FRAug: Tackling Federated Learning with Non-IID Features via Representation Augmentation

Haokun Chen [1,2] Ahmed Frikha [1,2,3] Denis Krompass [2] Jindong Gu [4*] Volker Tresp [1,3]
[1] Ludwig Maximilian University of Munich [2] Siemens Technology
[3] Munich Center for Machine Learning [4] University of Oxford
{haokun.chen, ahmed.frikha, denis.krompass}@siemens.com,
jindong.gu@outlook.com, volker.tresp@lmu.de

Abstract

Federated Learning (FL) is a decentralized machine learning paradigm, in which multiple clients collaboratively train neural networks without centralizing their local data, and hence preserve data privacy. However, real-world FL applications usually encounter challenges arising from distribution shifts across the local datasets of individual clients. These shifts may drift the global model aggregation or result in convergence to deflected local optimum. While existing efforts have addressed distribution shifts in the label space, an equally important challenge remains relatively unexplored. This challenge involves situations where the local data of different clients indicate identical label distributions but exhibit divergent feature distributions. This issue can significantly impact the global model performance in the FL framework. In this work, we propose Federated Representation Augmentation (FRAug) to resolve this practical and challenging problem. FRAug optimizes a shared embedding generator to capture client consensus. Its output synthetic embeddings are transformed into client-specific by a locally optimized RTNet to augment the training space of each client. Our empirical evaluation on three public benchmarks and a real-world medical dataset demonstrates the effectiveness of the proposed method, which substantially outperforms the current state-of-the-art FL methods for feature distribution shifts, including PartialFed and FedBN.

1. Introduction

Federated Learning (FL) is a machine learning paradigm in which a shared model is collaboratively trained using decentralized data sources. In the classical FL approach, *e.g.*, FedAvg [49], the central server obtains the model by iteratively averaging the optimized model weights uploaded from the active clients. FL has the benefit that it does not require direct access to the client local datasets, resulting in improved client-server communication efficiency and enhanced data confidentiality.

Despite these promising prospects, real-world FL applications encounter practical challenges arising from data heterogeneity, in which the client local datasets are not independent and identically distributed (*non-IID*). Non-IID data from different clients may cause local model drifts during the client update and overfitting to its local objective, making it challenging to obtain a stable and optimal convergence of the aggregated server model [41, 50].

As discussed in [28], data heterogeneity in FL can be categorized into label space heterogeneity and feature space heterogeneity. A variety of methods were developed to tackle problem settings where the client datasets are non-IID in the label space [75, 66]. However, the under-explored problem of feature distribution shift is also prevalent in real-world applications, *e.g.*, in the data collected from different scanners in clinical centers [10], as well as gathered by different machines in industrial manufacturing plants [39]. Most importantly, although these entities may diagnose the same types of cancers or detect the same types of anomalies, *i.e.*, having the same label distribution, they are not willing to share their original data to prevent competitive disadvantage or reverse engineering. Therefore, we propose an effective and privacy-preserving FL algorithm, *i.e.*, Federated Representation Augmentation (*FRAug*), to address this practical problem of feature space heterogeneity.

Unlike previous works that generate synthetic samples in the input space [69, 68] or acquire additional public datasets [44, 17], FRAug applies data augmentation in the low-dimensional feature embedding space, which is more efficient and confronts fewer confidentiality threats. Moreover, the proposed augmentation algorithm is especially suitable for FL applications, where collaborative training is often conducted by multiple edge devices (clients) with limited

*Corresponding author

computational powers and data quantities [49]. Specifically, we first aggregate the consensual knowledge from different clients in the embedding space by training a shared representation generator, which produces client-agnostic embeddings. However, solely optimizing the generator might be challenging, given its training representations following different local client feature distributions. Therefore, a Representation Transformation Network (RTNet) is locally trained at each client to transform the client-agnostic synthetic embeddings into client-specific. Hereby, we aim at aligning the client-agnostic embeddings with the local feature distribution. Finally, the local dataset of each client will be augmented by its client-specific synthetic embeddings.

The proposed method FRAug achieves state-of-the-art results on three benchmark datasets with feature distribution shift, surpassing the concurrent FL methods addressing the same problem, including PartialFed [55] and FedBN [43]. Moreover, the superior performance of FRAug on a medical dataset illustrates its applicability in complex real-world FL applications. Our contributions can be summarized as follows:

- We propose a novel representation augmentation algorithm (*FRAug*) to address FL with non-IID features.
- We conduct comprehensive experiments on three public benchmark datasets with feature distribution shifts, in which FRAug achieves SOTA results.
- We verify the maturity and scalability of FRAug on a real-world medical dataset, and further analyze the convergence rate and robustness of FRAug.

2. Related Work

2.1. Federated Learning (FL)

Federated Averaging (FedAvg) [49] is one of the classic FL algorithms for training machine learning models using decentralized data sources. This simple paradigm suffers from performance degradation when there exists data heterogeneity [28, 41]. Numerous studies have been conducted for label space heterogeneity, *i.e.*, class distributions are imbalanced across different clients, by adding additional regularization term in the client local update [42, 8, 53, 35, 26, 4, 31, 65], utilizing shared local data [70, 45, 16], introducing additional public datasets [37, 44, 17], fully or partially personalizing the client models [3, 12, 56, 40, 7, 52, 1], or performing data-free knowledge distillation [46] in the input space [20, 69, 68] or the feature space [21, 76, 47]. However, there are only limited studies addressing the heterogeneity in feature space, *i.e.*, non-IID features. Recently, [2] showed that Batch Normalization layers (BN) [24] with local statistics improve the robustness of the FL model to inter-center data variability and yield better out-of-domain generalization results, while FedBN [43] provided more theoretical analysis on the benefits of local BN layers for FL with feature non-IID. PartialFed [55] empirically found that partially initializing the client models could alleviate the effect of feature distribution shift. HarmoFL [27] focused on FL applications for heterogeneous medical images and applied amplitude normalization in frequency space and model weight perturbation to harmonize the training process. In this work, we tackle the problem of non-IID features in FL via a client-specific data augmentation approach performed in the embedding space. In particular, client-agnostic embeddings are initially synthesized by a shared generator that captures the knowledge from different distributions, which are then personalized by separate client-specific models. Training the local model with the resulting client-specific embeddings improves its robustness against the feature distribution shift.

2.2. Cross-Domain Learning

The problem of learning on centralized data with non-IID features, *i.e.*, cross-domain data, has been widely studied in the context of Unsupervised Domain Adaptation (UDA) [60, 5, 67, 6, 30, 62], where a model is trained using multiple source domains and finetuned using an unlabelled target domain, and Domain Generalization (DG) [71, 13, 72, 14, 36, 29], where the target domain data is not accessible during the training process of UDA. A variety of efforts have been made to tackle the problem of UDA and DG. CROSSGRAD [54] used adversarial gradients obtained from a domain classifier to augment the training data. L2A-OT [73] trained a generative model to transfer the training samples into pseudo-novel domains. MixStyle [74] performed feature-level augmentation by interpolating the style statistics of the output features from different network layers. While the aforementioned methods assume centralized access to all datasets from different domains, we address the problem where the datasets are decentralized and cannot be shared due to privacy concerns.

3. Methodology

3.1. Problem Statement

In this work, we address an FL problem setting with non-IID features, which we describe in the following. Let $\mathcal{X} \subset \mathbb{R}^{d_{in}}$ be an input space, $\mathcal{U} \subset \mathbb{R}^{d_u}$ be a feature space, and $\mathcal{Y} \subset \mathbb{N}$ be an output space. Let $\boldsymbol{\theta} := [\boldsymbol{\theta}_f, \boldsymbol{\theta}_h]$ denote the parameters of the classification model trained in an FL setting involving one central server and $K \in \mathbb{N}$ clients. The model consists of two components: a feature extractor $f : \mathcal{X} \rightarrow \mathcal{U}$ parameterized by $\boldsymbol{\theta}_f$, and a prediction head $h : \mathcal{U} \rightarrow \mathcal{Y}$ parameterized by $\boldsymbol{\theta}_h$. We assume that a dataset $D^k = \{(\boldsymbol{x}_i^k, y_i^k) | i \in \{1, .., N_k\}\}$, containing private data, is available on each client, where $N^k \in \mathbb{N}$ denotes

Method	OfficeHome				
	Art	Clipart	Product	Real	avg
w/o Add. Embeddings	57.47	56.74	73.32	71.25	64.69
w. Add. Embeddings	68.18	72.31	80.04	79.50	**75.01**

Table 1: Evaluation accuracies of models optimized with (w.) and without (w/o) prediction head finetuned using additional embeddings on OfficeHome benchmark, indicating the applicability of the representation generator given the performance increase.

the number of samples in D^k and $C \in \mathbb{N}$ denotes the number of classes. As discussed in [28], FL with non-IID data can be described by the distribution shift on local datasets: $P^{k_1}_{\mathcal{XY}} \neq P^{k_2}_{\mathcal{XY}}$ with $\forall k_1, k_2 \in \{1, ..., K\}, k_1 \neq k_2$, where $P^k_{\mathcal{XY}}$ defines the joint distribution of input space $\mathcal{X}$ and label space $\mathcal{Y}$ on D^k. The addressed problem setting, *i.e.*, FL with non-IID features, covers (1) *covariate shift*: The marginal distribution $P_{\mathcal{X}}$ varies across clients, while $P_{\mathcal{Y}|\mathcal{X}}$ is the same, and (2) *concept shift*: The conditional distribution $P_{\mathcal{X}|\mathcal{Y}}$ varies across clients, while $P_{\mathcal{Y}}$ is the same [43]. From the perspective of cross-domain learning literature [60, 71], local data from every client can be viewed as a separate domain.

3.2. Motivational Case Study

To motivate our representation augmentation algorithm, we present an empirical analysis to address the following research question: *Does finetuning only the prediction head using additional synthetic feature embeddings lead to performance improvement?* First, we optimize a classification model $\boldsymbol{\theta}^k$ with 10% of the local dataset D^k following prior FL work [49, 43]. Then, we fix the feature extractor and finetune *only* the prediction head with 100% of D^k. Finally, we evaluate both classification models. Here, we use the representations, extracted by the feature extractor using the additional real images, to simulate the output produced by a "perfect" embedding generator.

The results in Tab. 1 show that the feature extractor, trained with less data, still captures useful information when exposed to unseen image samples. Most importantly, a substantial average performance boost of 10.32% shows that generating additional representations benefits the client local update, proving the applicability and effectiveness of the proposed method.

3.3. Proposed Method

To tackle FL with non-IID features, we propose Federated Representation Augmentation (FRAug). Our algorithm is built upon FedAvg [49], which is the most widely used FL strategy. In FedAvg, the central server sends a copy of the global model $\boldsymbol{\theta}$ to each client to initialize their local models $\{\boldsymbol{\theta}^k | k \in K\}$. After training on its local dataset

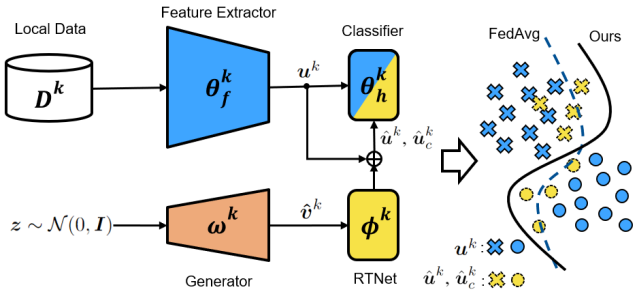


Figure 1: Overview of FRAug local update at client k: a shared generator is learned to aggregate knowledge from multiple clients and generate client-agnostic feature embeddings $\hat{\boldsymbol{v}}^k$, which are then fed into the local Representation Transformation Network (*RTNet*) to produce client-specific feature embeddings $\hat{\boldsymbol{u}}^k$ and $\hat{\boldsymbol{u}}^k_c$. Finally, the real feature embeddings $\boldsymbol{u}^k$, extracted by the feature extractor using local dataset D^k, will be augmented with $\hat{\boldsymbol{u}}^k$ and $\hat{\boldsymbol{u}}^k_c$ in the classification model optimization.

D^k, the client-specific updated models are sent back to the central server, where they are averaged and used as the global model. Such communication rounds are repeated until some predefined convergence criteria are met. Similarly, the training process of FRAug (Algorithm 1) can be divided into two stages: (1) The *Server Update*, where the central server aggregates the parameters uploaded by the clients and distributes the averaged parameters to each client, and (2) the *Client Update*, where each client receives the model parameters from the central server and performs local optimization. Unlike FedAvg, where only the local dataset of each client is used for training, FRAug generates additional feature embeddings to finetune the prediction head of the local classification model. Concretely, we train a shared generator and a local Representation Transformation Network (RTNet) for each client, which together produce *domain-specific* synthetic feature embeddings for each client to augment its local data in the embedding space. Hereby, the shared generator captures knowledge from all the clients to generate client-agnostic embeddings, which are then personalized by the local RTNet into client-specific embeddings. In the following, we provide a more detailed explanation of FRAug.

3.3.1 Server Update

At the beginning of the training, the server initializes the parameters of the classification model $\boldsymbol{\theta} := [\boldsymbol{\theta}_f, \boldsymbol{\theta}_h]$, as well as the *shared* generator $\boldsymbol{\omega}$. In each communication round r, all clients receive the aggregated model parameters and conduct the *Client Update* procedure in parallel. Subsequently, the server securely aggregates the optimized model parameters from all the clients into a single model that is used in the next communication round.

3.3.2 Client Update

As shown in Fig. 1, at the beginning of the first communication round, each client *locally* initializes a Representation Transformation Network (*RTNet*) parameterized by $\boldsymbol{\phi}^k$. Subsequently, each client receives the classification model parameters $\boldsymbol{\theta}^k$ and the generator parameters $\boldsymbol{\omega}^k$ from the server, and conducts T local update steps. Each local update comprises 2 stages: (1) Classification model optimization, and (2) Generator and RTNet optimization.

(1) Classification Model Optimization: In this stage, the generator and the RTNet are fixed, while the classification model is updated by minimizing the loss $\mathcal{L}_{cls}$, where

$$\begin{aligned} &\mathcal{L}_{cls} = \mathcal{L}_{real} + \mathcal{L}_{syn}, \\ &\text{with} \quad \mathcal{L}_{real} = L_{\text{CE}}(h^k(f^k(\boldsymbol{X}^k)), \boldsymbol{y}^k). \end{aligned} \tag{1}$$

While $\mathcal{L}_{real}$ is minimized to update the model parameter $\boldsymbol{\theta}^k$ by using real training samples from D^k, $\mathcal{L}_{syn}$ is minimized to update only the prediction head h^k as it is computed on synthetically generated samples in the embedding space $\mathcal{U}$. We use cross-entropy (L_{CE}) for both loss functions.

To generate domain-specific synthetic embeddings, the shared generator g^k and local RTNet m^k are used to generate residuals that are added to the embeddings of real examples produced by the local feature extractor f^k. Hereby, we first generate client-agnostic embeddings $\hat{\boldsymbol{v}}^k$ by feeding a batch of random vector $\boldsymbol{z}$, sampled from standard Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$, and class labels $\boldsymbol{y}$ into the generator g^k. Subsequently, $\hat{\boldsymbol{v}}^k$ are transformed by the local RTNet into client-specific residuals and added to the embeddings of real datapoints. We distinguish two types of synthetic embeddings that we generate to train the local prediction head: domain-specific synthetic embeddings $\hat{\boldsymbol{u}}^k$ and class-prototypical domain-specific synthetic embeddings $\hat{\boldsymbol{u}}_c^k$ for category c. The domain-specific embeddings $\hat{\boldsymbol{u}}^k$ are generated by adding synthetic residuals to the embeddings $\boldsymbol{u}^k$ of real examples from the current batch sampled from D^k. On the other hand, synthetic residuals are added to class-prototypes $\overline{\boldsymbol{u}}_c^k$, *i.e.*, class-wise average embeddings of real examples, to produce $\hat{\boldsymbol{u}}_c^k$, which stabilizes the training and increase the variance of the generated embeddings.

$$\mathcal{L}_{syn} = L_{\text{CE}}(h^k(\hat{\boldsymbol{u}}^k), \boldsymbol{y}) + \sum_{c \in C} L_{\text{CE}}(h^k(\hat{\boldsymbol{u}}_c^k), c), \tag{2}$$

$$\begin{aligned} \text{with} \quad \hat{\boldsymbol{u}}^k &= \boldsymbol{u}^k + \lambda_{syn} \cdot m^k(g^k(\boldsymbol{z}, \boldsymbol{y})), \\ \hat{\boldsymbol{u}}_c^k &= \overline{\boldsymbol{u}}_c^k + \lambda_{syn} \cdot m^k(g^k(\boldsymbol{z}', c)). \end{aligned} \tag{3}$$

To compute the class-wise average embedding $\overline{\boldsymbol{u}}_c^k$, we use the exponential moving average (EMA) scheme, at each local iteration. In particular,

$$\overline{\boldsymbol{u}}_c^k \leftarrow (1 - \lambda_c) \cdot \overline{\boldsymbol{u}}_c^k + \lambda_c \cdot \frac{\sum_{i \in B} \mathbb{1}(\boldsymbol{y}_i = c) \cdot f(\boldsymbol{x}_i)}{\sum_{i \in B} \mathbb{1}(\boldsymbol{y}_i = c) + \epsilon}, \tag{4}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function, B is the batch size of the real samples, and ϵ is a small number added for numerical stability. By using the average embeddings of previous iterations, we enable the examples of previously sampled batches to contribute to the computation of the current average embeddings. The ratio λ_c follows an exponential ramp-up schedule as proposed in [33].

We note that, in Eq. (3), for the generation of $\hat{\boldsymbol{u}}^k$, the original labels $\boldsymbol{y}$ of the sampled data batch are used for the residual generation, since the residuals are added to the embeddings of the examples corresponding to these labels. For $\hat{\boldsymbol{u}}_c^k$, we feed the label c that corresponds to the class of the average embedding $\overline{\boldsymbol{u}}_c^k$. While the residuals produced by the generator and the RTNet are random in early training iterations due to the random initialization of these models, they become more informative as training progresses. To reflect this in our algorithm, we employ the weighting coefficient λ_{syn} (Eq. (3)) that controls the impact of the residuals, and increase it following an exponential schedule throughout training.

To allow the different client-specific models to learn feature extractors tailored to their data distribution D^k, while still benefiting from the collaborative learning, we use local Batch Normalization layers (BN) [24] as introduced in [43].

(2) Generator and RTNet Optimization: In the second stage, the classification model is fixed while the generator and the RTNet are optimized. The class-conditional generator g^k takes a batch of random vectors $\boldsymbol{z}$ and class labels $\boldsymbol{y}$ to produce *client-agnostic* feature embeddings $\hat{\boldsymbol{v}}^k$. $\hat{\boldsymbol{v}}^k$ are then fed into the RTNet m^k to be adapted to the feature distribution of the corresponding client k. The resulting residuals are added on the embeddings of real examples to produce the *domain-specific* synthetic embeddings $\hat{\boldsymbol{u}}^k$ and $\hat{\boldsymbol{u}}_c^k$. The generator will be optimized by minimizing the loss $\mathcal{L}_{gen}$, with

$$\mathcal{L}_{gen} = L_{\text{CE}}(h^k(\hat{\boldsymbol{v}}^k), \boldsymbol{y}) - \alpha L_{\text{MMD}}(\hat{\boldsymbol{v}}^k, \boldsymbol{u}^k). \tag{5}$$

The minimization of the cross-entropy loss L_{CE} incentivizes the shared generator to produce features that are recognized by the prediction heads of all the clients. By sharing and optimizing the generator across all clients, we ensure that the synthetic embeddings produced by the generator, *i.e.*, $\hat{\boldsymbol{v}}^k$, capture client-agnostic semantic information. Additionally, we maximize the statistical distance [61] between $\hat{\boldsymbol{v}}^k$ and the real feature embeddings $\boldsymbol{u}^k$. By doing so, we force $\hat{\boldsymbol{v}}^k$ not to follow any client-specific distribution, and thus enhance the variance of the augmented feature space. Here, we adopt Maximum Mean Discrepancy (MMD) [18] as the distance metric. Subsequently, the client-agnostic embeddings are fed into the RTNet m^k parametrized by $\boldsymbol{\phi}^k$ to produce domain-specific embeddings $\hat{\boldsymbol{u}}^k$ and $\hat{\boldsymbol{u}}_c^k$. $\boldsymbol{\phi}^k$ is optimized by minimizing the loss $\mathcal{L}_{rt}$, where

Algorithm 1 Training procedure of FRAug

ServerUpdate

Randomly initialize $\boldsymbol{\theta}_0, \boldsymbol{\omega}_0$
for round r = 1 to R **do**
  for client k = 1 to K **do** {**in parallel**}
    $\boldsymbol{\theta}_r^k, \boldsymbol{\omega}_r^k \leftarrow \text{ClientUpdate}(\boldsymbol{\theta}_{r-1}, \boldsymbol{\omega}_{r-1}, k, r)$
  $\boldsymbol{\theta}_r \leftarrow \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\theta}_r^k$
  $\boldsymbol{\omega}_r \leftarrow \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\omega}_r^k$

ClientUpdate$(\boldsymbol{\theta}, \boldsymbol{\omega}, k, r)$

if $r = 1$ **then**
  Randomly initialize $\boldsymbol{\phi}^k$
$\boldsymbol{\theta}^k \leftarrow \boldsymbol{\theta}$, $\boldsymbol{\omega}^k \leftarrow \boldsymbol{\omega}$
for local step t = 1 to T **do**
  Sample $\{\boldsymbol{X}, \boldsymbol{y}\}$ from D_k
  Sample $\boldsymbol{z}, \boldsymbol{z}' \sim \mathcal{N}(0, I)$
  Optimize $\boldsymbol{\theta}^k$ (Eq. (1))
  Optimize $\boldsymbol{\omega}^k$ (Eq. (5)) and $\boldsymbol{\phi}^k$ (Eq. (6))

$$\begin{aligned}\mathcal{L}_{rt} = &- L_{\text{ent}}(h^k(\hat{\boldsymbol{u}}^k)) - \sum_{c\in C} L_{\text{ent}}(h^k(\hat{\boldsymbol{u}}_c^k)) \\ &+ \beta(L_{\text{MMD}}(\hat{\boldsymbol{u}}^k, \boldsymbol{u}^k) + \sum_{c\in C} L_{\text{MMD}}(\hat{\boldsymbol{u}}_c^k, \overline{\boldsymbol{u}}_c^k)).\end{aligned} \tag{6}$$

Here, we maximize the entropy (L_{ent}) of the prediction head output on $\hat{\boldsymbol{u}}^k$, $\hat{\boldsymbol{u}}_c^k$ to incentivize the generation of synthetic embeddings that are *hard* to classify for the prediction head h^k. To avoid generating outliers, we align the synthetic embedding distribution with that of the client local data by minimizing their Maximum Mean Discrepancy (MMD). In particular, we penalize high MMD distances between $\hat{\boldsymbol{u}}^k$ and $\boldsymbol{u}^k$, as well as $\hat{\boldsymbol{u}}_c^k$ and $\overline{\boldsymbol{u}}_c^k$ for each class c. α and β denote weighting coefficients in Eq. (5) and Eq. (6), respectively.

4. Experiments and Analyses

We conduct an extensive empirical analysis to investigate the proposed method and its viability. Firstly, we compare FRAug with several FL baseline methods on 3 popular benchmark datasets involving feature distribution shifts. Subsequently, we validate our approach on a real-world medical dataset for genetic treatment classification. We present additional analysis regarding convergence rate, communication overhead, and robustness to input noise. Finally, we demonstrate the ablation studies of FRAug and its comparison with other augmentation-based FL methods.

4.1. Benchmark Experiments

4.1.1 Datasets Description

We conduct experiments on three common image classification benchmarks with domain shift: (1) *OfficeHome* [59], which contains 65 classes in four domains: Art (A), Clipart (C), Product (P) and Real-World (R). (2) *PACS* [38], which includes images that belong to 7 classes from four domains Art-Painting (A), Cartoon (C), Photo (P), and Sketch (S). (3) *Digits* comprises images of 10 digits from the following four datasets: MNIST (MT) [34], MNIST-M (MM) [15], SVHN (SV) [51], and USPS (UP) [23]. Each client contains data from one of the domains, *i.e.*, there exists feature distribution shifts across different clients. To simulate data scarcity described in previous sections, we assume that only 10% (1% for the Digits dataset) of the original data is available for each client, resulting in ca. 100 to 1000 data samples per client following the experimental setup in the previous work [49, 43].

4.1.2 Baselines

We compare our approach with several baseline methods, including *Single*, *i.e.*, training an individual model on each client separately, *All*, *i.e.*, training a single model at the central server using data aggregated from all clients, *FedAvg* [49], *pFedAvg*, *i.e.*, FedAvg with local model personalization. We also compare FRAug with *FedProx* [42], *FedBABU* [52], and *FedProto* [57], which are strong concurrent methods handling label space heterogeneity in FL. We note that *All* is an oracle baseline as it requires centralizing the data from the different clients, hence infringing the data-privacy requirements. Furthermore, we compare our method with the current state-of-the-art FL methods for non-IID features, *i.e.*, *FedBN* [43] and *PartialFed* [55]. We use the published code of *FedBN* and reimplement *PartialFed* since the original implementation was not made public. We conduct the same hyperparameter search for all methods and report the best results. The detailed hyperparameter search spaces of different methods are provided in Appendix A.

4.1.3 Implementation Details

For the OfficeHome and PACS datasets, we use a ResNet18 [22] pretrained on ImageNet [9] as initialization of the classification model. For Digits, we use a 6-layer Convolution Neural Network (CNN) as the backbone following prior work [43]. We adopt a 2-layer MLP as the generator and RTNet architectures for all datasets. Besides, we apply the same data augmentation techniques on the input images during the classification model training for all clients following the previous work [19]. In Appendix A, we provide further details about model architectures and training hyperparameters. All experiments are repeated with 3 different random seeds.

Benchmark		Single	All	FedAvg	FedProx	FedProto	FedBABU	pFedAvg	PartialFed	FedBN	FRAug
Office Home	A	35.80±0.1	56.65±0.7	57.47±0.6	55.68±0.4	51.44±0.6	49.80±0.4	52.50±0.9	48.83±0.2	57.59±0.8	**57.61**±0.6
	C	45.54±0.8	58.81±1.6	56.74±0.9	56.88±0.5	52.63±0.7	54.23±0.7	52.09±1.1	49.96±0.2	56.52±0.3	**60.03**±0.5
	P	67.04±0.8	71.39±0.3	73.32±0.8	73.84±0.3	70.78±0.7	70.72±0.6	71.78±0.8	72.22±0.8	73.55±1.0	**74.03**±0.8
	R	61.16±0.7	72.63±1.3	71.25±0.3	72.15±0.9	65.13±0.2	66.74±0.5	66.28±0.4	65.82±0.6	72.40±0.9	**74.58**±0.4
	avg	52.42±0.4	64.87±0.9	64.69±0.6	64.63±0.6	60.00±0.3	60.37±0.3	60.67±0.7	59.20±0.5	65.02±0.7	**66.60**±0.3
Digits	MT	96.68±0.2	97.04±0.1	96.85±0.1	96.90±0.1	96.80±0.1	97.38±0.2	96.40±0.2	97.13±0.1	97.03±0.1	**97.81**±0.1
	MM	77.77±0.5	77.04±0.1	73.51±0.2	72.60±0.4	78.16±0.6	79.30±0.8	77.56±0.4	74.21±0.5	77.02±0.2	**81.65**±0.5
	SV	75.55±0.3	77.96±0.5	74.49±0.2	73.01±0.5	77.90±0.2	74.03±0.5	77.50±0.1	78.10±0.5	77.59±0.1	**81.24**±0.3
	UP	79.93±0.8	97.13±0.1	97.62±0.1	97.31±0.3	97.37±0.1	95.37±0.4	96.67±0.1	94.78±0.5	96.80±0.2	**97.67**±0.3
	avg	82.54±0.1	87.29±0.2	85.62±0.2	84.96±0.3	87.50±0.1	86.52±0.4	87.03±0.2	86.05±0.3	87.11±0.2	**89.59**±0.4
PACS	A	82.37±0.6	83.17±0.2	82.72±0.4	80.17±0.4	85.09±0.5	81.25±0.6	88.05±0.8	84.85±0.2	86.60±0.5	**87.34**±0.5
	C	86.08±0.9	86.92±0.8	84.04±1.3	82.04±0.8	86.91±0.3	87.76±1.1	86.20±0.7	87.92±0.5	87.76±1.0	**88.47**±0.9
	P	92.01±1.1	95.95±0.8	96.05±0.5	96.74±1.0	96.49±0.6	94.74±0.4	97.89±0.5	98.24±0.4	97.95±0.4	**98.64**±0.6
	S	87.52±0.8	88.70±0.7	89.50±0.7	88.50±1.0	89.20±0.4	89.41±0.3	88.89±0.9	90.10±0.8	90.75±0.3	**90.95**±0.4
	avg	87.00±0.5	88.68±0.6	88.08±0.9	86.86±0.9	89.42±0.5	88.29±0.6	90.26±0.6	90.28±0.7	90.76±0.3	**91.34**±0.1

Table 2: Evaluation results of different algorithms on three real-world benchmark datasets with feature distribution shift. We report the mean±std accuracy of each client from 3 runs with different seeds. The best results are marked in **bold** (The same applies to the subsequent tables).

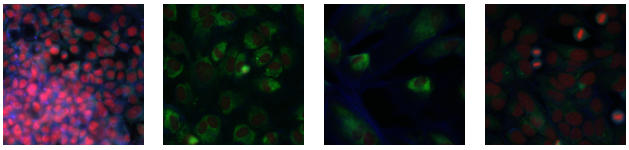
4.1.4 Results and Discussion

We report the accuracies achieved by the different methods on all three datasets in Tab. 2. We observe that FRAug outperforms all the baselines on all benchmark datasets. On OfficeHome, FRAug outperforms FedAvg and FedBN by 1.91% and 1.58%, respectively. On Digits, FRAug achieves a substantial 2.3% improvement on average compared with all the alternative methods. Likewise, FRAug yields the highest average accuracy on PACS. We note that FRAug achieves an average performance increase of 1.6% compared to FedBN across all three datasets, which surpasses the average performance improvement yielded by FedBN on FedAvg, *i.e.*, 1.5%. Moreover, we find that the performance improvement compared to the best baseline is the highest on the most challenging domains, *i.e.*, on which all methods yield lower results than on other domains. These include MNIST-M and SVHN from Digits, as well as Clipart from OfficeHome, where FRAug achieves impressive improvements of above 3%. Interestingly, our approach outperforms the centralized baseline *All*, demonstrating its effectiveness in aggregating the knowledge from different clients to enable a client-specific augmentation.

4.2. Validation on a Real-World Medical Dataset

4.2.1 Experimental Setup

To illustrate the effectiveness of FRAug on real-world applications, we further conduct experiments on the RxRx1 [58] medical dataset, which contains images (Fig. 2) of cells obtained by fluorescent microscopy. The task is to classify which genetic treatment the cells received. There are 4 different cell types adopted in the dataset, *i.e.*, HEPG2 (H), HUVEC (V), RPE (R), and U2OS (U), while multiple batches of experiments are executed for each cell type. Despite the careful control of experimental variables, *e.g.*, temperature and humidity, feature space heterogeneity is observed across different batches of experiments [32]. Therefore, we consider 4 different cell types as 4 different domains. We divide the batches of experiments from each domain, *i.e.*, for each cell type, into 4 groups, where each group has the same number of batches and is assigned to one client. By doing so, we simulate a real-world collaborative training setup of different medical institutions where every institution has conducted some batches of experiments on one specific cell type. We note that the number of domains is not equal to the number of clients. Following the FL setting described in the previous section, we select 50 classes from 1139 classes in the original dataset. We adopt ResNet18 [22] pretrained on ImageNet [9] as initialization of the classification model. To further evaluate the scalability of the proposed method, we conduct experiments where 2, 3, and 4 clients from each domain are selected, which gives in total 8, 12, and 16 clients joining the federated communication, respectively. Note that more clients correspond to larger data quantity.

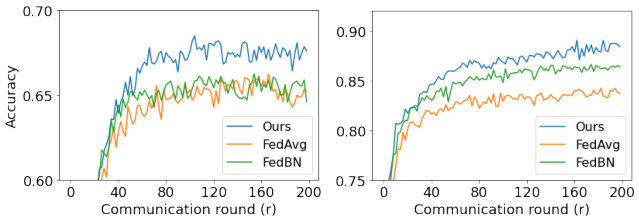


(a) HEPG2 (b) HUVEC (c) RPE (d) U2OS

Figure 2: Example images of different cell types, *i.e.*, local data from different clients, in RxRx1 dataset. Strong feature space heterogeneity can be observed between image appearance. *Best viewed in color.*

Method	8 clients					12 clients					16 clients				
	H	V	P	U	avg	H	V	P	U	avg	H	V	P	U	avg
FedAvg	24.31 ±0.3	34.39 ±0.8	20.19 ±1.3	17.65 ±0.9	24.14 ±0.8	28.84 ±1.3	40.60 ±0.9	19.72 ±0.7	16.67 ±0.8	26.46 ±0.8	28.17 ±0.7	41.60 ±1.0	23.55 ±0.8	17.65 ±0.8	27.74 ±0.6
HarmoFL	19.61 ±1.0	44.02 ±0.5	20.18 ±0.2	**22.53** ±0.9	26.58 ±1.0	26.61 ±0.8	**49.15** ±0.5	19.27 ±0.7	17.97 ±0.9	28.25 ±0.8	28.57 ±0.9	47.29 ±0.7	22.02 ±0.5	18.05 ±0.7	28.98 ±0.4
FedBN	22.94 ±0.9	43.70 ±0.5	25.92 ±1.0	18.63 ±0.9	27.80 ±1.0	27.22 ±0.4	46.01 ±0.4	26.85 ±0.8	16.95 ±1.1	29.26 ±0.6	29.35 ±0.6	**49.08** ±0.8	29.58 ±0.3	19.97 ±0.2	31.99 ±0.3
FRAug	**28.28** ±0.3	**45.33** ±0.9	**28.74** ±1.2	21.04 ±0.5	**30.84** ±0.5	**30.73** ±0.9	47.36 ±0.8	**30.58** ±0.2	**19.60** ±0.7	**32.07** ±0.5	**32.34** ±0.4	48.05 ±0.5	**31.83** ±1.0	**20.59** ±0.7	**33.20** ±0.8

Table 3: Evaluation results of different methods on real-world medical dataset RxRx1. We conduct experiments with different number of clients for each cell type and report average accuracy of clients holding the same cell type.



(a) OfficeHome (b) Digits

Figure 3: Convergence analysis of FedAvg, FedBN and FRAug on (a) OfficeHome and (b) Digits benchmarks.

4.2.2 Results and Discussion

In Tab. 3, we compare FRAug with FedAvg, FedBN, and HarmoFL [27], which is a concurrent work that proposed a strong FL method tailored for heterogeneous medical images, and report the average validation accuracy of clients owning data from the same domain (cell type). We observe that FRAug outperforms all competitors over all settings with different numbers of clients and different data amounts. We highlight the performance improvements achieved by FRAug compared with the baselines, *i.e.*, when 8, 12, and 16 clients join the federated collaborative training, our approach surpasses the other methods by at least 3.04%, 2.81%, and 1.21%, respectively. These results indicate the effectiveness of FRAug on settings with larger quantities of training data as well as its scalability to the complex real-world FL scenarios with more clients.

4.3. Additional Analyses

4.3.1 Convergence Analysis

In Fig. 3, we display the convergence analysis of the proposed method compared with the baseline FedAvg and FedBN on the OfficeHome and Digits benchmarks. Hereby, we report the average classification accuracy of all clients on their corresponding local testing set after conducting the communication round r. As shown in the figure, even though FRAug utilizes the representation augmentation technique, the learning curves of FRAug still exhibit better convergence rates. It's also worth noticing that FRAug already achieves distinct performance gain after 50 communication rounds, *i.e.*, 25% of the total rounds.

Model	Parameters(M)	MACs(G)
ResNet18	11.18	1.84
CNN for Digits	18.15	0.08
Generator	0.39	$\ll 0.01$
RTNet	0.26	$\ll 0.01$

Table 4: Parameters number and MACs (Multiply Accumulate operations) comparison of different components in FRAug.

4.3.2 Analysis of Communication Overhead

In Tab. 4, we demonstrate the number of model parameters and computational costs, *i.e.*, the number of operations, of different components used in the proposed method. We observe that both generator and RTNet take only 2-3% of the parameter numbers used in the classification model, proving the communication overhead between client and server is negligible. Besides, we notice that only less than 1% of operations are needed for the newly introduced components in FRAug compared with the classification model. Therefore, we conclude that FRAug is communication efficient and does not impose significant impacts on the clients local training, showing its applicability to clients with edge devices and limited computing power.

4.3.3 Ablation Study

To illustrate the importance of different FRAug components, we conduct an ablation study on three benchmark datasets. The results are shown in Tab. 5. We first notice that applying only the client-specific RTNet solely based on local data is ineffective: Its output $\hat{\boldsymbol{u}}^k$ is restricted in the client local distribution when the client-agnostic feature embeddings are inaccessible, which proves the criticality of optimizing a shared generator G. We further observe that using the client-agnostic synthetic embeddings $\hat{\boldsymbol{v}}^k$ instead of

G ($\hat{\boldsymbol{v}}$)	RTNet ($\hat{\boldsymbol{u}}$)	EMA ($\hat{\boldsymbol{u}}_c$)	OfficeHome	PACS	Digits
	✓		64.58±0.5	88.38±0.5	86.23±0.2
	✓	✓	65.08±0.4	88.50±0.2	86.60±0.1
✓			65.47±0.8	90.82±0.5	87.25±0.1
✓		✓	66.09±0.2	90.74±0.4	88.24±0.3
✓	✓		65.99±0.3	**91.35**±0.1	89.51±0.1
✓	✓	✓	**66.60**±0.4	91.05±0.3	**89.59**±0.2

Table 5: Ablation study for different components of FRAug on three benchmark datasets. The average evaluation accuracy of all clients are reported

the personalized versions leads to slight performance gain. This highlights the importance of the transformation by RTNets into personalized client-specific embeddings. Moreover, the results reveal that both types of synthetic embeddings, *i.e.*, $\hat{\boldsymbol{u}}_c^k$ and $\hat{\boldsymbol{u}}^k$, yield a performance boost when used separately. Employing them together further improves the results, which demonstrates their complementarity.

Additionally, we evaluate the proposed algorithm optimized with different combinations of hyperparameters. From the results, we observe low sensitivity of FRAug to the hyperparameter selection, highlighting its applicability on novel benchmark datasets without time-consuming fine-grained hyperparameter searches. Besides, we conduct experiments with varying numbers of datapoints available on each client. The superior performance of FRAug further indicates its robustness under both data-scarce and data-sufficient scenarios in FL. The detailed evaluation results are provided in Appendix B.

4.3.4 Robustness to Input Noise

Prior works [25, 63, 64] focus on generating or adversarially augmenting the clients local training data. On the contrary, the representation generators used in FRAug extract knowledge from the output of the existing feature extractor, i.e., they do not access the input images. More importantly, FRAug does not impose any constraints on the client local update and model aggregation, which indicates its compatibility with the defensive strategies introduced in [11, 48].

Noise Intensity	Weak	Medium	Strong
FedAvg	63.02±0.4	60.71±0.6	31.26±1.2
FedBN	63.97±0.6	60.12±0.5	30.90±0.9
FRAug	**64.72**±1.0	**61.45**±0.8	**31.65**±0.7

Table 6: Evaluation results of different methods on privatized OfficeHome with different noise intensity. The average accuracy of all clients are reported.

To exhibit the effectiveness of FRAug under the settings with noisy input, we add random noise $\delta \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ to the client local images when optimizing the classification model. More specifically, we select three noise intensities from weak ($\sigma = 0.01$), medium ($\sigma = 0.1$), to strong ($\sigma = 1.0$). The results in Tab. 6 indicate the effectiveness of FRAug under noisy client local data.

Method	A	C	P	R	avg
FedAvg	57.47±0.6	56.74±0.9	73.32±0.8	71.25±0.3	64.69±0.6
$\mathcal{U}(-\gamma, \gamma)$	56.79±0.2	57.47±0.8	72.07±0.2	73.51±0.2	64.96±0.3
$Lap(0, \gamma)$	56.52±0.4	56.37±0.2	72.29±0.2	73.83±0.9	64.75±0.4
$\mathcal{N}(0, \gamma)$	56.93±0.9	57.63±0.5	72.43±0.2	73.27±0.5	65.06±0.4
FAug	50.18±0.5	53.48±0.9	71.82±0.4	66.08±0.8	60.39±0.7
FedReg	53.50±0.3	56.52±0.4	69.36±0.7	68.57±0.2	62.00±0.4
FRAug	**57.61**±0.6	**60.03**±0.5	**74.03**±0.8	**74.58**±0.4	**66.60**±0.3

Table 7: Evaluation results of different augmentation methods on OfficeHome benchmark.

4.3.5 Comparison with Other Augmentation Methods

Since the proposed method applies augmentation in the representation space, we compare FRAug with other augmentation approaches using random noise $\Delta\boldsymbol{u}$ following different distributions. Specifically, we train the prediction head h with real feature embeddings $\boldsymbol{u}$ as well as their augmented variants $\boldsymbol{u} + \Delta\boldsymbol{u}$. We adopt three common distributions for sampling the values of $\Delta\boldsymbol{u}$: Uniform distribution $\mathcal{U}$, Laplace distribution Lap and Gaussian distribution $\mathcal{N}$. We define the standard deviation γ of each distribution as a hyperparameter and report the best results. Moreover, we compare our method with concurrent works applying data augmentation, *i.e.*, *FAug* [25] and *FedReg* [63].

In Tab. 7, we display the evaluation results of representation augmentation approaches with random noise, as well as the concurrent works, on the OfficeHome benchmark. We notice a distinct performance gap between these methods and FRAug, which further highlights the effectiveness of the proposed method.

5. Conclusion

In this work, we present a novel approach to tackle the under-explored feature non-IID problem in FL. The proposed Federated Representation Augmentation (FRAug) method performs client-personalized augmentation in the embedding space to improve the training robustness against feature distribution shift. For that, we optimize a shared generative model to synthesize embeddings by exploiting knowledge from all clients. The output client-agnostic embeddings are then transformed into client-specific embeddings by local Representation Transformation Networks (RTNets). FRAug achieves state-of-the-art results on three benchmark datasets involving feature distribution. Moreover, the superb results of FRAug on a medical dataset illustrate its effectiveness and scalability on complex real-world FL applications.

References

[1] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *arXiv preprint arXiv:2212.01548*, 2022.

[2] 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, 2020.

[3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[4] Chen Chen, Yuchen Liu, Xingjun Ma, and Lingjuan Lyu. Calfat: Calibrated federated adversarial training with label skewness. *arXiv preprint arXiv:2205.14926*, 2022.

[5] Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Evidential neighborhood contrastive learning for universal domain adaptation. 2022.

[6] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. Denoised maximum classifier discrepancy for source-free unsupervised domain adaptation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, volume 2, 2022.

[7] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

[8] Yatin Dandi, Luis Barba, and Martin Jaggi. Implicit gradient alignment in distributed and federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6454–6462, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] David Enthoven and Zaid Al-Ars. An overview of federated deep learning privacy attacks and defensive strategies. *Federated Learning Systems: Towards Next-Generation AI*, pages 173–196, 2021.

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[13] Ahmed Frikha, Haokun Chen, Denis Krompaß, Thomas Runkler, and Volker Tresp. Towards data-free domain generalization. *arXiv preprint arXiv:2110.04545*, 2021.

[14] Ahmed Frikha, Denis Krompaß, and Volker Tresp. Columbus: Automated discovery of new multi-level features for domain generalization via knowledge corruption. *arXiv preprint arXiv:2109.04320*, 2021.

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

[16] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Ensemble attention distillation for privacy-preserving federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15076–15086, 2021.

[17] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. page 3, 2022.

[18] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[19] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[20] Weituo Hao, Mostafa El-Khamy, Jungwon Lee, Jianyi Zhang, Kevin J Liang, Changyou Chen, and Lawrence Carin Duke. Towards fair federated learning with zero-shot data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3310–3319, 2021.

[21] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.

[22] 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.

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

[24] 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.

[25] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[26] Wonyong Jeong and Sung Ju Hwang. Factorized-fl: Personalized federated learning with parameter factorization & similarity matching. In *Advances in Neural Information Processing Systems*, 2022.

[27] 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*, volume 36, pages 1087–1095, 2022.

[28] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cum-

mings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[29] Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7130–7140, 2022.

[30] Payam Karisani. Multiple-source domain adaptation via coordinated domain encoders and paired classifiers. *arXiv preprint arXiv:2201.11870*, 2022.

[31] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *International Conference on Machine Learning*, pages 11058–11073. PMLR, 2022.

[32] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[33] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[34] 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.

[35] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021.

[36] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7399–7407, 2022.

[37] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[38] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[39] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[40] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021.

[46] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[47] 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.

[48] Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and S Yu Philip. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022.

[49] 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.

[50] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

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

[52] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

[53] Yichen Ruan and Carlee Joe-Wong. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8124–8131, 2022.

[54] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[55] Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34, 2021.

[56] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

[57] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence*, volume 1, page 3, 2022.

[58] J. Taylor, B. Earnshaw, B. Mabey, M. Victors, and J. Yosinski. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations (ICLR)*, 2019.

[59] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[60] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

[61] William K Wootters. Statistical distance and hilbert space. *Physical Review D*, 23(2):357, 1981.

[62] Renchunzi Xie, Hongxin Wei, Lei Feng, and Bo An. Gearnet: Stepwise dual learning for weakly supervised domain adaptation. *arXiv preprint arXiv:2201.06001*, 2022.

[63] Chencheng Xu, Zhiwei Hong, Minlie Huang, and Tao Jiang. Acceleration of federated learning with alleviated forgetting in local training. *arXiv preprint arXiv:2203.02645*, 2022.

[64] 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.

[65] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Fed2: Feature-aligned federated learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2066–2074, 2021.

[66] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[67] Luxin Zhang, Pascal Germain, Yacine Kessaci, and Christophe Biernacki. Interpretable domain adaptation for hidden subdomain alignment in the context of pre-trained source models. In *36th AAAI Conférence on Artificial Intelligence*, 2022.

[68] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. *arXiv preprint arXiv:2203.09249*, 2022.

[69] Lan Zhang and Xiaoyong Yuan. Fedzkt: Zero-shot knowledge transfer towards heterogeneous on-device models in federated learning. *arXiv preprint arXiv:2109.03775*, 2021.

[70] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[71] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv e-prints*, pages arXiv–2103, 2021.

[72] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[73] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pages 561–578. Springer, 2020.

[74] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

[75] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[76] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.