. Furthermore, permitting the exchange of averaged data introduces privacy concerns. In a more recent study, a data augmentation approach, namely FedFA [48], is proposed for feature distribution skew, which mitigates feature shift through feature augmentation based on the statistics of latent features. Even though, it requires modifications to the network structure, which cannot be seamlessly integrated into the data augmentation flow. Besides, FedFA adds additional computational and communication overhead, which can be constrained in resource-limited scenarios.

In this work, we propose a novel federated data augmentation technique, called FedRDN. We argue that the local bias is rooted in the model trained on the limited and skewed distribution. In centralized learning, we can collect data from many different distributions to learn generalized feature representations. This motivates us to augment the data to a more abundant distribution, which indirectly reduces the difference between different local distributions. To achieve that in the setting of FL, our FedRDN extends the standard data normalization to FL by randomly injecting statistics of local datasets into augmented samples, which is based on the insight that statistics of data are the essential characteristic of data distribution. It enables clients to access a wider range distribution of data, thereby enhancing the generalizability of local feature representations. Our FedRDN is a remarkably simple yet surprisingly effective method. It is a non-parametric approach that incurs negligible computation and communication overhead, without requiring modifications to the network structure, and seamlessly integrates into the data augmentation pipeline with just a few lines of code. After employing our method, significant improvements have been observed across various typical FL methods. In a nutshell, our contributions are summarized as follows:

- We explore the input-level data augmentation technique for feature distribution skew, which gives more insights into how to understand and solve this problem.
- We propose a novel plug-and-play data augmentation technique, FedRDN, which can be easily integrated into the data augmentation flow to mitigate feature shift for feature distribution skewed FL.
- We conduct extensive experiments on two classification datasets and an MRI segmentation dataset to demonstrate the effectiveness and generalization of our method, *i.e.*, it outperforms traditional data augmentation techniques and improves the performance of various typical FL methods.

2. Related Work

Federated Learning with Non-IID Data. FL [17, 38, 39, 45, 46] allows multiple discrete clients to collaborate in training a global model while preserving privacy. The pioneering work, FedAvg [27], is the most widely used FL algorithm, yielding success in various applications. However, the performance of FedAvg is inevitably degraded when suffering non-IID data [9, 21], posing a fundamental challenge in this field. To address this, a variety of novel FL frameworks have been proposed, with representative examples such as FedProx [21], Scaffold [9], and FedNova [37]. They improved FedAvg by modifying the local training [1, 16] or model aggregation [23, 42] to increase stability in heterogeneous data environments. Despite the progress, most of them ignore the differences in various non-IID scenarios, such as the heterogeneity of label distribution skew and feature distribution skew, which lie in labels and images [20], respectively. Therefore, recent studies attempt to more targeted methods for different non-IID scenarios. For example, FedLC [44] addressed label distribution skew by logits calibration. As for feature distribution skew, the target of this work, FedBN [22] learned the heterogeneous distribution of each client by personalized batch normalization layer parameters. In addition, HarmoFL [8] investigated specialized knowledge of frequency to mitigate feature shift. More recently, FedPCL [34] employed a pre-trained model to reduce the number of learnable parameters and applied prototype-wise contrastive learning to regularize feature representation learning across different clients. This offers an effective solution for training large models in the feature distribution skewed FL scenario. Motivated by prototype learning, FPL [7] utilized clustering to acquire unbiased class prototypes and then alleviated feature shift through prototype and local embedding alignment. In contrast to the aforementioned methods, ADCOL [19] introduced a novel adversarial collaborative learning approach to mitigate feature shift by replacing the model-averaging scheme with adversarial learning. Although achieving significant progress, they only focus on mitigating feature shift at the local optimization or global aggregation stage, neglecting the data itself which is the root of feature shift.

Different from them, we aim to address this issue from the perspective of data.

Data Augmentation Data augmentation [11, 12, 30] is a widely used technique in machine learning, which can alleviate overfitting and improve the generalization of the model. For computer vision tasks, the neural networks are typically trained with a data augmentation flow containing various techniques like random flipping, cropping, and data normalization. Different from these early label-preserving techniques [47], label-perturbing techniques are recently popular such as MIXUP [43] and CUTMIX [41]. It augments samples by fusing not only two different images but also their labels. Except for the above input-level augmentation techniques, some feature-level augmentation techniques [14, 15, 36] that make augmentation in feature space have achieved success. Recently, some studies have started to introduce data augmentation techniques into FL. For example, FedMix [40] proposed a variant of MIXUP in FL, which shows its effectiveness in the label distribution skewed FL scenario. However, its effectiveness in the feature distribution skewed FL scenario has not been demonstrated. More importantly, FedMix requires sharing averaged local data which increases the risk of privacy. For feature distribution skew, FedFA [48] augmented the features of clients by the statistics of features. However, it operates on the feature level, which requires modifications to the network structure and incurs additional computation and communication cost. Different from FedFA, our proposed augmentation method operates on the input level. By this, our method can be a plug-and-play component, combined with different FL methods to further improve their performance, which shows stronger compatibility and flexibility. In addition, our method can further improve the performance of FedFA due to operating in different spaces.

3. Methodology

3.1. Preliminaries

Federated Learning. Supposed that a FL system is composed of K clients $\{C_1, C_2, \ldots, C_K\}$ and a central server. For client C_k ($k \in [K]$), there are n_k supervised training samples $\{x_i, y_i\}_{i=1}^{n_k}$, where image x_i and label y_i from a joint distribution $(x_i, y_i) \sim P_k(x, y)$. Besides, each client trains a local model $f(\boldsymbol{w}_k)$ only on its private dataset, where $\boldsymbol{w}_k$ is the parameters of the local model. The goal of FL is to learn a global model by minimizing the summation empirical risk of each client:

$$\min \mathcal{L} = \sum_{k=1}^{K} \gamma_k \mathcal{L}_k, \quad \text{where} \quad \gamma_k = \frac{n_k}{\sum_{i=1}^{K} n_i}. \tag{1}$$

To achieve this goal, at each communication round $t \in [T]$, the standard FL method, FedAvg [27], performs E epochs local training and the local objective $\mathcal{L}_k$ can be written as:

$$\mathcal{L}_k = \frac{1}{n_k} \sum_{(x_i, y_i) \sim P_k} \ell(f(x_i; \boldsymbol{w}_k^t); y_i), \tag{2}$$

where ℓ is the loss function. After local training, it then averages the parameters of all local models to update the parameters of the global model, which can be described as:

$$\boldsymbol{w}_G^{t+1} = \sum_{k=1}^{K} \gamma_k \boldsymbol{w}_k^t. \tag{3}$$

The updated parameters of global model $\boldsymbol{w}_G^{t+1}$ will be returned to each client as the initialization for the next round of training.

Feature Distribution Skew. The underlying data distribution $P_k(x, y)$ can be rewritten as $P_k(y|x)P_k(x)$, and $P_k(x)$ varies across clients while $P_k(y|x)$ is consistent for all clients. Moreover, different underlying data distributions lead to inconsistent feature distribution across different clients, thereby degrading the performance of the global model.

Data Normalization. Normalization is a popular data preprocess operation, which can transform the data distribution to standard normal distribution. In detail, given an C-channel image $x \in \mathbb{R}^{C \times H \times W}$ with spatial size $(H \times W)$, it transforms image as:

$$\hat{x} = \frac{x - \mu}{\sigma}, \quad \mu, \sigma \in \mathbb{R}^C, \tag{4}$$

where μ and σ are channel-wise means and standard deviation, respectively, and they are usually manually set in experiential or statistical values from the real dataset.

3.2. Federated Random Data Normalization

In this section, we present the details of the proposed **F**ederated **R**andom **D**ata **N**ormalization (FedRDN) method. Different from previous FL methods, FedRDN focuses on mitigating the distribution discrepancy through input-level data augmentation during the data processing stage. The goal of FedRDN is to let each client learn as many distributions as possible instead of self-biased distribution, which is beneficial to feature generalization. To achieve this, it performs explicit data augmentation by manipulating multiple-clients channel-wise data statistics during training at each client. We will introduce the detail of our method in the following.

Data Distribution Statistic. The approximate distribution of the data can be estimated using statistical methods. Therefore, we can obtain an approximate distribution by computing the statistics of the local dataset, *i.e.*, $P_k \sim \mathcal{N}(\mu^k, (\sigma^k)^2)$, where μ^k and σ^k are mean and standard deviation, respectively. Specifically, to estimate such

Algorithm 1: FedRDN

Input: Number of Clients K, communication rounds T, epochs E

Compute Statistic:
for *client* $k = 1, 2, ..., K$ ***parallelly*** **do**
  for $x_i^k \sim P_k$ **do**
    $\mu_i^k = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} x_i^{k,(h,w)}$
    $\sigma_i^k = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_i^{k,(h,w)} - \mu_i^k)^2}$
  end
  $\mu^k = \sum_{i=1}^{n_k} \mu_i^k, \quad \sigma^k = \sum_{i=1}^{n_k} \sigma_i^k$
end
Return $\{(\mu^k, \sigma^k)\}_{k=1}^K$

Data Augmentation:
for *round* $t = 1, 2, ..., T$ **do**
  for *epoch* $e = 1, 2, ..., E$ **do**
    for $(x_i^k, y_i^k) \sim P_k$ **do**
      $(\mu^j, \sigma^j) \sim \{(\mu^k, \sigma^k)\}_{k=1}^K$
      $\hat{x}_i^k = \frac{x_i^k - \mu^j}{\sigma^j}$
      `// training`
    end
  end
end

underlying distribution information of each client, we compute the channel-wise statistics within each local dataset in client-side before the start of training:

$$\mu^k = \sum_{i=1}^{n_k} \mu_i^k \in \mathbb{R}^C, \quad \sigma^k = \sum_{i=1}^{n_k} \sigma_i^k \in \mathbb{R}^C, \tag{5}$$

where μ_i^k and σ_i^k are sample-level channel-wise statistics, and they can be computed as:

$$\begin{aligned} \mu_i^k &= \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} x_i^{k,(h,w)}, \\ \sigma_i^k &= \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_i^{k,(h,w)} - \mu_i^k)^2}, \end{aligned} \tag{6}$$

where $x_i^{k,(h,w)}$ represents the image pixel at spatial location (h, w). Following this, all data distribution statistics will be sent to the server and aggregated by the server. The aggregated statistics $\{(\mu^k, \sigma^k)\}_{k=1}^K$ are shared among clients.

Data Augmentation at Training Phase. After obtaining the statistical information of each client, we utilize them to augment data during training. Considering an image x_i^k, different from the normal data normalization that transforms the image according to a fixed statistic, we transform the image by randomly selecting the mean and standard deviation from statistics $\{(\mu^k, \sigma^k)\}_{k=1}^K$, which can be described as:

$$\hat{x}_i^k = \frac{x_i^k - \mu^j}{\sigma^j}, \quad (\mu^j, \sigma^j) \sim \{(\mu^k, \sigma^k)\}_{k=1}^K. \tag{7}$$

Notably, the statistic (μ^j, σ^j) will be randomly reselected for each image at each training epoch. Consequently, after multiple rounds, the number of selections surpasses the quantity of statistics by a substantial margin. This implies that each client leverages the local distribution information from all clients to augment each image, which potentially injects all local information from every client into the client-side training. In this way, we seamlessly inject global information into augmented samples. The local model can learn the distribution information of all clients, thereby making the learned features more generalized.

Data Processing at Testing Phase. During training, the random selection strategy integrates distribution information across all clients (*i.e.*, local statistics). This allows the model to remain robust to variations in client-specific distributions. However, at the testing stage, communication between clients should be disabled, and each client should operate independently, without access to distribution information from others. Thus, at test time, we only apply the local statistics pertinent to each individual client for data processing. This ensures the consistency between the distribution of training and testing data. For client k, the above process can be formalized as follows:

$$\hat{x}_i^k = \frac{x_i^k - \mu^k}{\sigma^k}. \tag{8}$$

The overview of two processes, *i.e.*, statistic computing and data augmentation, are presented in Algorithm 1.

3.3. Privacy Security

The previous input-level data augmentation method, *i.e.*, FedMix [40], shares the average images per batch, leading to the increased risk of privacy. Different from it, our method only shares the privacy irrelevant information, *i.e.*, dataset-level mean and standard deviation. In addition, we can not reverse the individual image from the shared information because it is statistical information of the whole dataset. Therefore, our method has a high level of privacy security.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct extensive experiments on three real-world datasets: **Office-Caltech-10** [4], **DomainNet** [28],

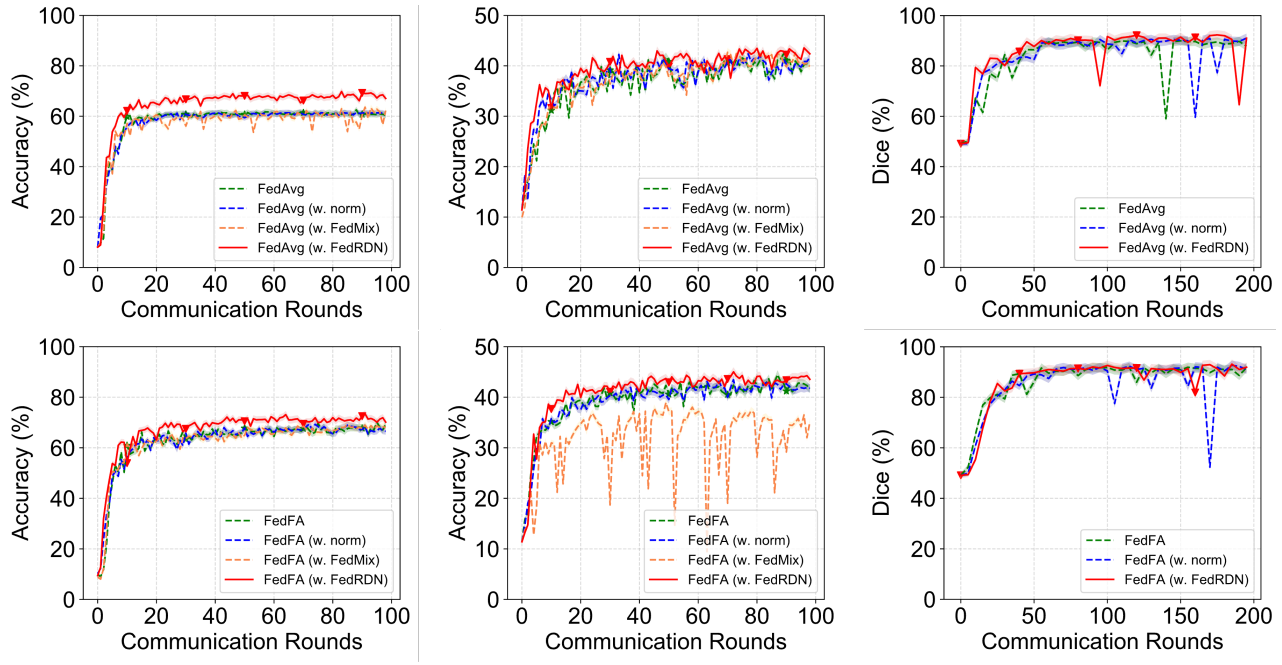
Table 1. **The test accuracy (%) of all approaches on office-Caltech-10 [4] and DomainNet [28]**. For a detailed comparison, we present the test accuracy of each client *i.e.*, **Office-Caltech-10**: A(Amazon), C(Caltech), D(DSLR), W(Webcam), **DomainNet**: C(Clipart), I(Infograph), P(Painting), Q(Quickdraw), R(Real), S(Sketch), and the average result. ↑ and ↓ show the rise and fall of the average result before and after augmentation. We mark best results in bold. (norm.: conventional data normalization)

Method	**Office-Caltech-10**					**DomainNet**						
	A	C	D	W	Avg.	C	I	P	Q	R	S	Avg.
FedAvg [27]	53.12	44.88	65.62	86.44	62.51	**50.38**	22.83	36.99	58.10	46.09	39.53	42.32
+ *norm*	50.52	43.55	68.75	83.05	61.46$_{(1.05)}$↓	48.28	23.28	37.80	54.20	48.97	**41.69**	42.37$_{(0.05)}$↑
+ *FedMix* [40]	49.47	41.77	75.00	88.13	63.59$_{(1.08)}$↑	48.66	**23.43**	38.12	55.10	49.46	41.33	42.68$_{(0.36)}$↑
+ ***FedRDN***	**60.93**	**45.77**	**84.37**	**88.13**	**69.80**$_{(7.29)}$↑	48.85	22.67	**39.41**	**60.30**	**49.46**	40.61	**43.55**$_{(1.23)}$↑
FedProx [21]	53.12	45.33	62.50	86.44	61.84	**52.66**	23.89	35.21	56.70	46.75	41.87	42.85
+ *norm*	51.04	**45.77**	68.75	84.74	62.57$_{(0.73)}$↑	47.14	24.35	34.57	59.60	44.86	38.98	41.58$_{(1.27)}$↓
+ *FedMix* [40]	47.39	38.66	78.12	**91.52**	63.92$_{(2.08)}$↑	47.90	22.37	37.31	53.90	48.47	**43.14**	42.18$_{(0.67)}$↓
+ ***FedRDN***	**61.45**	44.88	**84.37**	88.13	**69.71**$_{(7.87)}$↑	50.57	**24.96**	**38.77**	**61.20**	**51.35**	40.97	**44.63**$_{(1.78)}$↑
FedNova [37]	50.00	42.22	62.50	88.13	60.71	**51.71**	23.74	38.77	56.20	45.52	38.44	42.39
+ *norm*	52.08	**45.33**	68.75	86.44	63.15$_{(2.44)}$↑	49.23	**24.35**	34.24	55.80	45.52	42.23	41.90$_{(0.49)}$↓
+ *FedMix* [40]	48.95	42.66	78.12	83.05	63.20$_{(2.49)}$↑	47.90	24.04	36.67	**59.10**	46.67	**42.41**	42.80$_{(0.41)}$↑
+ ***FedRDN***	**63.02**	41.33	**84.37**	**89.83**	**69.63**$_{(8.71)}$↑	50.57	23.43	**40.22**	57.30	**51.84**	40.43	**43.96**$_{(1.57)}$↑
Scaffold [9]	52.60	**42.66**	53.12	81.35	57.43	46.95	22.83	34.57	46.50	47.00	40.97	39.80
+ *norm*	46.87	40.00	59.37	86.44	58.17$_{(0.74)}$↑	47.33	22.83	33.11	58.30	46.01	**42.05**	41.61$_{(1.81)}$↑
+ *FedMix* [40]	52.08	40.88	75.00	**89.83**	64.45$_{(7.02)}$↑	45.24	23.28	34.73	47.50	44.78	40.97	39.42$_{(0.38)}$↓
+ ***FedRDN***	**65.10**	41.77	**81.25**	86.44	**68.64**$_{(11.21)}$↑	**51.52**	**23.89**	**37.96**	**56.20**	**48.97**	38.80	**42.89**$_{(3.09)}$↑
FedAvgM [6]	48.43	**45.33**	62.50	83.05	59.83	45.81	22.52	37.96	50.10	48.23	**41.87**	41.08
+ *norm*	51.04	44.88	62.50	86.44	61.21$_{(1.38)}$↑	46.95	24.96	35.86	49.70	45.43	40.43	40.55$_{(0.53)}$↓
+ *FedMix* [40]	50.00	41.77	65.62	83.05	60.11$_{(0.29)}$↑	**48.28**	**25.87**	40.06	51.50	48.56	38.62	42.15$_{(1.07)}$↑
+ ***FedRDN***	**62.50**	43.11	**84.37**	**88.13**	**69.53**$_{(9.70)}$↑	48.09	22.98	**41.03**	**63.80**	**49.79**	38.08	**43.96**$_{(2.88)}$↑
FedBN [22]	67.18	**44.00**	**84.97**	86.44	70.65	49.23	24.96	36.32	63.60	47.74	39.53	43.56
+ *norm*	64.84	41.07	84.11	87.87	69.47$_{(1.18)}$↓	48.47	23.81	36.67	**63.76**	**49.71**	40.79	43.88$_{(0.32)}$↑
+ *FedMix* [40]	65.62	44.88	84.37	88.45	70.83$_{(0.18)}$↑	49.61	**25.72**	**38.44**	57.30	46.01	38.08	42.53$_{(1.03)}$↓
+ ***FedRDN***	**67.74**	43.11	84.37	**89.35**	**71.14**$_{(0.49)}$↑	**50.76**	25.26	37.15	61.98	48.97	**43.14**	**44.54**$_{(0.98)}$↑
FedProto [33]	55.72	44.44	68.75	86.44	63.84	48.28	**25.11**	35.86	51.30	43.79	37.18	40.25
+ *norm*	53.64	44.88	56.25	86.44	60.30$_{(3.54)}$↓	45.81	23.43	35.70	**58.30**	45.27	40.79	41.55$_{(1.30)}$↑
+ *FedMix* [40]	53.64	41.77	84.37	88.13	66.98$_{(3.14)}$↑	47.33	23.43	37.47	52.70	44.94	**42.41**	41.38$_{(1.13)}$↑
+ ***FedRDN***	**66.14**	**46.22**	**84.37**	**89.83**	**71.64**$_{(7.80)}$↑	**49.42**	22.37	**41.51**	57.90	**51.43**	38.44	**43.51**$_{(3.26)}$↑
FedFA [48]	60.93	48.44	81.25	89.83	70.11	48.09	23.74	**39.58**	**64.20**	48.06	**43.14**	44.47
+ *norm*	60.93	**50.66**	81.25	84.74	69.40$_{(0.71)}$↓	49.04	24.20	38.28	60.70	45.93	42.59	43.46$_{(1.01)}$↓
+ *FedMix* [40]	56.25	48.44	84.37	88.13	69.30$_{(0.81)}$↓	45.81	22.52	33.11	51.10	43.13	37.90	38.93$_{(5.54)}$↓
+ ***FedRDN***	**62.50**	48.88	**90.62**	**91.52**	**73.38**$_{(3.27)}$↑	**52.85**	**24.50**	37.80	61.90	**50.45**	42.59	**45.01**$_{(0.54)}$↑

and **ProstateMRI** [24], which are widely used in the feature distribution skewed FL scenario [8, 22, 48]. There are two different tasks including image classification (Office-Caltech-10 and DomainNet) and medical image segmentation (ProstateMRI). Following previous work [22, 48], we employ the subsets as clients when conducting experiments on each dataset.

Baselines. To demonstrate the effectiveness of our method, we build four different data augmentation flows: ❶ ***base***: one flow has some basic data augmentation techniques like random flipping, ❷ ***+norm*****:** one flow adds the conventional normalization technique into *base*, ❸ ***+FedMix*****:** another flow adds the FedMix [40] data augmentation technique into *base*, and ❹ ***+FedRDN*****:** the rest one integrates our proposed augmentation method with *base*. Since it is infeasible to deploy FedMix into segmentation tasks, we only utilize it for image classification tasks. Following, we integrate them into different typical FL methods. In detail, we employ eight state-of-the-art FL methods to demonstrate the generalizability of our method, including **FedAvg** [27], **FedAvgM** [6], **FedProx** [21], **Scaffold** [9], **FedNova** [37], **FedBN** [22], **FedProto** [33], and **FedFA** [48] for validation of image classification task. Moreover, we select five of them which are general for different tasks to validate the effectiveness of our method on medical image segmentation tasks. To quantitatively evaluate the performance, we utilize the top-1 accuracy for image classification while the medical segmentation is evaluated with the Dice coefficient.

Implementation Details. All methods are implemented by PyTorch, and we conduct all experiments on a single NVIDIA GTX 1080Ti GPU with 11GB of memory. Following previous work [22, 48], we employ the AlexNet [10] as the image classification model and the U-Net [32] as the medical image segmentation model. We use cross-entropy loss for image classification and Dice loss for medical image segmentation. The batch size is 32 for two image classification datasets and 16 for the ProstateMRI dataset. We adopt the SGD optimizer with a learning rate



(a) Office-Caltech-10 (b) DomainNet (c) ProstateMRI

Figure 1. **Illustration of test performance versus communication rounds** on (a) Office-Caltech-10 [4], (b) DomainNet [28], and (c) ProstateMRI [24].

of 0.01 and weight decay of 1e-5 for image classification datasets, and the Adam optimizer with a learning rate of 1e-3 and weight decay of 1e-4 for ProstateMRI. Furthermore, we run 100 communication rounds on image classification tasks while the number of rounds for the medical segmentation task is 200, and each round has 5 epochs of local training. More importantly, for a fair comparison, we train all methods in the same environment and ensure that all methods have converged. Due to page limitation, we present additional experimental details and results in ***Supplementary Material***.

4.2. Main Results

In this section, we present the overall results on Office-Caltech-10 and DomainNet in Table 1 and ProstateMRI in Table 2. For a detailed comparison, we present the test accuracy of each client and the average result.

All FL methods yield significant improvements combined with FedRDN consistently over three datasets. As we can see, FedRDN leads to consistent performance improvement for all FL baselines across three benchmarks compared with using the basic data augmentation flow. The improvements of FedRDN can be large as **11.21%** on Office-Caltech-10, **3.26%** on DomainNet, and **1.37%** on ProstateMRI, respectively. Especially for FedFA, the state-of-the-art FL method for feature distribution skewed FL setting, can still gain improvements, *e.g.*, **3.27%** on Office-Caltech-10. This indicates that input-level augmentation and feature-level augmentation are not contradictory and can be used simultaneously. Due to the utilization of personalized batch normalization layers, the advancements achieved with FedBN might not be as pronounced as with other methods. However, it's crucial to note that there are still modest improvements present. Moreover, some weaker FL methods can even achieve better performance than oth-

Table 2. **The dice score (%) of all approaches on ProstateMRI [24]**. For a detailed comparison, we present the test results of six clients: BIDMC, HK, I2CVB, BMC, RUNMC, UCL, and the average result. ↑ and ↓ show the rise and fall of the average result before and after augmentation. We mark best results in bold. (norm.: conventional data normalization)

Method	ProstateMRI						
	BIDMC	HK	I2CVB	BMC	RUNMC	UCL	Avg.
FedAvg [27]	84.16	**94.51**	94.60	88.43	92.78	85.64	90.02
+ *norm*	86.20	92.53	**94.74**	89.85	92.18	87.91	90.57 $_{(0.55)}$↑
+ ***FedRDN***	**89.34**	94.41	93.85	**91.46**	**94.19**	**90.65**	**92.32** $_{(2.30)}$↑
FedProx [21]	84.47	**94.60**	94.87	90.29	92.72	86.60	90.59
+ *norm*	84.47	94.48	95.06	88.79	92.90	85.35	90.18 $_{(0.41)}$↓
+ ***FedRDN***	**88.87**	94.19	**95.09**	**90.99**	**93.03**	**89.17**	**91.89** $_{(1.30)}$↑
FedAvgM [6]	87.02	94.32	94.29	**91.35**	92.83	86.75	91.09
+ *norm*	**89.05**	93.59	94.75	89.93	**93.52**	88.22	91.51 $_{(0.42)}$↑
+ ***FedRDN***	88.37	**94.67**	**95.40**	90.40	93.28	**88.39**	**91.75** $_{(0.66)}$↑
FedBN [22]	86.42	**94.45**	95.27	90.96	93.13	87.48	91.28
+ *norm*	87.45	93.01	95.44	90.16	93.22	86.88	91.02 $_{(0.26)}$↓
+ ***FedRDN***	**89.47**	93.61	**95.65**	**90.99**	**93.26**	**87.72**	**91.78** $_{(0.50)}$↑
FedFA [48]	89.18	92.77	94.18	**92.62**	93.63	89.04	91.90
+ *norm*	89.12	94.40	95.22	91.95	93.42	89.28	92.23 $_{(0.33)}$↑
+ ***FedRDN***	**91.81**	**94.65**	**95.67**	92.37	**94.33**	**90.19**	**93.14** $_{(1.24)}$↑

Table 3. **Generalization performance of local model** on Office-Caltech-10 [4]. We mark the best result in bold.

Source-site	Target-site	FedAvg [27]	+ *norm*	+ *FedMix* [40]	+ ***FedRDN***
Amazon	Caltech	48.00	48.09	42.22	**48.88**
	DSLR	46.87	44.41	59.37	**81.25**
	Webcam	77.96	77.96	**84.74**	83.05
Caltech	Amazon	58.33	58.85	51.56	**63.02**
	DSLR	68.75	68.75	75.00	**84.37**
	Webcam	83.05	83.05	**91.52**	86.35
DSLR	Amazon	42.70	41.14	33.85	**60.93**
	Caltech	33.33	35.11	35.55	**35.56**
	Webcam	79.66	77.96	74.57	**84.74**
Webcam	Amazon	53.64	55.20	47.91	**65.10**
	Caltech	41.33	43.55	40.00	**48.00**
	DSLR	68.75	68.75	78.12	**87.50**

ers when using FedRDN. For example, after using FedRDN, the accuracy of FedAvg can be significantly higher than all other FL methods except FedFA, and FedProto can be even higher than FedFA. The above results demonstrate the effectiveness of the data-level solution, which can effectively mitigate feature shift. Besides, compared with other heterogeneous FL methods, our method has a stronger flexibility and generalization ability.

FedRDN is superior to other input-level data augmentation techniques. As shown in Table 1 and 2, FedRDN shows leading performance compared with conventional data normalization and FedMix, a previous input-level data augmentation technique. Besides, these two augmentation techniques can even decrease the performance of the method in several cases, while FedRDN can achieve consistent improvements. For instance, FedAvg and FedProto yield a drop as large as **1.05%** and **3.54%** with conventional data normalization on Office-Caltech-10, respectively. FedProx and FedFA show a drop as **0.67%** and **5.54%** on DomainNet, respectively, when they are combined with FedMix. The above results demonstrated the effectiveness of FedRDN, and it has stronger compatibility compared with other data augmentation methods.

4.3. Communication Efficiency

Convergence. To explore the impact of various data augmentation techniques on the convergence, we draw the test performance curve of FedAvg and a state-of-the-art FL method, *i.e.*, FedFA, with different communication rounds on three datasets as shown in Fig. 1. Apparently, FedRDN will not introduce any negative impact on the convergence of the method and even yield a faster convergence at the early training stage ($0 \sim 10$ rounds) in some cases. As the training goes on, FedRDN achieves a more optimal solution. Besides, compared with other methods, the convergence curves of FedRDN are more stable.

Communication Cost. In addition to the existing communication overhead of FL methods, the additional communication cost in FedRDN is only for statistical information. The dimension of the statistic is so small (mean and standard deviation are $\mathbb{R}^3$ for RGB images), that the increased communication cost can be neglected. This is much different from the FedMix, which needs to share the average images per batch. The increased communication cost of FedMix is as large as 156MB on Office-Caltech-10 and 567MB on DomainNet while the batch size of averaged images is 5, even larger than the size of model parameters.

Table 4. **The performance of FedRDN over FedRDN-V** on three datasets.

Method	**Office-Caltech-10**	**DomainNet**	**ProstateMRI**
FedAvg [27]	62.51	42.32	90.02
+ *FedRDN-V*	61.46	42.99	91.14
+ ***FedRDN* (Ours)**	**69.80**	**43.55**	**92.32**

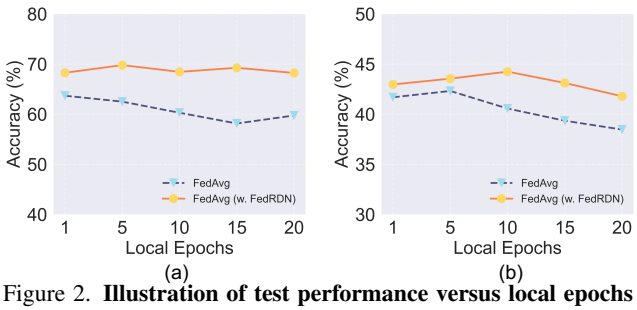


Figure 2. **Illustration of test performance versus local epochs** on (a) Office-Caltech-10 [4] and (b) DomainNet [28].

4.4. Cross-site Generalization performance

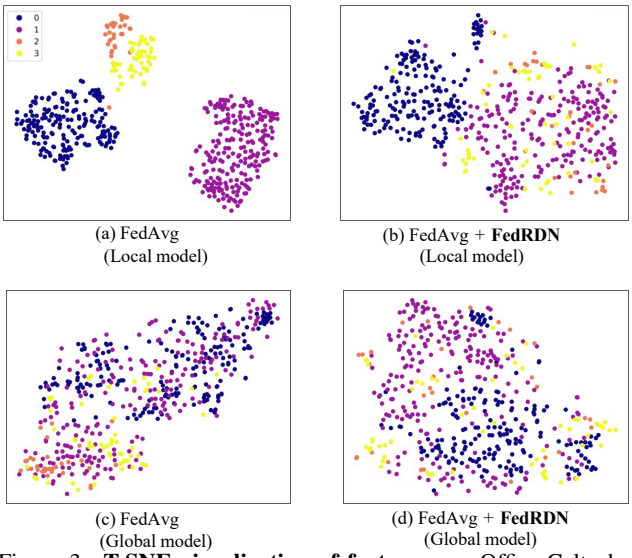
As stated before, FedRDN augments the data with luxuriant distribution from all clients to learn the more generalized feature representation. Therefore, we further explore the generalization performance of local models by cross-site evaluation, and the results are presented in Table 3. As we can see, All local models of FedRDN yield a consistent improvement over FedAvg, and our method shows better generalization performance compared to other methods. The above results demonstrate that FedRDN can effectively improve the generalization of local feature representation, thereby mitigating feature shift across different clients. This is the reason why our method works.

4.5. Analytical Studies

FedRDN vs. FedRDN-V. To deeply explore FedRDN, we develop a variant, FedRDN-V. Instead of randomly transforming, it transforms the images with the average mean $\hat{u}$ and standard deviation $\hat{\sigma}$ of all clients during training and testing phases:

$$\hat{u} = \sum_{k=1}^{K} \mu_k, \quad \hat{\sigma} = \sum_{k=1}^{K} \sigma_k. \tag{9}$$

This degenerates into traditional data normalization with the hyper-parameter $\hat{u}$ and $\hat{\sigma}$. The results of the comparison over three datasets are presented in Table 4. Apparently, FedRDN-V yields a significant drop compared with our method. This indicates the effectiveness of our method



(a) FedAvg (Local model) (b) FedAvg + **FedRDN** (Local model)

(c) FedAvg (Global model) (d) FedAvg + **FedRDN** (Global model)

Figure 3. **T-SNE visualization of features** on Office-Caltech-10 [4]. The T-SNE is conducted on the test sample features of four clients. We use different colors to mark features from different clients.

is not from the traditional data normalization but augmenting samples with the information from the multiple real local distributions. By this, each local model will be more generalized instead of biasing in skewed underlying distributions.

Robust to Local Epochs. To explore the robustness of FedRDN for different local epochs, we tune the local epochs from the {1, 5, 10, 15, 20} and evaluate the performance of the trained model. The results are presented in Fig. 2. Generally, more epochs of local training will increase the discrepancy under non-IID data, leading to slower convergence, which degrades the performance at the same communication rounds. The result of FedAvg validates this. By contrast, our method can obtain consistent improvements with different settings of local epochs. Moreover, our approach has stable performance across different local epoch settings due to effectively addressing feature shift.

Varying Architectures. To validate the robustness of our method for different network architectures, we conduct additional experiments for different input-level augmentation methods using ResNet-18 [5] on Office-Caltech-10. We only change the network and do not alter any other experimental setup. The results are presented in Table 5. Due to the small size of the dataset, the performance of ResNet-18 is not as good as that of AlexNet. This is consistent with the findings reported in [49]. However, the results still show the consistent improvements obtained through our proposed augmentation approach, indicating its strong robustness and adaptability for different network architectures.

Feature Distribution. To yield more insights about FedRDN, we utilize T-SNE [35] , a popular analysis tool, to visualize the distribution of the features outputted by the

Table 5. **Accuracy(%) of input-level data augmentation methods with varying architectures** on Office-Caltech-10 [4].

Network	FedAvg [27]	+ *norm*	+ *FedMix* [40]	+ ***FedRDN***
AlexNet	62.51	61.46	63.59	**69.80**
ResNet-18	43.74	43.60	53.81	**56.70**

local model (Fig. 3 (a) and (b)) and global model (Fig. 3 (c) and (d)) before and after augmentation. Specifically, we visualize the features of test samples for each client and mark them with different colors: 0 (Amazon), 1 (Caltech), 2 (DSLR), and 3 (Webcam). Due to the different data distributions among these clients, their local feature distributions exhibit a shift (Fig. 3 (a)). This makes it challenging for the global model to learn consistent feature distributions across different clients. Intuitively, as shown in Fig. 3 (c), the features of client 2 (DSLR) are clustered in the bottom-left corner, showing a noticeable shift from the features of other clients. In contrast, our method significantly improves the feature generalization of local models, narrowing the distance between different local feature distributions (Fig. 3 (b)). This benefits the global model to learn a more consistent feature distribution. As shown in Fig. 3 (d), the global features of all four clients are uniformly distributed, indicating that the features from these clients are now in a shared feature space. The above results reveal the working principle of FedRDN and further confirm its effectiveness in mitigating feature shift.

5. Conclusion

In this paper, we focus on addressing the feature distribution skew in FL. Different from previous insights for this problem, we try to solve this challenge from the input-level data. The proposed novel data augmentation technique, FedRDN, is a plug-and-play component that can be easily integrated into the data augmentation flow, thereby effectively mitigating feature shift. Our extensive experiments show that FedRDN can further improve the performance of various state-of-the-art FL methods across three datasets, which demonstrates the effectiveness, compatibility, and generalizability of our method.

6. Limitation

This work primarily focuses on visual tasks, where data statistics are privacy-agnostic information, as they only capture the distribution information instead of individual-level information. Besides, we mainly demonstrated the effectiveness of FedRDN in the feature distribution skewed FL scenario by the empirical analysis. In the future, we plan to undertake an in-depth theoretical analysis to further enhance the understanding and explanation of our method. Considering the effectiveness, compatibility, and generalizability of FedRDN, we believe this work contributes substantively to advancing the development of FL.

Acknowledgment

This work is supported by the Guangdong Science and Technology Department (No. 2024ZDZX2004), the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0671), the H. Fu's Agency for Science, Technology and Research (A*STAR) Central Research Fund ("Robust and Trustworthy AI system for Multi-modality Healthcare"), and the RIE2025 Industry Alignment Fund - Industry Collaboration Project (IAF-ICP) (Award No: I2301E0020), administered by A*STAR.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[2] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10164–10173, 2022.

[3] Xiuwen Fang and Mang Ye. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10081, 2022.

[4] 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.

[5] 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.

[6] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[7] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16312–16322. IEEE, 2023.

[8] Meirui Jiang, Zirui Wang, and Qi Dou. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1087–1095, 2022.

[9] 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.

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

[11] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[12] Teerath Kumar, Muhammad Turab, Kislay Raj, Alessandra Mileo, Rob Brennan, and Malika Bendechache. Advanced data augmentation approaches: A comprehensive survey and future directions. *arXiv preprint arXiv:2301.02830*, 2023.

[13] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. *Advances in Neural Information Processing Systems*, 2021.

[14] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12383–12392, 2021.

[15] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[16] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10713–10722, 2021.

[17] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2021.

[18] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

[19] Qinbin Li, Bingsheng He, and Dawn Song. Adversarial collaborative learning on non-IID features. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19504–19526. PMLR, 2023.

[20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[21] 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.

[22] 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.

[23] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[24] Quande Liu, Qi Dou, Lequan Yu, and Pheng Ann Heng. Ms-net: multi-site network for improving prostate segmentation with heterogeneous mri data. *IEEE transactions on medical imaging*, 39(9):2713–2724, 2020.

[25] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[26] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

[27] 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.

[28] 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.

[29] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2020.

[30] Rui Qian, Weiyao Lin, John See, and Dian Li. Controllable augmentations for video representation learning. *Visual Intelligence*, 2(1):1, 2024.

[31] Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33:21554–21565, 2020.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[33] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8432–8440, 2022.

[34] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems*, 35:19332–19344, 2022.

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11), 2008.

[36] Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19174–19183, 2022.

[37] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

[38] Yunlu Yan, Hong Wang, Yawen Huang, Nanjun He, Lei Zhu, Yong Xu, Yuexiang Li, and Yefeng Zheng. Cross-modal vertical federated learning for mri reconstruction. *IEEE Journal of Biomedical and Health Informatics*, 28(11):6384–6394, 2024.

[39] Yunlu Yan, Lei Zhu, Yuexiang Li, Xinxing Xu, Rick Siow Mong Goh, Yong Liu, Salman Khan, and Chun-Mei Feng. A new perspective to boost performance fairness for medical federated learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 13–23. Springer, 2024.

[40] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. *arXiv preprint arXiv:2107.00233*, 2021.

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[42] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

[43] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[44] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning*, pages 26311–26329. PMLR, 2022.

[45] Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and S Yu Philip. A game-theoretic federated learning framework for data quality improvement. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):10952–10966, 2022.

[46] Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and S Yu Philip. A robust game-theoretical federated learning framework with joint differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3333–3346, 2022.

[47] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020.

[48] Tianfei Zhou and Ender Konukoglu. Fedfa: Federated feature augmentation. *International Conference on Learning Representations*, 2023.

[49] Weiming Zhuang and Lingjuan Lyu. Is normalization indispensable for multi-domain federated learning? *arXiv preprint arXiv:2306.05879*, 2023.