Federated Feature Augmentation and Alignment

Tianfei Zhou, Ye Yuan, Binglu Wang, Ender Konukoglu

Abstract—Federated learning is a distributed paradigm that allows multiple parties to collaboratively train deep learning models without direct exchange of raw data. Nevertheless, the inherent non-independent and identically distributed (non-i.i.d.) nature of data distribution among clients results in significant degradation of the acquired model. The primary goal of this study is to develop a robust federated learning algorithm to address *feature shift* in clients' samples, potentially arising from a range of factors such as acquisition discrepancies in medical imaging. To reach this goal, we first propose federated feature augmentation (FEDFA^l), a novel feature augmentation technique tailored for federated learning. FEDFA^l is based on a crucial insight that each client's data distribution can be characterized by first-/second-order statistics (*a.k.a.*, mean and standard deviation) of latent features; and it is feasible to manipulate these local statistics *globally*, *i.e.*, based on information in the entire federation, to let clients have a better sense of the global distribution across clients. Grounded on this insight, we propose to augment each local feature statistic based on a normal distribution, wherein the mean corresponds to the original statistic, and the variance defines the augmentation scope. Central to FEDFA^l is the determination of a meaningful Gaussian variance, which is accomplished by taking into account not only biased data of each individual client, but also underlying feature statistics represented by all participating clients. Beyond consideration of *low-order* statistics in FEDFA^l, we propose a federated feature alignment component (FEDFA^h) that exploits *higher-order* feature statistics to gain a more detailed understanding of local feature distribution and enables explicit alignment of augmented features in different clients to promote more consistent feature learning. Combining FEDFA^l and FEDFA^h yields our full approach **FEDFA**+. FEDFA+ is non-parametric, incurs negligible additional communication costs, and can be seamlessly incorporated into popular CNN and Transformer architectures. We offer rigorous theoretical analysis as well as extensive empirical justifications to demonstrate the effectiveness of the algorithm. Our implementation is publicly available at https://github.com/tfzhou/FedFA.

Index Terms—Federated Learning, Feature Augmentation, Feature Alignment, Feature Statistics

1 INTRODUCTION

FEDERATED learning (FL) [2] is an emerging collaborative training framework facilitating the learning process on decentralized data obtained from a host of clients like mobile phones, wearable devices, or alternative local information sources [3], [4]. FL algorithms, *e.g.*, the *de facto* FEDAVG [5], iterate between training local models on each data source and distilling them into a global federated model, all without explicitly combining data from different sources. Accordingly, it has garnered great interest in various applications, *e.g.*, healthcare [6], [7], finance [8], [9], transportation [10], [11]. Despite its advantages of data privacy, FL encounters a primary challenge that affects its performance and convergence, *i.e.*, distribution shift, which implies that data of each client adheres to a distinct distribution. In this study, we concentrate on the *feature-level shift*, which is prevalent in numerous real-world scenarios, like medical data procured from devices supplied by different manufacturers or natural image collected in diverse environments.

While the challenge of feature-level shift has been extensively studied in centralized learning scenarios, such as domain generalization [12], [13], its study in the context of FL is still relatively unexplored. Some early studies are [14]–[23]. FEDROBUST [15] and FEDBN [14] address the problem through client-dependent learning, wherein the parameters of affine distributions [15] or Batch Normalization layers [14] are exclusively learned for each client using respective local data. However, client-dependent learning raises the risk of algorithmic degradation in the presence of severe dataset bias in local clients, and also hinders the global model to effectively generalize to unseen test clients with distinct data distributions. Other works [16]–[18], [24]–[26] endeavor to learn robust models by adopting Sharpness Aware Minimization (SAM) [27] as the local optimizer, which, however, entails twice the computational cost compared to Stochastic Gradient Descent (SGD) or Adam. Prior research has also looked at sharing information (*e.g.*, style [21], [22], averaged training data [28]) between clients to alleviate the potential domain shift. Inspired by prototype learning theory [29], [30], some methods [23], [31], [32] leverage semantic prototypes to capture domain cues and further constrain the training process. Meanwhile, another stream of works enhances FL generalization by explicitly imposing regularizers to local representations [19] or reducing the variance of generalization gaps among local clients [20].

In this paper, we take a unique route to tackle the problem by exploiting statistical information of latent features. Our approach, *i.e.*, FEDFA+, includes two core components: (i) federated feature augmentation (FEDFA^l) based on a probabilistic modeling of *low-order* feature statistics, and (ii) federated feature alignment (FEDFA^h) based on explicit alignment of *high-order* feature statistics across clients.

First, our main insight is that first-/second-order feature statistics can encapsulate essential domain-aware characteristics, as confirmed in prior research [33]–[35], and as such,

- *T. Zhou is with Beijing Institute of Technology, China, and also with Computer Vision Lab (CVL), ETH Zurich, Switzerland. (Email: ztfei.debug@gmail.com)*
- *Y. Yuan and B. Wang are with Beijing Institute of Technology, China. (Email: yuan-ye@bit.edu.cn)*
- *E. Konukoglu is with Computer Vision Lab, ETH Zurich, Switzerland.*

- *This work builds upon our conference paper [1].*

- *Corresponding author: Ye Yuan*

can be treated as "features of participating client". Accordingly, we argue that the problem of feature shift in FL can be interpreted as a shift in these low-order feature statistics regardless of the source of the shift. This understanding motivates us to develop federated feature augmentation (FEDFA^l), that uses universal statistics characterized by all participants in the federation to correct local statistics.

FEDFA^l embodies this idea by online augmenting feature statistics of each sample during local model training, promoting the robustness of local models to certain changes of "features of participating client". Concretely, we model the augmentation procedure via a multivariate Gaussian distribution. The Gaussian *mean* is set to the original statistic, while the *variance* is determined as to cover potential distribution shifts. In this manner, novel statistics can be effortlessly synthesized by drawing samples from the Gaussian distribution. For effective augmentation, we determine a reasonable variance based not only on variances of feature statistics within each client, but also a global variance by aggregating information from participating clients. The augmentation allows each local model to explore more broadly in the feature space, and potentially to be trained with 'novel samples' that would appear in other clients, thereby facilitating the mitigation of local distribution shift.

Second, although FEDFA^l is distinguished by its simplicity and principled formulation, it relies solely on low-order statistics, which limits its ability to fully capture the intricacies of high-dimensional feature representations. Additionally, it only focuses on promoting exploration within the feature space, without explicitly addressing domain alignment. To resolve these issues, we augment FEDFA^l with FEDFA^h, yielding the final solution **FEDFA+**. The key idea of FEDFA^h is to estimate a global, high-order feature statistic encompassing all clients, which serves as a target distribution that each client is required to align with. For the computation of high-order feature statistics, we opt to softly-binned histograms [36], [37] to approximate the marginal feature distribution in each client. These histograms are uploaded along with model weights to the server for high-order statistic aggregation. During local model training, the alignment process involves minimization of the Kullback-Leibler (KL) divergence between the aggregated statistic and each client's high-order statistic.

FEDFA+ is a conceptually simple yet remarkably effective approach. It is non-parametric, necessitates minimal additional memory and communication costs, and can be effortlessly integrated into arbitrary CNN architectures. To recapitulate, our primary contributions are as follows:

- **FEDFA+ Algorithm**. We address heterogeneous FL based on the exploration of feature statistics. First, we introduce FEDFA^l to reduce the impact of local dataset bias. It is based on a principled probabilistic modeling of low-order feature statistics that facilitates more broadly exploration of the feature space. Second, we present FEDFA^h that enforces the consistency of augmented features across clients by explicitly aligning their high-order feature statistics.
- **Theoretical Analysis.** We provide rigorous theoretical analysis for FEDFA^l, showing that it implicitly introduces regularization to local model learning by regularizing the gradients of latent representations, weighted by variances of feature statistics derived from the entire federation.
- **Empirical Analysis.** Through extensive experiments on five benchmarks, we show the benefits of FEDFA+ in: (i) robustness to different types of heterogeneity, (ii) generalization to new (unseen) test clients, (iii) handling extremely small local dataset and thousands of clients.

This work builds upon our conference paper [1]. The extension is reflected in various aspects. ***First***, technically, we extend FEDFA^l to FEDFA+ in §3 by incorporating a new component FEDFA^h, which aims to align more representative, higher-order feature statistics to further reduce the discrepancy among clients. FEDFA^h conceptually complements to FEDFA^l, and FEDFA+ demonstrates consistent and notable performance improvements over FEDFA^l on a set of benchmarks. ***Second***, empirically, we conduct extensive experiments in terms of new datasets (*i.e.*, FEMNIST [38], [39], Tiny-ImageNet [40]), advanced Transformer backbone (*i.e.*, Swin Transformer), and extend ablative experiments to examine the essential components of our algorithms §5. ***Third***, we delve into the complexity and privacy concerns of the algorithms, and offer more in-depth discussions in §6. ***Last but not least***, we offer more thorough discussions on the motivations, formulations, theoretical analysis, and implementation details throughout the article.

2 Related Work

2.1 Federated Learning

Recent years have witnessed remarkable advancements in FL [2], which has paved the way for privacy-preserving deep learning [41], *i.e.*, train a global model on distributed datasets without exposing private data information. A notable milestone is FEDAVG [5], which involves training local models independently across multiple clients and periodically aggregating the resulting model updates via a central server. However, FEDAVG is designed for i.i.d. data, and may suffer from performance degradation or even divergence when faced with non-i.i.d. client data. Numerous efforts have been devoted to learn in heterogeneous federated environments by, *e.g.*, adding a dynamic regularizer to local objectives in FEDPROX [42] and FEDDYN [43], regularizing network features in FEDSR [19], correcting local drift by variance reduction in SCAFFOLD [44], FEDDC [45] and FEDPVR [46], global adaptive optimization in FEDOPT [47] and local adaptive optimization in FEDLADA [48], local batch normalization in FEDBN [14], model ensemble in FEDBE [49] and FEDMA [50], training with perturbed losses in [15], [18], [24]–[26], or synthesizing out-of-distribution samples in [21], [51]. These methods seek to improve statistical accuracy and convergence under non-i.i.d. conditions, demonstrating the ongoing evolution of FL to better accommodate real-world data distributions and maintain privacy.

A method that closely relates to FEDFA^l is FEDMIX [28] that tackles FL through data augmentation. It adapts the well-known MIXUP algorithm [52] from centralized learning to FL. Nevertheless, FEDMIX requires the exchange of local data (or averaged version) across clients for data interpolation, which raises privacy concerns. In addition, FEDMIX operates at the input level, while FEDFA^l focuses on feature-level augmentation. Since deeper representations tend to better disentangle the underlying factors of variation [53], traversing along latent space could potentially make our

method encounter more realistic samples. This is evidenced by the consistent performance improvements achieved by FEDFA^l over FEDMIX across various scenarios.

2.2 Data Augmentation

Data augmentation has a long and rich history in machine learning. Nowadays it has been an essential regularizer for high-performing DNNs in diverse domains including image classification [54]–[56], object detection [57], and image segmentation [58]–[62]. Early studies [63], [64] focus on *label-preserving* transformations to employ regularization through data, alleviating overfitting and enhancing generalization. For image data, techniques like random horizontal flipping and cropping are commonly used in training advanced neural networks [55]. Recently, there has been a growing trend for *label-perturbing* augmentation, exemplified by methods like MIXUP [52], CUTMIX [65], MIXSTYLE [34] and the very recent approach SOFTAUG [66]. Distinct from these input-level augmentation techniques are feature augmentation methods [34], [35], [67]–[69] that make augmentation in latent feature space. These various data augmentation techniques have shown great successes to learn domain-invariant models in the centralized setup.

FEDFA^l is an instance of label-preserving feature augmentation, tailored for federated learning. It draws inspiration from recent advancements in implicit feature augmentation [34], [35], which focuses on synthesizing samples of new domains by manipulating instance-level feature statistics. These methods consider feature statistics as informative representations that encode domain-specific information. In FEDFA^l, we introduce Gaussian modeling of low-order feature statistics in the FL setting, which facilitates more meaningful feature augmentation within individual clients, eventually leading to better generalization.

2.3 Federated Domain Generalization

Since the seminal study in [70], domain generalization has been actively studied to alleviate dataset bias [71] so as to obtain centralized, domain-agnostic models. Two common strategies have emerged in this context: (i) domain-invariant representation learning [34], [68]–[70], [72]–[76], which involves learning a representation that remains unchanged in terms of its marginal distribution and/or class-conditional distribution across domains, and (ii) meta-learning [77]–[80], which is based on the rationale that a model exhibiting effective adaptation among source domains is more likely to adapt well to unseen target domains. These methods have served as sources of inspiration for tackling DG in the context of FL [19]–[23], [81]–[83].

Our approach aligns with the first paradigm and improves FL generalization through two fundamental techniques: data (feature) augmentation and domain alignment. In terms of data augmentation, FEDFA+ shares similarities with a concurrent work [21] that also employs low-order statistics for augmentation in FL. However, FEDFA+ distinguishes itself by utilizing a probabilistic formulation and demonstrating higher communication efficiency. In terms of domain alignment, we aim for aligning high-order feature statistics of different clients, rather than raw features centroids (*i.e.*, prototypes) in existing studies [23], [31], [32].

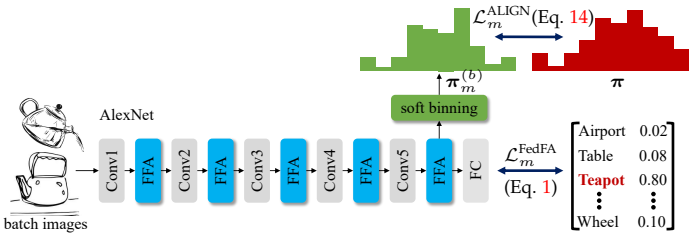


Fig. 1. **Local model training in client m.** Our approach includes two components: (i) FEDFA^l (§3.2) based on the FFA layer and (ii) FEDFA^h (§3.3) based on soft binning. Notably, FFA is inserted after each convolutional stage and activated with a probability p; it is removed during inference, thus making no changes to the network.

More specifically, all these works [23], [31], [32] align class-wise prototypes, each correpsonding to the *mean* feature of all samples from a particular class. In contrast, our approach is class-agnostic. The marginal feature distribution is a global representation of high-order feature statistics of all samples, regardless of their classes. Compared to aligning class-wise, low-order statistics, our class-agnostic, high-order statistic alignment owns a better privacy guarantee. Beyond this, in our approach, the communication cost of the alignment is agnostic to the number of classes, leading to higher scalability than the existing methods [23], [31], [32].

3 OUR APPROACH

In this section, we first introduce the background of federated learning along with the formal definition (§3.1). Then, we present the two essential components in FEDFA+, including FEDFA^l (§3.2) that performs feature augmentation with low-order feature statistics, as well as FEDFA^h (§3.3) that establishes feature alignment across clients based on high-order feature statistics. Fig. 1 illustrates the process of local model training in client m.

3.1 Preliminary: Federated Learning

We assume a standard FL setup with a server that can transmit and receive messages from M client devices. Each client $m \in [M]$ has access to N_m training instances $\{(x_i, y_i)\}_{i=1}^{N_m}$ in the form of image $x_i \in \mathcal{X}$ and corresponding labels $y_i \in \mathcal{Y}$ that are drawn i.i.d. from a device-indexed joint distribution, *i.e.*, $(x_i, y_i) \sim \mathbb{P}_m(x, y)$. The goal of standard FL is to train a deep neural network: $f(\boldsymbol{w}_g, \boldsymbol{w}_h) \triangleq g(\boldsymbol{w}_g) \circ h(\boldsymbol{w}_h)$, where $h : \mathcal{X} \to \mathcal{Z}$ is a feature extractor consisting of K convolutional stages: $h = h^K \circ h^{K-1} \circ \cdots \circ h^1$, and $g : \mathcal{Z} \to \mathcal{Y}$ is a classifier. To learn network parameters $\boldsymbol{w} = \{\boldsymbol{w}_g, \boldsymbol{w}_h\}$, the empirical risk minimization (ERM) is generally used:

$$\min_{\boldsymbol{w}} \mathcal{L}^{\text{ERM}}(\boldsymbol{w}) \triangleq \frac{1}{M} \sum_{m \in [M]} \mathcal{L}_m^{\text{ERM}}(\boldsymbol{w}), \quad \text{where} \quad \mathcal{L}_m^{\text{ERM}}(\boldsymbol{w}) = \mathbb{E}_{(x_i, y_i) \sim \mathbb{P}_m}[\ell_i(g \circ h(x_i), y_i; \boldsymbol{w})]. \tag{1}$$

Here the global objective $\mathcal{L}^{\text{ERM}}$ is decomposable as a sum of client-level empirical losses, *i.e.*, $\{\mathcal{L}_m^{\text{ERM}}\}_m$. Each $\mathcal{L}_m^{\text{ERM}}$ is computed based on a per-data loss function ℓ_i. Due to the separation of clients' data, $\mathcal{L}^{\text{ERM}}$ cannot be solved directly.

FEDAVG [5] is a leading algorithm to solve this optimization problem. It starts with local training of a partial

of clients in parallel, with each client optimizing $\mathcal{L}_m^{\text{ERM}}$ independently. Afterwards, the algorithm performs model aggregation to average all client models into an updated global model, which will then be broadcast to a new set of clients for the next round of local training. Here the client training objective in FEDAVG is equivalent to empirically approximating the local distribution $\mathbb{P}_m$ using a finite N_m number of examples, *i.e.*, $\mathbb{P}_m^e(x,y) = {}^1/_{N_m} \sum_{i=1}^{N_m} \delta(x = x_i, y = y_i)$, where $\delta(x{=}x_i, y{=}y_i)$ is a Dirac mass centered at (x_i, y_i).

3.2 Federated Feature Augmentation (FEDFA^l)

3.2.1 ERM vs. VRM

The ERM-based formulation (*c.f.* Eq. 1) has achieved great success, but the solution strongly depends on how well each approximated local distribution $\mathbb{P}_m^e$ mimics the underlying universal distribution $\mathbb{P}$. In real-world FL setup however, in all but trivial cases each $\mathbb{P}_m^e$ exhibits a unique distribution shift from $\mathbb{P}$, which causes not only inconsistency between local and global empirical losses [43], [84], but also generalization issues [85]. In this work, we circumvent this issue by fitting each local dataset a *richer* distribution (instead of the delta distribution) in the *vicinal region* of each sample (x_i, y_i) so as to estimate a more informed risk. This is precisely the principle behind vicinal risk minimization (VRM) [86]. Particularly, for data point (x_i, y_i), a vicinity distribution $\mathbb{V}_m(\hat{x}_i, \hat{y}_i | x_i, y_i)$ is defined, from which novel virtual samples can be generated to enlarge the support of the local data distribution. This leads to an improved approximation of $\mathbb{P}_m$ as $\mathbb{P}_m^v = {}^1/_{N_m} \sum_{i=1}^{N_m} \mathbb{V}_m(\hat{x}_i, \hat{y}_i | x_i, y_i)$. In centralized learning scenarios, various successful instances of $\mathbb{V}_m$, *e.g.*, MIXUP [52], CUTMIX [65], have been developed. However, simply applying them to local clients, though allowing for performance improvements (see Table 10), is sub-optimal since, without injecting any global information, $\mathbb{P}_m^v$ only provides a better approximation to the local distribution $\mathbb{P}_m$, rather than the true distribution $\mathbb{P}$. We introduce FEDFA^l to estimate a more reasonable $\mathbb{V}_m$ in FL.

3.2.2 Probabilistic First-/Second-Order Statistic Modeling

FEDFA^l belongs to the family of label-preserving feature augmentation [87]. During training, it estimates a vicinity distribution $\mathbb{V}_m^k$ at each convolutional layer h^k for client m, which is subsequently used to augment corresponding latent features. Denote $\boldsymbol{X}_m^k \in \mathbb{R}^{B\times C\times H\times W}$ as the intermediate feature representation of B mini-batch images, with spatial size $(H \times W)$ and channel number (C), and Y_m^k as the corresponding label. $\mathbb{V}_m^k$ is label-preserving in the sense that $\mathbb{V}_m^k(\hat{\boldsymbol{X}}_m^k, \hat{Y}_m^k | \boldsymbol{X}_m^k, Y_m^k) \triangleq \mathbb{V}_m^k(\hat{\boldsymbol{X}}_m^k | \boldsymbol{X}_m^k)\delta(\hat{Y}_m^k = Y_m^k)$, *i.e.*, it only transforms the latent feature $\boldsymbol{X}_m^k$ to $\hat{\boldsymbol{X}}_m^k$, but the label Y_m^k is kept unchanged.

Instead of explicitly modeling $\mathbb{V}_m^k(\hat{\boldsymbol{X}}_m^k | \boldsymbol{X}_m^k)$, our method performs implicit feature augmentation by manipulating channel-wise feature statistics. Specifically, for $\boldsymbol{X}_m^k$, its channel-wise, first-/second-order statistics, *i.e.*, mean μ_m^k and standard deviation σ_m^k, are computed as follows:

$$\begin{aligned} \mu_m^k &= \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} \boldsymbol{X}_m^{k,(h,w)} \quad \in \mathbb{R}^{B\times C}, \\ \sigma_m^k &= \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} (\boldsymbol{X}_m^{k,(h,w)} - \mu_m^k)^2} \quad \in \mathbb{R}^{B\times C}, \end{aligned} \tag{2}$$

where $\boldsymbol{X}_m^{k,(h,w)} \in \mathbb{R}^{B\times C}$ represents features at spatial location (h, w). As the abstract of latent features, these statistics carry domain-specific information (*e.g.*, style), and have been instrumental to image generation [33]. They have also been recently used for data augmentation to improve image recognition [35]. In heterogeneous FL scenarios, feature statistics among local clients will be inconsistent, and exhibit unknown feature statistic shifts from the true distribution's statistics. Our method explicitly captures such shift via probabilistic modeling. Specifically, instead of representing each feature $\boldsymbol{X}_m^k$ with deterministic statistics $\{\mu_m^k, \sigma_m^k\}$, we assume that the feature is conditioned on probabilistic statistics $\{\hat{\mu}_m^k, \hat{\sigma}_m^k\}$, which are sampled from the vicinal region of the original statistics based on a multivariate Gaussian distribution:

$$\hat{\mu}_m^k \sim \mathcal{N}(\mu_m^k, \hat{\Sigma}_{\mu_m^k}^2), \quad \hat{\sigma}_m^k \sim \mathcal{N}(\sigma_m^k, \hat{\Sigma}_{\sigma_m^k}^2). \tag{3}$$

Here each Gaussian's center corresponds to the original statistic, and the variance is expected to capture the potential feature statistic shift from the true distribution. Our core goal is thus to estimate appropriate variances $\hat{\Sigma}_{\mu_m^k}^2$ and $\hat{\Sigma}_{\sigma_m^k}^2$ to facilitate reasonable and informative augmentation.

Our approach accomplishes this through three major steps: (i) *client-specific statistic variance estimation* (§3.2.3), which determines local variances within each client ; (ii) *client-sharing statistic variance estimation* (§3.2.4), which determines global variances for the entire federation and (iii) *adaptive variance fusion* (§3.2.5), which combines both local and global variances to yield a more meaningful estimation.

3.2.3 Client-specific Statistic Variance Estimation

In *client-side*, we compute client-specific variances of feature statistics based on the information within each mini-batch:

$$\begin{aligned} \Sigma_{\mu_m^k}^2 &= \frac{1}{B}\sum_{b=1}^{B} (\mu_m^{k,(b)} - \mathbb{E}_B[\mu_m^k])^2 \quad \in \mathbb{R}^C, \\ \Sigma_{\sigma_m^k}^2 &= \frac{1}{B}\sum_{b=1}^{B} (\sigma_m^{k,(b)} - \mathbb{E}_B[\sigma_m^k])^2 \quad \in \mathbb{R}^C, \end{aligned} \tag{4}$$

where $\mu_m^{k,(b)} \in \mathbb{R}^C$ and $\sigma_m^{k,(b)} \in \mathbb{R}^C$ represent the feature mean and standard deviation of the b-th image, respectively. $\mathbb{E}_B[\cdot]$ computes the expectation along the batch dimension. $\Sigma_{\mu_m^k}^2$ and $\Sigma_{\sigma_m^k}^2$ denote the variance of feature mean μ_m^k and standard deviation σ_m^k that are specific to each client. Each value in $\Sigma_{\mu_m^k}^2$ or $\Sigma_{\sigma_m^k}^2$ corresponds to the variance of feature statistics in a particular channel, and its magnitude manifests how the channel will potentially change in the feature space.

3.2.4 Client-sharing Statistic Variance Estimation

The client-specific variances are solely computed based on the data in each individual client, and thus likely biased due to local dataset bias. To solve this, we further estimate client-sharing feature statistic variances taking information of all clients into account. Particularly, we maintain a momentum version of feature statistics for each client, which are online estimated during training:

$$\begin{aligned} \bar{\mu}_m^k &\leftarrow \alpha\bar{\mu}_m^k + (1-\alpha)\mathbb{E}_B[\mu_m^k] \quad \in \mathbb{R}^C, \\ \bar{\sigma}_m^k &\leftarrow \alpha\bar{\sigma}_m^k + (1-\alpha)\mathbb{E}_B[\sigma_m^k] \quad \in \mathbb{R}^C, \end{aligned} \tag{5}$$

where $\bar{\mu}_m^k$ and $\bar{\sigma}_m^k$ are the updated feature statistics of layer h^k in client m, and they are initialized as C-dimensional

all-zero and all-one vectors, respectively. α is a momentum coefficient. We set a same α for both updating, and found no benefit to set it differently. In each communication, these accumulated local feature statistics are sent to the server along with model parameters. Let $\bar{\mu}^k = [\bar{\mu}_1^k, \ldots, \bar{\mu}_M^k] \in \mathbb{R}^{M \times C}$ and $\bar{\sigma}^k = [\bar{\sigma}_1^k, \ldots, \bar{\sigma}_M^k] \in \mathbb{R}^{M \times C}$ denote collections of accumulated feature statistics of all clients, client-sharing statistic variances are determined in *server-side* by:

$$\begin{aligned} \Sigma^2_{\mu^k} &= \frac{1}{M}\sum\nolimits_{m=1}^{M}(\bar{\mu}_m^k - \mathbb{E}_M[\bar{\mu}^k])^2 \quad \in \mathbb{R}^C, \\ \Sigma^2_{\sigma^k} &= \frac{1}{M}\sum\nolimits_{m=1}^{M}(\bar{\sigma}_m^k - \mathbb{E}_M[\bar{\sigma}^k])^2 \quad \in \mathbb{R}^C. \end{aligned} \tag{6}$$

In addition, it is intuitive that certain channels are more likely to change than others, and it will be beneficial to highlight these channels to ensure a sufficient and reasonable exploration of the feature statistic space. To this end, we modulate client-sharing estimations with a *Student's t-distribution* [88], [89] with one degree of freedom to convert the variances to probabilities. The *t-distribution* has heavier tails compared to alternatives such as Gaussian distribution, allowing to highlight the channels with larger statistic variances while avoiding excessive penalization of others. Formally, let $\Sigma^{2,(j)}_{\mu^k}$ and $\Sigma^{2,(j)}_{\sigma^k}$ represent the shared variances of the jth channel in $\Sigma^2_{\mu^k}$ and $\Sigma^2_{\sigma^k}$ (*c.f.* Eq. 6), respectively. They are modulated using the *t-distribution* as follows:

$$\begin{aligned} \gamma^{(j)}_{\mu^k} &= \frac{C(1 + {}^1\!/\!_{\Sigma^{2,(j)}_{\mu^k}})^{-1}}{\sum_{c=1}^{C}(1 + {}^1\!/\!_{\Sigma^{2,(c)}_{\mu^k}})^{-1}} \quad \in \mathbb{R}, \\ \gamma^{(j)}_{\sigma^k} &= \frac{C(1 + {}^1\!/\!_{\Sigma^{2,(j)}_{\sigma^k}})^{-1}}{\sum_{c=1}^{C}(1 + {}^1\!/\!_{\Sigma^{2,(c)}_{\sigma^k}})^{-1}} \quad \in \mathbb{R}, \end{aligned} \tag{7}$$

where $\gamma^{(j)}_{\mu^k}$ and $\gamma^{(j)}_{\sigma^k}$ denote the modulated variances of the j-th channel. By applying Eq. 7 to each channel independently, we obtain $\gamma_{\mu^k} = [\gamma^{(1)}_{\mu^k}, \ldots, \gamma^{(C)}_{\mu^k}] \in \mathbb{R}^C$ and $\gamma_{\sigma^k} = [\gamma^{(1)}_{\sigma^k}, \ldots, \gamma^{(C)}_{\sigma^k}] \in \mathbb{R}^C$ as modulated statistic variances of all feature channels at layer h^k. In this way, the channels with large values in $\Sigma^2_{\mu^k}$ (or $\Sigma^2_{\sigma^k}$) are assigned with much higher importance in γ_{μ^k}(or γ_{σ^k}) than other channels, allowing for more extensive augmentation along those directions.

3.2.5 Adaptive Variance Fusion

The modulated client-sharing estimations $\{\gamma_{\mu^k}, \gamma_{\sigma^k}\}$ provide a quantification of distribution differences among clients, and larger values indicate a higher potential of more significant changes in corresponding channels within the underlying feature statistic space. To incorporate this information for local learning, we weight the client-specific statistic variances $\{\Sigma^2_{\mu^k_m}, \Sigma^2_{\sigma^k_m}\}$ by $\{\gamma_{\mu^k}, \gamma_{\sigma^k}\}$, so that each client has a sense of such differences. To prevent excessive modifications of the client-specific statistic variances, a residual layer is applied during fusion, resulting in estimations of Gaussian ranges as:

$$\begin{aligned} \hat{\Sigma}^2_{\mu^k_m} &= (\gamma_{\mu^k} + 1) \odot \Sigma^2_{\mu^k_m} \quad \in \mathbb{R}^C, \\ \hat{\Sigma}^2_{\sigma^k_m} &= (\gamma_{\sigma^k} + 1) \odot \Sigma^2_{\sigma^k_m} \quad \in \mathbb{R}^C, \end{aligned} \tag{8}$$

where $\odot$ denotes the Hadamard product.

Algorithm 1 FEDFA^l: training phase. (We omit the parameter updating procedure, which is exactly same to FEDAVG.

Input: Number of clients M; number of communication rounds T; neural network $f = g \circ h$; each $\hat{\boldsymbol{X}}^0_m$ represents the collection of training images in client m;
Output: $\gamma_{\mu^k}, \gamma_{\sigma^k}$;

for $t = 1, 2, \ldots, T$ **do**
  for each client $m \in [M]$ **do**
    $\bar{\mu}_m = \mathbf{0}$, $\bar{\sigma}_m = \mathbf{1}$
    for each layer $k \in [K]$ **do**
      $\boldsymbol{X}^k_m = h^k(\hat{\boldsymbol{X}}^{k-1}_m)$
      $\hat{\boldsymbol{X}}^k_m, \bar{\mu}^k_m, \bar{\sigma}^k_m = \text{FFA}(\boldsymbol{X}^k_m, \bar{\mu}^k_m, \bar{\sigma}^k_m)$ ▷ Algo. 2
    $Y = g(\hat{\boldsymbol{X}}^K_m)$
    Run loss computation and optimization
  $\Sigma^2_{\mu^k} = \frac{1}{M}\sum_{m=1}^{M}(\bar{\mu}^k_m - \mathbb{E}[\bar{\mu}^k])^2$ ▷ Eq. 6
  $\Sigma^2_{\sigma^k} = \frac{1}{M}\sum_{m=1}^{M}(\bar{\sigma}^k_m - \mathbb{E}[\bar{\sigma}^k])^2$
  for each channel $j \in [C]$ **do** ▷ Eq. 7
    $\gamma^{(j)}_{\mu^k} = \frac{C(1+1/\Sigma^{2,(j)}_{\mu^k})^{-1}}{\sum_{c=1}^{C}(1+1/\Sigma^{2,(c)}_{\mu^k})^{-1}}$
    $\gamma^{(j)}_{\sigma^k} = \frac{C(1+1/\Sigma^{2,(j)}_{\sigma^k})^{-1}}{\sum_{c=1}^{C}(1+1/\Sigma^{2,(c)}_{\sigma^k})^{-1}}$
return $\gamma_{\mu^k}, \gamma_{\sigma^k}$

3.2.6 Federated Feature Augmentation Layer

After establishing the Gaussian distribution, we design a federated feature augmentation (FFA) layer to synthesize novel feature $\hat{\boldsymbol{X}}^k_m$ in the vicinity of $\boldsymbol{X}^k_m$ as:

$$\begin{aligned} &\hat{\boldsymbol{X}}^k_m = \text{FFA}(\boldsymbol{X}^k_m) = \hat{\sigma}^k_m \frac{\boldsymbol{X}^k_m - \mu^k_m}{\sigma^k_m} + \hat{\mu}^k_m, \\ &\text{where} \quad \hat{\mu}^k_m \sim \mathcal{N}(\mu^k_m, \hat{\Sigma}^2_{\mu^k_m}), \quad \hat{\sigma}^k_m \sim \mathcal{N}(\sigma^k_m, \hat{\Sigma}^2_{\sigma^k_m}). \end{aligned} \tag{9}$$

Here $\boldsymbol{X}^k_m$ is first normalized with its original statistics by ${}^{(\boldsymbol{X}^k_m - \mu^k_m)}\!/\!_{\sigma^k_m}$, and further scaled with novel statistics $\{\hat{\mu}^k_m, \hat{\sigma}^k_m\}$ that are randomly sampled from corresponding Gaussian distribution. To make the sampling differentiable, we use the re-parameterization trick [90]:

$$\hat{\mu}^k_m = \mu^k_m + \epsilon_\mu \hat{\Sigma}_{\mu^k_m}, \quad \hat{\sigma}^k_m = \sigma^k_m + \epsilon_\sigma \hat{\Sigma}_{\sigma^k_m}, \tag{10}$$

where $\epsilon_\mu \sim \mathcal{N}(0, 1)$ and $\epsilon_\sigma \sim \mathcal{N}(0, 1)$ follow the normal Gaussian distribution.

FFA is a plug-and-play layer, *i.e.*, it can be inserted at arbitrary layers in the feature extractor h. In our implementation, we add a FFA layer after each convolutional stage of the networks (Fig. 1 illustrates how FFA is inserted into AlexNet in our experiments). During training, we follow the stochastic learning strategy [34], [67], [68] to activate each FFA layer with a probability p (0.5 by default). This allows for more diverse augmentation from iteration to iteration (based on the activated FFA layers). At test time, no augmentation is applied. We summarize FEDFA^l and FFA in Algo. 1 and Algo. 2, respectively.

3.3 Federated Feature Alignment (FEDFA^h)

3.3.1 Marginal Feature Distribution Approximation by High-Order Statistics

While the idea of aligning marginal feature distribution across domains is straightforward in centralized domain generalization settings, it becomes challenging in FL due to privacy constraints. To overcome this challenge, we propose

Algorithm 2 Algorithm description of FFA for the kth layer in client m.

Input: Original feature $\boldsymbol{X}_m^k$; momentum $\alpha = 0.99$; probability $p = 0.5$; client-sharing fusion coefficients $\gamma_{\mu^k} \in \mathbb{R}^C$ and $\gamma_{\sigma^k} \in \mathbb{R}^C$ downloaded from the server; accumulated feature statistics $\bar{\mu}_m^k \in \mathbb{R}^C$ and $\bar{\sigma}_m^k \in \mathbb{R}^C$;
Output: Augmented feature $\hat{\boldsymbol{X}}_m^k, \bar{\mu}_m^k, \bar{\sigma}_m^k$;

if np.random.random() $< p$ **then**
- $\mu_m^k = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} \boldsymbol{X}_m^{k,(h,w)}$ ▷ Eq. 2
- $\sigma_m^k = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(\boldsymbol{X}_m^{k,(h,w)} - \mu_m^k)^2}$
- $\Sigma^2_{\mu_m^k} = \frac{1}{B}\sum_{b=1}^{B}(\mu_m^k - \mathbb{E}[\mu_m^k])^2$ ▷ Eq. 4
- $\Sigma^2_{\sigma_m^k} = \frac{1}{B}\sum_{b=1}^{B}(\sigma_m^k - \mathbb{E}[\sigma_m^k])^2$
- $\hat{\Sigma}^2_{\mu_m^k} = (\gamma_{\mu^k} + 1)\Sigma^2_{\mu_m^k}$ ▷ Eq. 8
- $\hat{\Sigma}^2_{\sigma_m^k} = (\gamma_{\sigma^k} + 1)\Sigma^2_{\sigma_m^k}$
- $\hat{\mu}_m^k = \mu_m^k + \epsilon_\mu \hat{\Sigma}_{\mu_m^k}$ ▷ Eq. 10
- $\hat{\sigma}_m^k = \sigma_m^k + \epsilon_\sigma \hat{\Sigma}_{\sigma_m^k}$
- $\hat{\boldsymbol{X}}_m^k = \hat{\sigma}_m^k \frac{\boldsymbol{X}_m^k - \mu_m^k}{\sigma_m^k} + \hat{\mu}_m^k$ ▷ Eq. 9
- $\bar{\mu}_m^k \leftarrow \alpha\bar{\mu}_m^k + (1-\alpha)\frac{1}{B}\sum_{b=1}^{B}\mu_m^k$ ▷ Eq. 5
- $\bar{\sigma}_m^k \leftarrow \alpha\bar{\sigma}_m^k + (1-\alpha)\frac{1}{B}\sum_{b=1}^{B}\sigma_m^k$

return $\hat{\boldsymbol{X}}_m^k, \bar{\mu}_m^k, \bar{\sigma}_m^k$

the use of feature histograms as a practical approximation of the marginal feature distribution within each client. These histograms can be interpreted as high-order feature statistics. They are computed using a differentiable soft binning function [36], [37].

Formally, denote $z_c \sim p_m^{z_c}$ as a continuous 1D variable for which we have N_m samples $\{z_c^i\}_{i=1}^{N_m}$. Here $p_m^{z_c}$ is the marginal feature distribution of the c-dimensional feature in client m, and z_c^i denotes the c-th dimension of the i-th sample's feature. Following [36], [37], we approximately parameterize $p_m^{z_c}$ using a histogram with L normalized bin counts $\boldsymbol{\pi}_m^{z_c} = [\pi_m^{z_c,1}, \cdots, \pi_m^{z_c,L}]$, which is computed as:

$$\boldsymbol{\pi}_m^{z_c} = \sum_{i=1}^{N_m} \frac{\Phi(z_c^i)}{N_m} \quad \in [0,1]^L, \tag{11}$$

where Φ is the vector-valued soft binning function:

$$\begin{aligned} &\Phi(z_c^i) = \text{softmax}\left(\left(\boldsymbol{w}\hat{z}_c^i + \boldsymbol{b}\right)/\tau\right), \\ &\text{where} \quad \hat{z}_c^i = \frac{z_c^i - z_c^{min}}{z_c^{max} - z_c^{min}} \quad \in [0,1], \\ &\text{and} \quad z_c^{min} = \min_i z_c^i, \quad z_c^{max} = \max_i z_c^i. \end{aligned} \tag{12}$$

As seen, Φ is defined as a softmax layer with a temperature τ. Its input is obtained by applying a linear transformation to the normalized feature $\hat{z}_c^i$, compuated by scaling z_c^i to the range $[0,1]$ using the minimum z_c^{min} and maximum z_c^{max} values of $\{z_c^i\}_i$ observed in the client's samples. Notably, $\boldsymbol{w}$ and $\boldsymbol{b}$ are constant vectors rather than learnable parameters. Following [36], [37], they are set as $\boldsymbol{w} = [1, 2, \ldots, L]$ and $\boldsymbol{b} = [0, -\rho_1, -\rho_1 - \rho_2, \ldots, -\sum_{l=1}^{L-1}\rho_l]$, respectively, where $[\rho_1, \rho_2, \ldots, \rho_{L-1}] = \frac{1}{L-2}[0, 1, 2, \ldots, L-2]$ are $L-1$ uniformly-spaced and monotonically-increasing cut points over the normalized range $[0,1]$.

By computing the histogram for each of the C dimensions, we obtain a full approximation for client m as:

$$\boldsymbol{\pi}_m = [\boldsymbol{\pi}_m^{z_1}; \boldsymbol{\pi}_m^{z_2}; \ldots; \boldsymbol{\pi}_m^{z_C}] \quad \in [0,1]^{LC}. \tag{13}$$

Remark. The soft binning function makes few assumptions about the form of marginal distributions, and thus is flexible to handle diverse situations with complex distributions. Furthermore, the storage requirements for the soft histogram approximation are constant, *i.e.*, $\mathcal{O}(LC)$, scaling linearly with the bin number L and feature dimension C. This scalability makes it feasible to communicate the histograms between clients and the server. In practice, we perform alignment only for the final layer of the feature extractor h^K to conserve network bandwidth (see Fig. 1). As will be shown in Tables 3-7, promising performance improvements can be achieved with the selective alignment strategy.

3.3.2 High-Order Feature Statistic Alignment

After local model training, each $\boldsymbol{\pi}_m$ is uploaded to the server, along with model weights. These histograms are aggregated by averaging to yield a global histogram $\boldsymbol{\pi} = \frac{\sum_{m=1}^{M}\boldsymbol{\pi}_m}{M}$. Subsequently, $\boldsymbol{\pi}$ is distributed back to each client to guide local model learning in the next round. The objective is to align the approximated feature distribution in the b-th mini-batch $\boldsymbol{\pi}_m^{(b)}$ with the global histogram π using the symmetric KL divergence:

$$\mathcal{L}_m^{\text{FedFA}^h} = \frac{1}{2}(D_{\text{KL}}(\boldsymbol{\pi}_m^b \| \boldsymbol{\pi}) + D_{\text{KL}}(\boldsymbol{\pi} \| \boldsymbol{\pi}_m^b)), \tag{14}$$

where D_{KL} is the KL divergence.

3.4 Training Loss

The overall loss function of our method are:

$$\mathcal{L} \triangleq \frac{1}{M}\sum\nolimits_{m\in[M]} \mathcal{L}_m^{\text{FedFA}^l} + \lambda\mathcal{L}_m^{\text{FedFA}^h}. \tag{15}$$

Here the coefficient λ balances the two terms. $\mathcal{L}_m^{\text{FedFA}^l}$ denotes the training loss function of FedFA^l in client m. A rigorous theoretical analysis of $\mathcal{L}_m^{\text{FedFA}^l}$ is provided in Theorem 4.1. For FedFA^l, we set the coefficient to $\lambda = 0$, while for FedFA+, we empirically set it to $\lambda = 0.1$.

4 Theoretical Analysis

In this section, we provide mathematical analysis to gain more insights on FedFA^l. We begin with interpreting FedFA^l as a noise injection process [91]–[93], which is a case of VRM, and show that FedFA^l injects federation-aware noises to latent features (§4.1). Next, we show that, induced by federation-aware noise injection, FedFA^l exhibits a natural form of federation-aware implicit regularization to local client training (§4.2). Without loss of generality, we conduct all analysis for an arbitrary client $m \in [M]$.

4.1 Understanding FedFA^l as Federation-Aware Noise Injection

4.1.1 Noise Injection in Neural Networks

Let x be a training sample and $\boldsymbol{x}^k$ its latent representation at the k-th layer, with no noise injections. The $\boldsymbol{x}^k$ can be noised under a process $\hat{\boldsymbol{x}}^k = \boldsymbol{x}^k + \boldsymbol{e}^k$, where $\boldsymbol{e}^k$ is an addition noise drawn from a probability distribution, and $\hat{\boldsymbol{x}}^k$ is the noised representation of $\boldsymbol{x}^k$.

A popular choice of $\boldsymbol{e}^k$ is isotropic Gaussian noise [92], *i.e.*, $\boldsymbol{e}^k \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, where $\boldsymbol{I}$ is an identity matrix and σ

is a scalar, controlling the amplitude of $\boldsymbol{e}^k$. To avoid over-perturbation that may cause model collapse, σ is typically set as a small value. Despite its simplicity, the strategy serves as an effective regularization for tackling domain generalization [76] and adversarial samples [94], [95]. However, as shown in Table 9, its performance (see FEDFA-R) is only marginally better or sometimes worse than FEDAVG in FL.

4.1.2 Federation-Aware Noise Injection

In Eq. 10, it is evident that the feature statistic augmentation in FFA follows the noise injection process above. Subsequently, we show that this ultimately results in features perturbed under a federation-aware noising procedure.

Lemma 4.1. *Consider client m, for a batch-wise latent feature $\boldsymbol{X}_m^k$ at the k-th layer, its augmentation in* FEDFA^l *follows a noising process $\hat{\boldsymbol{X}}_m^k = \boldsymbol{X}_m^k + \boldsymbol{e}_m^k$, with the noise $\boldsymbol{e}_m^k$ taking the form:*

$$\boldsymbol{e}_m^k = \epsilon_\sigma \hat{\Sigma}_{\sigma_m^k} \bar{\boldsymbol{X}}_m^k + \epsilon_\mu \hat{\Sigma}_{\mu_m^k}, \tag{16}$$

where $\epsilon_\mu \sim \mathcal{N}(0,1)$, $\epsilon_\sigma \sim \mathcal{N}(0,1)$, $\bar{\boldsymbol{X}}_m^k = (\boldsymbol{X}_m^k - \mu_m^k)/\sigma_m^k$.

Proof of Lemma 4.1. By substituting Eq. 10 into Eq. 9, we have

$$\begin{aligned}
\hat{\boldsymbol{X}}_m^k &= \hat{\sigma}_m^k \frac{\boldsymbol{X}_m^k - \mu_m^k}{\sigma_m^k} + \hat{\mu}_m^k, \\
&= (\sigma_m^k + \epsilon_\sigma \hat{\Sigma}_{\sigma_m^k}) \frac{\boldsymbol{X}_m^k - \mu_m^k}{\sigma_m^k} + (\mu_m^k + \epsilon_\mu \hat{\Sigma}_{\mu_m^k}), \\
&= \boldsymbol{X}_m^k + \underbrace{(\epsilon_\sigma \hat{\Sigma}_{\sigma_m^k} \bar{\boldsymbol{X}}_m^k + \epsilon_\mu \hat{\Sigma}_{\mu_m^k})}_{\boldsymbol{e}_m^k},
\end{aligned} \tag{17}$$

where $\epsilon_\mu \sim \mathcal{N}(0,1), \epsilon_\sigma \sim \mathcal{N}(0,1), \bar{\boldsymbol{X}}_m^k = (\boldsymbol{X}_m^k - \mu_m^k)/\sigma_m^k$.

As compared to Gaussian noise injections [92], [94], [95], the noise term $\boldsymbol{e}_m^k$ in FEDFA^l shows three desiderata: it is (i) ***data-dependent***, adaptively determined based on the normalized input feature $\bar{\boldsymbol{X}}_m^k$; (ii) ***channel-independent***, allowing for more extensive exploration along different directions in the feature space; (iii) most importantly ***federation-aware***, *i.e.*, its strength is controlled by statistic variances $\hat{\Sigma}_{\mu_m^k}$ and $\hat{\Sigma}_{\sigma_m^k}$ (*cf.* Eq. 8), which are known carrying global statistic information of all participating clients.

4.2 Federation-Aware Implicit Regularization in FEDFA^l

Next we show that with noise injections, FEDFA^l imposes federation-aware implicit regularization to local client training. By this, we mean regularization imposed implicitly by the stochastic learning strategy, without explicit modification of the loss, and the regularization effect is affected by the federation-aware noise (in Lemma 4.1).

Recall the deep neural network f defined in §3.1: $f \triangleq g \circ h$, where $h = h^K \circ h^{K-1} \circ \cdots \circ h^1$ is a K-layer CNN feature extractor and g is a classifier. Given a batch of samples X_m with labels Y_m, its latent representation at the k-th layer is computed as $\boldsymbol{X}_m^k = h^k \circ h^{k-1} \circ \cdots \circ h^1(X_m)$, or we write it in a simpler form, $\boldsymbol{X}_m^k = h^{1:k}(X_m)$. Note that we only add noises to layers in h, but not to g. Concretely, in each mini-batch training, FEDFA^l follows a stochastic optimization strategy to randomly select a subset of layers from $\{h^k\}_{k=1}^K$ and add noises to them. For simplicity, we denote $\mathcal{K} = \{1, \ldots, K\}$ as the index of all layers in h, $\mathcal{Z} \subseteq \mathcal{K}$ as the subset of layer indexes that are selected, $\mathcal{E} = \{\boldsymbol{e}_m^z\}_{\forall z \in \mathcal{Z}}$ as the corresponding set of noises. Then, the loss function $\mathcal{L}_m^{\text{FEDFA}^l}$ of client m in FEDFA^l can be equivalently written as $\mathcal{L}_m^{\text{FEDFA}^l} = \mathbb{E}_{\mathcal{Z} \sim \mathcal{K}} \mathcal{L}_m^{\mathcal{Z}}$, where $\mathcal{L}_m^{\mathcal{Z}}$ is a standard loss function $\mathcal{L}_m^{\text{ERM}}$ (*cf.* Eq. 1) imposed by adding noises to layers in $\mathcal{Z}$. In the remainder, we relate the loss function $\mathcal{L}_m^{\text{FEDFA}^l}$ to the original ERM loss $\mathcal{L}_m^{\text{ERM}}$ as well as a regularization term conditioned on $\mathcal{E}$.

Theorem 4.1. *In* FEDFA^l*, the loss function $\mathcal{L}_m^{\text{FEDFA}^l}$ of client m can be expressed as:*

$$\mathcal{L}_m^{\text{FEDFA}^l} = \mathcal{L}_m^{ERM} + \mathcal{L}_m^{REG}. \tag{18}$$

Here $\mathcal{L}_m^{ERM}$ is the ERM loss, and $\mathcal{L}_m^{REG}$ is the regularization term:

$$\begin{aligned}
\mathcal{L}_m^{ERM} &= \mathbb{E}_{(X_m, Y_m) \sim \mathbb{P}_m} \ell(g(h^{1:K}(X_m)), Y_m), \\
\mathcal{L}_m^{REG} &= \mathbb{E}_{\mathcal{Z} \sim \mathcal{K}} \mathbb{E}_{(X_m, Y_m) \sim \mathbb{P}_m} \\
&\quad \nabla_{h^{1:K}(X_m)} \ell(g(h^{1:K}(X_m)), Y_m)^\top \sum_{z \in \mathcal{Z}} \boldsymbol{J}^z(X_m) \boldsymbol{e}_m^z,
\end{aligned}$$

where $\boldsymbol{J}^z \in \mathbb{R}^{C_K \times C_z}$ indicates the Jacobian of layer z, i.e., *$\boldsymbol{J}^z(X)_{i,j} = \frac{\partial h^{1:K}(X_m)_i}{\partial h^{1:z}(X_m)_j}$, where C_K and C_z denote the number of neurons in layer K and z, respectively.*

Theorem 4.1 implies that, FEDFA^l implicitly introduces regularization to local client learning by regularizing the gradients of latent representations (*i.e.*, $\nabla_{h^{1:K}(X_m)} \ell(g(h^{1:K}(X_m)), Y_m)^\top$), weighted by federation-aware noises $\sum_{z \in \mathcal{Z}} \boldsymbol{J}^z(X_m) \boldsymbol{e}_m^z$. The proof of Theorem 4.1 is provided in the appendix.

5 EMPIRICAL RESULT

5.1 Setup

Datasets. We validate our algorithms on six datasets in terms of feature distribution heterogeneity (Office-Caltech 10 [96], DomainNet [97], ProstateMRI [98], see Fig. 2), label distribution heterogeneity (CIFAR-10 [99] and Tiny-ImageNet [40]), and data size heterogeneity (FEMNIST [38]):

- **Office-Caltech 10** [96] has four data sources, with three originating from Office-31 [100] and one from Caltech-256 [101]. They are collected from different camera devices or in diverse environments with varying background.
- **DomainNet** [97] contains images from six domains, which are gathered by searching a category name along with a domain name in different search engines.
- **ProstateMRI** [98] is a prostate segmentation dataset including six data sources of T2-weighted MRI from various medical institutions.
- **CIFAR-10** [99] is a classification dataset. It contains 50K training and 10K test images. Heterogeneity is simulated by sampling local data based on Dirichlet distribution.
- **Tiny-ImageNet** [40] contains 100K training and 10K testing images of 200 classes. Each image is downsampled to a resolution of 64×64. Heterogeneity is simulated in a same way as CIFAR-10.
- **FEMNIST** [38] is an image classification dataset with 62 classes, including all 26 capital and small letter of alphabet as well as numbers. We follow the setup in FEDSCALE [39] to simulate a FL setting with thousands of clients.

Baselines. For a comprehensive evaluation, we compare our approach against several state-of-the-art federated learning

TABLE 1
Summary of key experimental configuration for each dataset.

Hyper-parameters	Office	DomainNet	ProstateMRI	CIFAR-10	Tiny-ImageNet	FEMNIST
federation-aware configuration						
# Rounds	400	400	500	100	500	1000
# Epochs per round	1	1	1	10	5	5
# Clients	4	6	6	100	100	3400
# Categories	10	10	2	10	200	62
Participation rate	1.0	1.0	1.0	0.1	0.05	0.015
local client training configuration						
Network	AlxeNet	AlexNet	U-Net	ResNet	ResNet	ResNet
Optimizer	SGD	SGD	Adam	SGD	SGD	SGD
Local batch size	32	32	16	10	20	32
Local learning rate	1e-2	1e-2	1e-4	1e-1	1e-1	5e-2

TABLE 2
Numbers of samples in the training, validation, and testing sets of each client in Office-Caltech 10, DomainNet, and ProstateMRI.

Split	Office-Caltech 10 [96]				DomainNet [97]						ProstateMRI [98]					
	Amazon	Caltech	DSLR	Webcam	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	BIDMC	HK	I2CVB	BMC	RUNMC	UCL
train	459	538	75	141	672	840	791	1280	1556	708	156	94	280	230	246	105
val	307	360	50	95	420	525	494	800	972	442	52	31	93	76	82	35
test	192	225	32	59	526	657	619	1000	1217	554	52	31	93	76	82	35

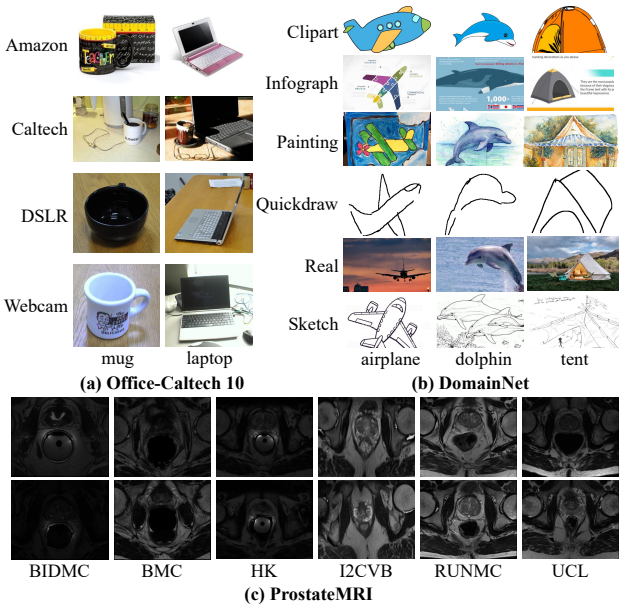


Fig. 2. Representative images in (a) Office-Caltech 10 [96], (b) DomainNet [97] and (c) ProstateMRI [98] with feature-level shift across clients.

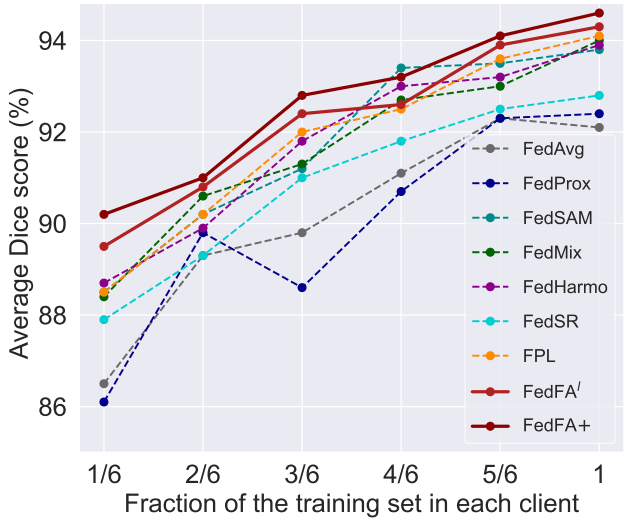


Fig. 3. Segmentation performance on ProstateMRI *w.r.t* local data size (*i.e.*, fraction of training samples over the entire training set). See §5.2.2.

techniques, including FedAvg [5], FedAvgM [102], FedProx [42], FedSAM [18], FedBN [14], FedRobust [15], FedMix [28], FedSR [19], and FPL [23]. We also compare with FedHarmo [16] in ProstateMRI, which are specifically designed for medical imaging. For federated domain generalization, we additionally compare against very recent approaches, *i.e.*, FedDG [22], FedGA [20], and Ccst [21].

Furthermore, to gain more insights into FedFA^l, we develop two baselines: FedFA-R(andom) and FedFA-C(lient). FedFA-R randomly perturbs feature statistics based on Gaussian distribution with a same standard deviation for all channels, *i.e.*, $\hat{\Sigma}_{\mu_m^k} = \hat{\Sigma}_{\sigma_m^k} = 0.5$. FedFA-C performs augmentation based only on client-specific variances, *i.e.*, Eq. 8 becomes $\hat{\Sigma}^2_{\mu_m^k} = \Sigma^2_{\mu_m^k}$, $\hat{\Sigma}^2_{\sigma_m^k} = \Sigma^2_{\sigma_m^k}$.

Evaluation Protocol. As conventions [5], [14], [16], [28], we use top-1 accuracy for image classification and Dice coefficient for medical image segmentation, respectively. All scores are reported for **the global model**.

Implementation Details. We implement our algorithms and other baselines using PyTorch. Following FedBN [14], we adopt AlexNet [54] on Office-Caltech 10 and DomainNet, using the SGD optimizer with learning rate 0.01 and batch size 32. In line with FedHarmo [16], we employ U-Net [103] on ProstateMRI using Adam as the optimizer with learning rate 1e-4 and batch size 16. The communication rounds are set to 400 for Office-Caltech 10 and DomainNet, and 500 for ProstateMRI, with the number of local update epoch setting to 1 in all cases. For CIFAR-10 and Tiny-ImageNet, we sample local data based on Dirichlet distribution to simulate label distribution heterogeneity. As [18], [104]–[106], we set α to 0.3 or 0.6, and train a ResNet-18

[55]. The client number is 100 with participation rate 0.1. The training is carried out for 100 rounds in total. τ in Eq. 12 is empirically set to 0.01 by default. For FEMNIST, we strictly follow FEDSCALE [39] to verify our approach under a large number (3,400) of local clients. Notably, unlike the simulated non-IID distribution in CIFAR-10, FEDSCALE studies real-world non-IID patterns in realistic datasets, that is, no data sampling is involved in the training stage. We train a ResNet-18 using SGD with batch size 32. We run 1,000 rounds in total, and 50 clients are sampled per communication round. A summary of key training configuration is provided in Table 1. Beyond the CNN backbones listed in Table 1, we investigate the applicability of our approach in terms of Transformer architectures. Particularly, we study Swin-T(iny) [107]. As Fig. 1, we add one FFA layer after each stage in Swin and the alignment component is only applied to the last stage. We follow the official data splits for Office, DomainNet and ProstateMRI, which are summarized in Table 2. For EMNIST and CIFAR-10, we strictly follow the strategies in [18], [28], [104] to partition the data.

5.2 Main Result

5.2.1 Performance on Office-Caltech 10 and DomainNet

As reported in Table 3, FEDFA+ achieves consistent performance gains over the competitors on both the Office-Caltech 10 and DomainNet benchmarks. The gains achieved by FEDFA+ over FEDAVG are substantial, reaching **6.3%** and **4.7%** on the respective benchmarks. In comparison to the advanced method FPL [23], FEDFA+ also shows promising improvements of **2.0%** and **1.8%**. Moreover, we observe immediate improvements (**1.1%**/**1.8%** on Office/DomainNet) after employing Swin-T as the backbone. This verifies the applicability of our method to Transformer architectures.

5.2.2 Performance on ProstateMRI

In certain practical scenarios like healthcare, the size of local dataset might be very small, which poses a challenge for FL. To assess the performance of FL algorithms under these conditions, we construct *mini*-ProstateMRI by randomly sampling only $1/6$ of all training samples in each client for training. Note that the test set is kept identical to FEDHARMO. As seen from Table 4, FEDFA+ yields a **1.4%** improvement in overall performance against the second-best approach FEDBN, reaching a **90.2%** Dice score. Further, our approach improves the average score from 90.2% to **91.3%** after using the Swin-T backbone. Additionally, Fig. 3 shows how the performance of methods changes *w.r.t.* the size of local training data. We train methods with various fractions (*i.e.*, $1/6$, $2/6$, $3/6$, $4/6$, $5/6$, 1) of training samples. Here a sampling rate of '1' means that all training data are used without downsampling. We see from the figure that FEDFA+ show promising performance in all cases.

5.2.3 Performance on CIFAR-10, FEMNIST and Tiny-ImageNet

Apart from addressing feature-shift non-i.i.d., our algorithms show consistent improvements in label distribution heterogeneity on CIFAR-10 [99]. As depicted in Table 5, FEDFA+ surpasses FEDMIX by **1.8%** and **1.9%** *w.r.t.* two non-i.i.d levels Dir(0.6) and Dir(0.3), respectively. Moreover, we investigate the effect of FEDFA+ in terms of a larger number of classes in Tiny-ImageNet. Under two non-i.i.d levels Dir(0.6) and Dir(0.3), FEDFA+ yields consistent and notable improvements. In FEMNIST, which contains 3,400 clients, our approach demonstrates leading performance, and outperforms the second-best algorithm FEDMIX by **0.8%**. Last, employing Swin-T as the backbone leads to consistent improvements on the three datasets.

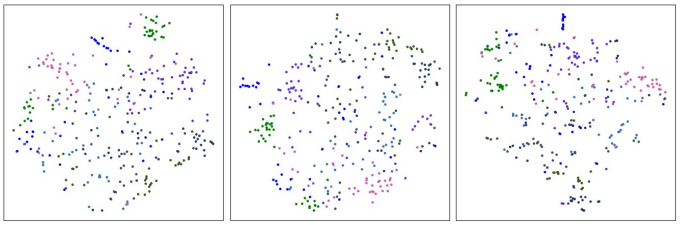


Fig. 4. Feature visualization using t-sne for FEDAVG (left), FEDFA^l (middle), and FEDFA+ (right) on Office `test`. See §5.4.1 for details.

5.3 Federated Domain Generalization Performance

FL systems exhibit dynamic behavior, where new clients may join the system after model training, most possibly with distribution shift from old clients that participated in the training. However, the majority of existing FL algorithms focuses on enhancing model performance for participating clients, overlooking model generalizability to unseen, non-participating clients. As distribution shift frequently transpire during deployment, it is essential to evaluate the generalizability of FL algorithms in such situations. With feature augmentation and alignment as the foundation, FEDFA+ is supposed to alleviate domain gap during training, and potentially enhance the model's generalization capacity.

To validate this, we conduct experiments in the setup of federated domain generalization based on the *leave-one-client-out* strategy, *i.e.*, training on $M-1$ clients and testing on the held-out client. As seen from Table 6, FEDFA+ attains leading generalization performance on the majority of unseen clients. For example, it yields consistent improvements as compared to FEDGA [20], *i.e.*, **0.6%** on Office-Caltech 10, **0.5%** on DomainNet, and **0.4%** on ProstateMRI. In addition, our model with Swin-T demonstrates promising gains over FEDFA+, leading to improvements of $0.9\%/0.7\%/0.8\%$ in the three datasets, respectively. Nevertheless, by comparing these results with those reported in Tables 3-4, we find that current FL algorithms still face a considerable performance gap [85], *i.e.*, the performance difference between participating and non-participating clients, which is a critical issue that should be tackled in future.

5.4 Diagnostic Experiment

To gain more insights into our algorithms, we conduct a set of ablation studies on Office, DomainNet and ProstateMRI.

5.4.1 Key Component Analysis

We commence our analysis by examining the two fundamental components in our approach, *i.e.*, FEDFA^l and FEDFA^h. **First**, we observe notable performance improvements of FEDFA^l over FEDAVG across all five datasets.

TABLE 3
Image classification results on Office-Caltech 10 [96] and DomainNet [97] `test` (§5.2.1). Top-1 accuracy (%) is reported. Office-Caltech 10 has 4 clients: A(mazon), C(altech), D(SLR), and W(ebcam), while DomainNet has 6: C(lipart), I(nfograph), P(ainting), Q(uickdraw), R(eal), and S(ketch). 'FEDFA+ (Swin-T)' indicates FEDFA+ using Swin-Tiny [107] as the backbone.

Algorithm	Office-Caltech 10 [96]					DomainNet [97]						
	A	C	D	W	Average	C	I	P	Q	R	S	Average
FEDAVG [AISTAT17] [5]	84.4	66.7	75.0	88.1	78.5	71.5	33.2	57.8	76.5	72.9	65.2	62.8
FEDAVGM [arXiv19] [102]	85.9	64.0	71.9	94.9	79.2	79.8	33.3	58.8	72.6	72.8	66.1	62.5
FEDPROX [MLSys20] [42]	84.9	64.0	78.1	88.1	78.8	70.9	32.9	61.2	74.1	71.1	67.9	63.0
FEDROBUST [NeurIPS20] [15]	82.3	64.0	81.3	93.2	80.2	70.9	32.9	60.7	75.7	72.6	68.5	63.6
FEDBN [ICLR20] [14]	82.3	63.6	81.2	94.9	80.5	72.4	32.7	64.3	74.0	69.9	70.8	64.0
FEDMIX [ICLR21] [28]	81.7	63.1	81.3	93.2	79.8	75.9	34.1	61.7	73.8	69.4	70.6	64.3
FEDSAM [ICML22] [18]	81.7	63.1	50.0	81.4	69.1	60.1	30.1	53.0	64.8	61.9	47.3	52.9
FEDSR [NeurIPS22] [19]	86.7	65.3	79.4	88.1	79.9	73.1	32.8	60.3	73.0	73.7	68.2	63.5
FPL [CVPR23] [23]	86.9	68.8	81.0	94.5	82.8	74.5	35.6	61.3	76.4	75.4	71.1	65.7
FEDFA+	90.5	68.8	90.9	88.5	**84.8**	78.6	37.5	60.2	78.5	74.1	75.9	**67.5**
FEDFA+ (Swin-T)	91.8	71.0	91.6	89.1	**85.9**	80.3	40.6	62.0	79.8	75.8	77.1	**69.3**

TABLE 4
Medical image segmentation accuracy on *mini*-ProstateMRI `test` [98] with small-size local datasets (§5.2.2). Dice score (%) is reported. The number in the bracket denotes the number of training samples in each client.

Algorithm	BIDMC (32)	HK (32)	I2CVB (46)	BMC (38)	RUNMC (41)	UCL (32)	Average
FEDAVG [AISTAT17] [5]	81.2	90.8	86.1	84.0	91.0	86.2	86.5
FEDAVGM [arXiv19] [102]	80.3	91.6	88.2	82.2	91.2	86.5	86.7
FEDPROX [MLSys20] [42]	82.8	89.1	89.8	79.5	89.8	85.6	86.1
FEDROBUST [NeurIPS20] [15]	81.7	91.3	91.5	88.5	89.4	84.2	87.7
FEDBN [ICLR20] [14]	88.9	92.3	90.6	88.1	87.6	85.4	88.8
FEDMIX [ICLR21] [28]	86.3	91.6	89.6	88.1	89.8	85.2	88.4
FEDSAM [ICML22] [18]	82.7	92.5	91.8	83.6	92.6	88.1	88.5
FEDSR [NeurIPS22] [19]	80.1	92.8	88.9	84.0	92.0	89.5	87.9
FEDHARMO [AAAI22] [16]	86.7	91.6	92.7	84.2	92.5	84.6	88.7
FPL [CVPR23] [23]	82.4	91.6	91.3	85.4	93.7	86.5	88.5
FEDFA+	89.1	92.8	90.1	89.0	91.9	88.4	**90.2**
FEDFA+ (Swin-T)	89.8	93.1	92.2	90.3	92.7	89.6	**91.3**

TABLE 5
Image classification results on CIFAR-10 [99], FEMNIST [38], and Tiny-ImageNet [40] (§5.2.3).

Algorithm	CIFAR-10 [99]		Tiny-ImageNet [40]		FEMNIST [38]
	Dir (0.6)	Dir (0.3)	Dir (0.6)	Dir (0.3)	
FEDAVG [AISTAT17] [5]	73.3	69.2	33.9	31.8	78.5
FEDAVGM [arXiv19] [102]	73.4	69.1	36.3	34.2	78.8
FEDPROX [MLSys20] [42]	74.0	69.5	34.3	32.3	78.4
FEDBN [ICLR20] [14]	73.7	69.8	35.6	33.5	79.1
FEDROBUST [NeurIPS20] [15]	74.9	70.5	35.8	33.1	79.3
FEDSAM [ICML22] [18]	74.3	70.0	37.2	35.4	79.2
FEDMIX [ICLR21] [28]	75.5	70.7	36.8	34.9	79.7
FEDFA+	**77.3**	**72.6**	**38.8**	**36.2**	**80.5**
FEDFA+ (Swin-T)	**81.6**	**76.1**	**39.6**	**37.1**	**81.6**

This verifies the efficacy of federated-aware augmentation in addressing feature-level heterogeneity. **Second**, FEDFA^h can improve the performance of FEDAVG, but the improvements are not as significant as FEDFA^l, especially for Office/DomainNet/ProstateMRI that exhibits high degree of feature heterogeneity. However, when combined it with FEDFA^l in the full model FEDFA+, considerable improvements are realized. These results reveal the indispensability of joint federated feature augmentation and alignment. Third, employing the Swin-T backbone leads to substantial improvements of all model variants across all datasets. This confirms the applicability of essential components in our model to more advanced architectures. Last, we examine the impacts of various normalization layers to our approach. As reported in Table 8, our algorithm shows strong robustness to common normalization layers (*i.e.*, BatchNorm, LayerNorm, and GroupNorm). By default, we follow conventions [14], [16] to adopt BatchNorm for CNN architectures.

To gain more in-depth understanding of these components, we visualize features derived from FEDAVG, FEDFA^l, and FEDFA+ on Office `test` in Fig. 4. The features are extracted from the final layer of AlexNet. To better understand the feature distribution, we further quantify the quality of features using the *alignment* and *uniformity* metrics in [110]. Concretely, *alignment* measures the distance between feature pairs with the same class, and a smaller value indicates higher intra-class compactness; *uniformity* measures the pairwise similarity between all samples regardless of category, computed based on Gaussian potential, and a smaller value indicates that the feature distribution is more uniform. On Office `test`, we achieve an *alignment* score (lower the better) of 0.32 for FEDAVG. Our FEDFA^l

TABLE 6
Comparison of generalization performance to unseen test clients on the three benchmarks (§5.3).

Algorithm	Office-Caltech 10 [96]					DomainNet [97]							ProstateMRI [98]						
	A	C	D	W	AVG	C	I	P	Q	R	S	AVG	B	H	I	M	R	U	AVG
FedAvg [AISTAT17] [5]	64.6	49.3	71.9	55.9	60.4	63.1	27.5	49.6	44.7	51.7	48.2	47.5	60.7	85.3	78.4	67.2	83.0	59.0	72.3
FedProx [MLSys20] [42]	63.0	50.7	68.7	62.7	61.3	62.2	26.9	49.6	42.4	50.5	48.9	46.8	61.5	86.2	79.3	68.6	84.5	62.4	73.8
FedBN [ICLR20] [14]	64.2	50.5	72.6	57.8	61.3	63.9	28.1	50.0	44.3	52.7	53.2	48.7	62.7	85.9	78.2	69.5	84.4	61.9	73.8
FedRobust [NeurIPS20] [15]	64.9	53.0	73.2	58.1	62.3	63.5	28.5	49.8	44.6	53.5	56.7	49.4	62.4	87.2	78.0	77.1	88.0	65.3	76.3
FedMix [ICLR21] [28]	65.1	52.6	73.8	58.9	62.6	63.3	28.0	50.1	45.9	53.3	56.8	49.6	62.1	86.7	78.1	76.8	87.7	65.6	76.2
FedDG [CVPR21] [22]			–						–				63.5	88.0	75.4	78.5	89.5	68.0	77.1
CCST [WACV23] [21]	63.0	50.1	73.0	61.7	62.0	62.1	27.7	49.6	46.0	56.6	60.5	50.3	64.0	87.8	78.4	78.8	89.0	68.4	77.7
FedSR [NeurIPS22] [19]	65.8	53.2	75.6	60.8	63.9	63.7	28.5	49.6	46.5	55.6	60.3	50.7	63.5	88.3	77.6	78.6	88.5	67.1	77.3
FPL [CVPR23] [23]	65.5	54.5	77.3	58.9	64.1	63.5	28.5	48.9	46.5	56.6	60.5	50.8	63.7	87.5	78.0	78.5	88.3	66.3	77.1
FedGA [CVPR23] [20]	66.2	54.2	78.6	59.0	64.5	65.0	28.0	49.4	46.9	56.3	61.0	51.1	64.8	88.0	79.6	77.5	88.0	67.8	77.6
FedFA^l [ICLR23] [1]	65.6	54.2	78.1	59.3	64.3	64.1	28.8	49.4	47.5	56.6	61.0	51.2	64.0	88.3	75.9	79.0	89.1	68.8	77.5
FedFA+	66.8	54.9	79.0	59.7	**65.1**	64.6	29.1	49.8	48.1	56.9	61.3	**51.6**	64.5	88.5	78.2	78.6	89.0	69.2	**78.0**
FedFA+ (Swin-T)	68.0	55.9	79.7	60.3	**66.0**	65.4	29.5	50.3	49.3	57.4	61.8	**52.3**	65.3	89.0	79.7	79.5	89.6	69.9	**78.8**

TABLE 7
Analysis of essential components in FedFA+.

Variant	Office [96]	DomainNet [97]	ProstateMRI [98]	CIFAR-10 [99]		Tiny-ImageNet [40]		FEMNIST [38]
				Dir (0.6)	Dir (0.3)	Dir (0.6)	Dir (0.3)	
FedAvg [5]	78.5	62.8	86.5	73.3	69.2	33.9	31.8	78.5
FedFA^l	83.1	66.5	89.5	76.3	71.9	36.5	34.9	79.6
FedFA^h	78.9	63.0	87.0	74.2	69.9	35.0	33.1	79.0
FedFA+	**84.8**	**67.5**	**90.2**	**77.3**	**72.6**	38.8	36.2	**80.5**
FedAvg (Swin-T) [5]	80.1	64.1	87.5	77.1	72.8	35.0	32.7	79.8
FedFA^l (Swin-T)	84.5	67.7	90.6	80.1	75.1	37.3	35.4	81.1
FedFA^h (Swin-T)	81.1	64.5	88.1	78.0	73.6	35.8	34.0	80.4
FedFA+ (Swin-T)	**85.9**	**69.3**	**91.3**	**81.6**	**76.1**	39.6	37.1	**81.6**

TABLE 8
Impacts of different normalization methods in FedFA+.

Normalization	Office [96]	DomainNet [97]	ProstateMRI [98]
GroupNorm	82.6	66.0	88.6
LayerNorm	82.8	66.7	89.1
BatchNorm (*default*)	83.1	66.5	89.5

TABLE 9
Efficacy of FedFA^l against FedFA-C and FedFA-R.

Variant	Office [96]	DomainNet [97]	ProstateMRI [98]
FedAvg [5]	78.5	62.8	86.5
FedFA-R	78.6	61.0	86.1
FedFA-C	79.5	63.7	87.8
FedFA^l	**83.1**	**66.5**	**89.5**

TABLE 10
Efficacy of FedFA^l against conventional augmentation techniques.

Algorithm	Office [96]	DomainNet [97]	ProstateMRI [98]
FedAvg [5]	78.5	62.8	86.5
Mixup [52]	79.2	63.4	87.0
M-Mixup [67]	79.6	63.5	87.6
MixStyle [34]	79.9	64.1	88.5
MoEx [35]	80.2	64.6	88.3
DAC-SC [108]	82.1	65.2	88.9
FedFA^l	**83.1**	**66.5**	**89.5**
FedFA^l+Mixup [52]	83.7	67.0	89.9
FedFA^l+M-Mixup [67]	83.6	66.9	90.2
FedFA^l+MixStyle [34]	84.0	67.2	90.2
FedFA^l+MoEx [35]	83.9	67.0	90.1
FedFA^l+DAC-SC [108]	84.5	67.7	90.8

TABLE 11
Analysis of different fusion strategies.

Variant	Office [96]	DomainNet [97]	ProstateMRI [98]
Direct Fusion	80.6	64.1	86.9
Adaptive Fusion	83.1	66.5	89.5

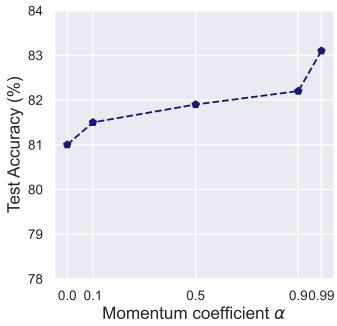
and FedFA+ improve the score to 0.25 and 0.21, respectively. For *uniformity* (lower the better), we obtain a score of -1.89 for FedAvg, -2.14 for FedFA^l, and -2.38 for FedFA+, respectively. These quantitative results suggest that the proposed components lead to a feature distribution that is more uniform over the embedding space, where classes are better clustered and separated. We hypothesize that this improvement can be attributed to two factors: on one hand, our feature augmentation algorithm extends the scope of features, and potentially make the feature space more uniform. On the other hand, the feature alignment algorithm intuitively helps to align features across clients, thereby improving the alignment metric. Their collaboration begets to a more structured feature space.

5.4.2 Ablation Study of Federated Feature Augmentation

FedFA^l *vs.* FedFA-C and FedFA-R. We compare FedFA^l against the two baseline variants outlined in §5.1. These variants exclusively involve device-dependent augmentation of feature statistics, without explicitly considering any global information. Table 9 shows that, by randomly perturbing feature statistics, FedFA-R exhibits no improvements or even a decline in performance on DomainNet and ProstateMRI when compared to FedAvg. FedFA-C yields promising gains by incorporating client-specific fea-

TABLE 12
Impact of different sets of eligible layers to apply FFA. The network architecture used for each experiment is included in the bracket. 'R50': ResNet 50 [55]; 'Dense100': DenseNet 100 [109] with a growth rate of 12.

Variant	Office [96] (AlexNet)	DomainNet [97] (AlexNet)	DomainNet [97] (R50)	DomainNet [97] (Dense100)	ProstateMRI [98] (U-Net)
FedAvg	78.5	62.8	74.8	70.7	86.5
$\{1\}$	78.8	63.5	75.3	71.0	88.5
$\{1,2\}$	80.0	63.9	75.6	71.5	88.6
$\{1,2,3\}$	80.0	64.0	75.9	71.7	89.0
$\{1,2,3,4\}$	80.6	64.3	76.1	72.0	88.8
$\{1,2,3,4,5\}$	**83.1**	**66.5**	**77.2**	**73.4**	**89.5**
$\{2,3,4,5\}$	81.6	65.2	76.7	72.8	88.6
$\{1,2,4,5\}$	82.0	65.8	76.9	72.9	88.8
$\{3,4,5\}$	78.4	63.8	75.8	71.6	87.0
$\{4,5\}$	79.4	64.7	75.6	71.5	85.9
$\{5\}$	79.2	64.6	75.7	71.3	86.3
$\{1,5\}$	80.4	65.5	76.1	71.9	88.5
$\{2,3,4\}$	79.5	64.3	75.5	71.4	88.8
$\{2,3\}$	78.7	64.0	75.3	71.2	88.5
$\{3,4\}$	78.3	63.2	75.1	70.9	86.5
$\{3\}$	78.0	63.1	74.9	70.8	86.5



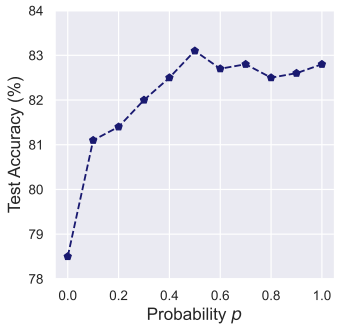
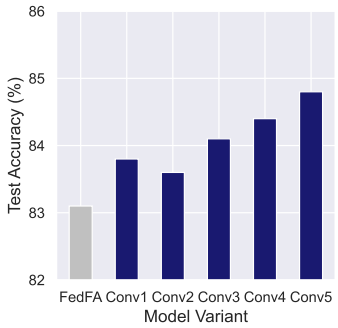


Fig. 5. Hyper-parameter analysis of α and p on Office-Caltech 10 [96].



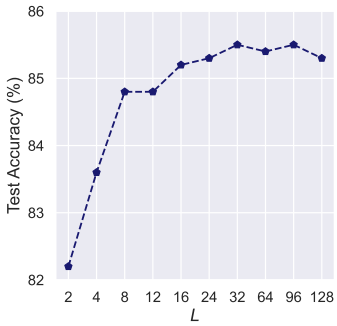


Fig. 6. Analysis of FedFA^h on Office-Caltech 10 [96]. Left: classification performance of different variants that perform alignment in different layers; Right: performance w.r.t. bin number L.

ture statistic variances; by comparing FedFA^l and FedFA-C, we affirm the significance of integrating universal feature statistic information in federated augmentation.

FedFA^l *vs.* Traditional Augmentation Methods. We compare FedFA^l with five data/feature augmentation techniques, *i.e.*, Mixup [52], Manifold Mixup [67], MixStyle [34] MoEx [35], and DAC-SC [108]. The results are presented in Table 10. We find that (i) all the five techniques exhibit notable improvements over FedAvg, and some of them (*e.g.*, MoEX, DAC-SC) even outperform well-designed federated learning algorithms (as compared to Tables 3-4). (ii) By accounting for global feature statistics, FedFA^l surpasses all of them, achieving **1.0%**/**0.5%**/**0.6%**/ improvements over the second-best results on Office/DomainNet/ProstateMRI, respectively. (iii) Combining FedFA^l with them allows for further performance uplifting, highlighting the complementary roles of FedFA^l to conventional augmentation techniques.

Adaptive variance fusion. To examine the impact of adaptive variance fusion in Eqs. 7-8, we introduce a baseline 'Direct Fusion' that directly combines client-specific and client-sharing statistic variances by $\hat{\Sigma}^2_{\mu^k_m} = (\Sigma^2_{\mu^k}+1)\Sigma^2_{\mu^k_m}$, $\hat{\Sigma}^2_{\sigma^k_m} = (\Sigma^2_{\sigma^k}+1)\Sigma^2_{\sigma^k_m}$. As presented in Table 11, the baseline shows severe degradation across all three benchmarks. One possible explanation is that the two types of variances are not properly aligned, and the simplistic fusion strategy may result in significant changes of client-specific statistic variances, which could be detrimental to local model learning.

Eligible layers for FFA. Next, we study the sensitivity of FedFA^l to the selection of eligible layers on which FFA is applied. We use '$\{1\}$' to denote that FFA is applied to the 1st convolutional stage; '$\{1,2\}$' to represent its application to both the 1st and 2nd convolutional stages, and so forth. In addition to lightweight CNNs (*e.g.*, AlexNet, U-Net), we study two heavy networks, *i.e.*, ResNet-50 [55] and DenseNet-100 [109], with DomainNet as the benchmark. All results are summarized in Table 12. We observe that (i) our default design (using five layers) consistently obtains the best performance across all cases, which we conjecture is due to its potential to beget more comprehensive augmentation; (ii) applying FFA to only one specific layer yields minor gains over FedAvg; (iii) as more layers are included for FFA, the performance tends to improve. This suggests that the model benefits from the complementary nature of features extracted from different layers.

Hyper-parameter analysis. FedFA^l involves two hyper-parameters, *i.e.*, momentum coefficient α in Eq. 5 and probability p to apply FFA during training. As shown in Fig. 5 (left), (i) the model is overall robust to α. It even yields promising results at $\alpha = 0$, in which the model only uses feature statistics of the last mini-batch in each local epoch to compute client-sharing statistic variances. This reveals that FedFA^l is insensitive to errors in client-sharing statistic vari-

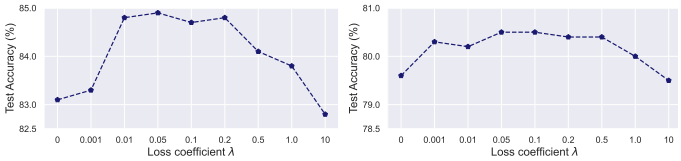


Fig. 7. Hyper-parameter analysis of λ (Eq. 15) on Office-Caltech 10 [96] (left) and FEMNIST [38] (right).

ances. It obtains the best score at $\alpha=0.99$ (ii) Regarding the probability p, as seen in Fig. 5 (right), FEDFA^l improves the baseline at $p=0$ significantly, even with a small probability (*e.g.*, $p=0.1$). The best performance is reached at $p=0.5$.

5.4.3 Ablation Study of Federated Feature Alignment

Eligible layers to apply $\mathcal{L}^{\text{ALIGN}}$. To gain a deeper understanding of FEDFA^h, we investigate the effects of applying alignment in different convolutional stages. Our analysis is carried out on Office-Caltech 10 [96] based on AlexNet (see Fig. 1). As depicted from Fig. 6 (left), we observe performance improvements regardless of the layers to apply alignment, and more notable gains are obtained by aligning in high-level layers ('Conv4', 'Conv5') compared to low-level layers (*i.e.*, 'Conv1', 'Conv2'). It is worth noting that simultaneous alignment in multiple layers is feasible, but will result in increased communication costs. Consequently, we opt to perform alignment solely at the last layer of the feature extractor, such as 'Conv5' in the case of AlexNet.

Bin number L. Next, we examine the impact of L. The results are shown in Fig. 6 (right). At $L=2$, FEDFA+ exhibits a performance degradation compared to FEDFA^l, suggesting that the approximation of the marginal feature distribution is inadequate. However, this can be alleviated by increasing L, which leads to progressive performance improvements. The performance is stable within the range $L\in[24,128]$. In our experiments, we set $L=8$ by default to strike a proper trade-off between performance and communication cost.

Loss coefficient λ. Last, we study the effect of λ in Eq. 15 on Office-Caltech 10 and FEMNIST. A diverse set of values is chosen, *i.e.*, $\{0, 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1.0, 10\}$. As seen from Fig. 7 , our algorithm's performance remains stable within the range $\lambda\in[0.01, 0.2]$. Notably, on Office-Caltech 10, the model obtains a minor improvement at $\lambda=0.001$; however, the performance sharply boosts upon increasing the value to 0.01. When further increasing the value of λ, we observe performance degradation, *e.g.*, the model even performs worse than FEDFA^l (*i.e.*, $\lambda=0$) at $\lambda=10$. On the larger-scale FEMNIST dataset, our algorithm shows even more stable performance compared to Office. We set $\lambda=0.1$ by default throughout all the experiments.

6 Discussion

6.1 Complexity Analysis

Memory cost. ***(i) Federated Feature Augmentation:*** compared to FEDAVG, FEDFA^l requires $4\sum_{k=1}^{K}C_k$ more GPU memory allocation to store four statistic values ($\bar{\mu}_m^k$, $\bar{\sigma}_m^k$, γ_{μ^k}, γ_{σ^k}) at each of the K FFA layers. Here C_k is the number of feature channel at each layer k. The costs are in practice very minor, *e.g.*, 18 KB/15.5 KB for AlexNet/U-Net. For comparison, FEDMIX requires $2\times$ more GPU memory than FEDAVG. ***(ii) Federated Feature Alignment:*** as stated in §3.3, FEDFA^h requires an extra memory with size LC_K. For AlexNet/U-Net, the costs are 8 KB/16 KB.

Communication cost. ***(i) Federated Feature Augmentation:*** In each round, FEDFA^l incurs additional communication costs since it requires the sending 1) *from client to server* the momentum feature statistics $\bar{\mu}_m^k$ and $\bar{\sigma}_m^k$, as well as 2) *from server to client* the client sharing feature statistic variances γ_{μ^k} and γ_{σ^k} at each layer k. Thus, for K layers in total, the extra communication cost for each client is $c_e=4\sum_{k=1}^{K}C_k$, where the factor of 4 is for server receiving and sending two statistic values. In our experiments, we append one FFA layer after each convolutional stage of feature extractors in AlexNet and U-Net. Hence, the incurred additional communication costs for AlexNet and U-Net are:

$$\text{AlexNet: } 4\times(64+192+384+256+256)/1024\times4=18\text{KB},$$
$$\text{U-Net: } 4\times(32+64+128+256+512)/1024\times4=15.5\text{KB}.$$

(ii) Federated Feature Alignment: similar with FEDFA^l, the alignment process necessitates transmissions of soft histograms. For AlexNet and U-Net, the actual costs are:

$$\text{AlexNet: } 2\times8\times256/1024\times4=16\text{KB},$$
$$\text{U-Net: } 2\times8\times512/1024\times4=32\text{KB}.$$

It should be noted that these additional costs are minor in comparison with the cost required for exchanging model parameters, which are 2×49.5 MB and 2×29.6 MB for AlexNet and U-Net, respectively. Hence, the extra communication bruden in FEDFA+ is almost negligible.

6.2 Privacy Issue in FEDFA+

We note that for modern neural networks with BN layers (which are common), FEDFA^l has a similar degree of privacy guarantees as most purely parameter-sharing methods (excepting for FEDBN [14] that intentionally keeps BN locally without any aggregation). Taking FEDAVG [5] as an example, in addition to aggregating model parameters, the server receives local BN statistics (*i.e.*, running mean and variance), computes their first moments (*i.e.*, mean) and then distributes them back to clients. FEDFA^l is similar in that each client's momentumed local statistics (*c.f.* Eq. 5) are sent to the server; but slightly different in that we calculate the second moments (*i.e.*, variance) as in Eq. 6. Note that there is no direct local data sharing across clients in FEDFA^l, making it privacy-aware superior to the counterpart FEDMIX [28] that exchanges averaged raw data across clients. In addition, FEDFA^h only involves the transmission of high-order feature statistics between clients and server, while avoiding the transfer of more sensitive raw features or prototypes as done in prior studies [23], [31], [32], [111]. As a result, it has better privacy-preserving capabilities. Overall, by focusing on statistical information, FEDFA+ strikes a favorable balance between improving model performance and ensuring data privacy in FL. We note that introducing Gaussian noise to feature statistics can potentially enhances privacy guarantees. Our preliminary results on Office-Caltech 10 indicate that FEDFA+ exhibits strong robustness when subjected to low levels of noise (standard variance ≤ 0.3), yielding

scores of 84.8%/84.7%/84.6%/84.6% for standard variances of 0/0.1/0.2/0.3, respectively. We posit that this robustness stems from the nature of FEDFA+ as a feature augmentation approach. Nevertheless, as the noise level increases, the model tends to degrade, *e.g.*, the score decreases to 83.7% at standard variance 0.4.

Moreover, we point out that privacy is difficult to quantify [112] and the notion itself is an active research topic in FL. For example, even FEDAVG suffers certain privacy issues, *i.e.*, an adversary (*e.g.*, one of the clients) can infer whether a sample belonging to other clients [113] or even precisely reconstruct their training data if gradients are shared [114], [115]. While our work centers on the algorithmic aspects, it is promising to further quantify the privacy-preserving capabilities of FEDFA+ in future research.

Furthermore, we highlight the trade-off between performance and privacy-preserving can be easily achieved in FEDFA^l/FEDFA+ for scenarios that have strict privacy requirements. By adjusting the number of FFA layers used in the network, one can strike a balance between accuracy, communication cost, and privacy risk. While using more FFA layers generally leads to higher accuracy (*c.f.* Table 12), it indeed increases information sharing, which could result in higher communication costs and a greater risk of data leakage. For scenarios with stringent privacy requirements, reducing the number of FFA layers can help maintain a reasonable balance between performance and privacy. As shown in Table 12, even a model variant with a reduced number of FFA layers can still achieve promising gains against FedAvg while minimizing the amount of shared feature statistics. For example, the model variant '{1,5}' (with only two FFA layers) can already lead to promising gains against FEDAVG, *i.e.*, $+1.9\%/2.7\%/2.0\%$ for Office/DomainNet/ProstateMRI. This adaptability makes FEDFA+ a versatile and privacy-aware solution for various target applications with diverse privacy constraints.

7 CONCLUSION

In this work, we present FEDFA+ for addressing heterogeneity in federated learning. It is unique in exploiting statistical information of latent features, and achieves the goal through two critical components: (i) federated feature augmentation (FEDFA^l), which models low-order feature statistics using Gaussian distribution to perform fedreation-aware feature augmentation, and (ii) federated feature alignment (FEDFA^h), which approximates latent features using higher-order statistics and aligns them across clients to reduce discrepancy between clients. We offer both theoretical and empirical justifications to understand and validate the approach. Moreover, FEDFA+ exhibits a remarkable ability to adapt to different privacy requirements and achieve a favorable balance between performance and privacy preservation. This allows it to cater to a wide range of real-world applications, making it a promising solution in the ever-evolving landscape of FL.

REFERENCES

[1] T. Zhou and E. Konukoglu, "FedFA: Federated feature augmentation," in *Int. Conf. Learn. Representations*, 2023.

[2] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273–1282.

[6] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[7] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[8] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving ai," *Communications of the ACM*, vol. 63, no. 12, pp. 33–36, 2020.

[9] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*, 2020, pp. 240–254.

[10] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Trans. on Vehicular Technology*, vol. 71, no. 8, pp. 8810–8824, 2022.

[11] X. Liang, Y. Liu, T. Chen, M. Liu, and Q. Yang, "Federated transfer reinforcement learning for autonomous driving," in *Federated and Transfer Learning*, 2022, pp. 357–371.

[12] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, 2022.

[13] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and S. Y. Philip, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Know. and Data Engineer.*, vol. 35, no. 8, pp. 8052–8072, 2022.

[14] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-iid features via local batch normalization," in *Int. Conf. Learn. Representations*, 2020.

[15] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," in *Advances Neural Inf. Process. Syst*, 2020, pp. 21 554–21 565.

[16] M. Jiang, Z. Wang, and Q. Dou, "Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images," in *AAAI Conference on Artificial Intelligence*, 2022, pp. 1087–1095.

[17] D. Caldarola, B. Caputo, and M. Ciccone, "Improving generalization in federated learning by seeking flat minima," in *Eur. Conf. Comput. Vis.*, 2022, pp. 654–672.

[18] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," in *Int. Conf. Mach. Learn.*, 2022, pp. 18 250–18 280.

[19] A. T. Nguyen, P. Torr, and S. N. Lim, "FedSR: A simple and effective domain generalization method for federated learning," in *Advances Neural Inf. Process. Syst*, 2022, pp. 38 831–38 843.

[20] R. Zhang, Q. Xu, J. Yao, Y. Zhang, Q. Tian, and Y. Wang, "Federated domain generalization with generalization adjustment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3954–3963.

[21] J. Chen, M. Jiang, Q. Dou, and Q. Chen, "Federated domain generalization for image recognition via cross-client style transfer," in *Proc. Winter Conf. Appli. Comput. Vis.*, 2023, pp. 361–370.

[22] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1013–1023.

[23] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking federated learning with domain shift: A prototype view," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16 312–16 322.

[24] R. Dai, X. Yang, Y. Sun, L. Shen, X. Tian, M. Wang, and Y. Zhang, "Fedgamma: Federated learning with global sharpness-aware minimization," *IEEE Trans. Neural Netw. Learning Sys.*, 2023.

[25] Y. Sun, L. Shen, S. Chen, L. Ding, and D. Tao, "Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape," in *Int. Conf. Mach. Learn.*, 2023, pp. 32 991–33 013.

[26] Y. Sun, L. Shen, T. Huang, L. Ding, and D. Tao, "Fedspeed: Larger local interval, less communication round, and higher generalization accuracy," in *Int. Conf. Learn. Representations*, 2023.

[27] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Int. Conf. Learn. Representations*, 2021.

[28] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "Fedmix: Approximation of mixup under mean augmented federated learning," in *Int. Conf. Learn. Representations*, 2021.

[29] T. Zhou and W. Wang, "Prototype-based semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.

[30] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances Neural Inf. Process. Syst*, 2017.

[31] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *AAAI Conference on Artificial Intelligence*, 2022, pp. 8432–8440.

[32] J. Xu, X. Tong, and S.-L. Huang, "Personalized federated learning with feature alignment and classifier collaboration," in *Int. Conf. Learn. Representations*, 2023.

[33] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1501–1510.

[34] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *Int. Conf. Learn. Representations*, 2021.

[35] B. Li, F. Wu, S.-N. Lim, S. Belongie, and K. Q. Weinberger, "On feature normalization and data augmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 383–12 392.

[36] Y. Yang, I. G. Morillo, and T. M. Hospedales, "Deep neural decision trees," *arXiv preprint arXiv:1806.06988*, 2018.

[37] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Machine learning proceedings 1995*, 1995, pp. 194–202.

[38] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *International Joint Conference on Neural Networks*, 2017, pp. 2921–2926.

[39] F. Lai, Y. Dai, S. Singapuram, J. Liu, X. Zhu, H. Madhyastha, and M. Chowdhury, "Fedscale: Benchmarking model and system performance of federated learning at scale," in *Int. Conf. Mach. Learn.*, 2022, pp. 11 814–11 827.

[40] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[41] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *ACM SIGSAC*, 2015, pp. 1310–1321.

[42] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[43] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Int. Conf. Learn. Representations*, 2021.

[44] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[45] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "Feddc: Federated learning with non-iid data via local drift decoupling and correction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 112–10 121.

[46] B. Li, M. N. Schmidt, T. S. Alstrøm, and S. U. Stich, "On the effectiveness of partial variance reduction in federated learning with heterogeneous data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3964–3973.

[47] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Int. Conf. Learn. Representations*, 2021.

[48] Y. Sun, L. Shen, H. Sun, L. Ding, and D. Tao, "Efficient federated learning via local adaptive amended optimizer with linear speedup," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 453–14 464, 2023.

[49] H.-Y. Chen and W.-L. Chao, "FedBE: Making bayesian model ensemble applicable to federated learning," in *International Conference on Learning Representations*, 2021.

[50] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *International Conference on Learning Representations*, 2020.

[51] S. Yu, J. Hong, H. Wang, Z. Wang, and J. Zhou, "Turning the curse of heterogeneity in federated learning into a blessing for out-of-distribution detection," in *Int. Conf. Learn. Representations*, 2023.

[52] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Int. Conf. Learn. Representations*, 2018.

[53] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *Int. Conf. Mach. Learn.*, 2013, pp. 552–560.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[56] W. Wang, C. Han, T. Zhou, and D. Liu, "Visual recognition with deep nearest centroids," in *Int. Conf. Learn. Representations*, 2022.

[57] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[58] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2582–2593.

[59] T. Zhou and W. Wang, "Cross-image pixel contrasting for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.

[60] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[61] T. Zhou, M. Zhang, F. Zhao, and J. Li, "Regional semantic contrast and aggregation for weakly supervised semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4299–4309.

[62] T. Zhou, Y. Yang, and W. Wang, "Differentiable multi-granularity human parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8296–8310, 2023.

[63] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," in *International conference on artificial neural networks*, 1996, pp. 47–52.

[64] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," *arXiv preprint arXiv:1710.10686*, 2017.

[65] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.

[66] Y. Liu, S. Yan, L. Leal-Taixé, J. Hays, and D. Ramanan, "Soft augmentation for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16 241–16 250.

[67] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.

[68] X. Li, Y. Dai, Y. Ge, J. Liu, Y. Shan, and L. DUAN, "Uncertainty modeling for out-of-distribution generalization," in *Int. Conf. Learn. Representations*, 2022.

[69] Z. Zhou, L. Qi, X. Yang, D. Ni, and Y. Shi, "Generalizable cross-modality medical image segmentation via style augmentation and dual normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20 856–20 865.

[70] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Int. Conf. Mach. Learn.*, 2013, pp. 10–18.

[71] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1521–1528.

[72] Y. Zhang, M. Li, R. Li, K. Jia, and L. Zhang, "Exact feature distribution matching for arbitrary style transfer and domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8035–8045.

[73] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in *Advances Neural Inf. Process. Syst*, 2011.

[74] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5400–5409.

[75] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5542–5550.

[76] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales, "A simple feature augmentation for domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8886–8895.

[77] Q. Liu, Q. Dou, and P. A. Heng, "Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains," in *Int. Conf. Med. Image Comput. Comput.- Assist. Interv*, 2020, pp. 475–485.

[78] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *Advances Neural Inf. Process. Syst*, 2018.

[79] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI Conference on Artificial Intelligence*, 2018.

[80] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *Advances Neural Inf. Process. Syst*, 2019.

[81] L. Zhang, Y. Luo, Y. Bai, B. Du, and L.-Y. Duan, "Federated learning for non-iid data via unified feature learning and optimization objective alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4420–4428.

[82] J. Yuan, X. Ma, D. Chen, F. Wu, L. Lin, and K. Kuang, "Collaborative semantic aggregation and calibration for federated domain generalization," *IEEE Trans. Know. and Data Engineer.*, vol. 35, no. 12, pp. 12 528–12 541, 2023.

[83] F. Yu, W. Zhang, Z. Qin, Z. Xu, D. Wang, C. Liu, Z. Tian, and X. Chen, "Fed2: Feature-aligned federated learning," in *Proceedings of ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2066–2074.

[84] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances Neural Inf. Process. Syst*, pp. 7611–7623, 2020.

[85] H. Yuan, W. R. Morningstar, L. Ning, and K. Singhal, "What do we mean by generalization in federated learning?" in *Int. Conf. Learn. Representations*, 2022.

[86] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," *Advances Neural Inf. Process. Syst*, 2000.

[87] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances Neural Inf. Process. Syst*, pp. 6256–6268, 2020.

[88] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[89] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[90] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[91] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.

[92] A. Camuto, M. Willetts, U. Simsekli, S. J. Roberts, and C. C. Holmes, "Explicit regularisation in gaussian noise injections," in *Advances Neural Inf. Process. Syst*, 2020, pp. 16 603–16 614.

[93] S. H. Lim, N. B. Erichson, F. Utrera, W. Xu, and M. W. Mahoney, "Noisy feature mixup," in *Int. Conf. Learn. Representations*, 2022.

[94] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 656–672.

[95] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Int. Conf. Mach. Learn.*, 2019, pp. 1310–1320.

[96] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[97] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1406–1415.

[98] Q. Liu, Q. Dou, L. Yu, and P. A. Heng, "Ms-net: multi-site network for improving prostate segmentation with heterogeneous mri data," *IEEE Trans. Medical imaging*, vol. 39, no. 9, pp. 2713–2724, 2020.

[99] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*. Technical report, University of Toronto, 2009.

[100] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[101] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[102] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[103] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.- Assist. Interv*, 2015, pp. 234–241.

[104] J. Kim, G. Kim, and B. Han, "Multi-level branched regularization for federated learning," in *Int. Conf. Mach. Learn.*, 2022, pp. 11 058–11 073.

[105] S. Seo, J. Kim, G. Kim, and B. Han, "Relaxed contrastive learning for federated learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 12 279–12 288.

[106] Y. Shi, J. Liang, W. Zhang, V. Tan, and S. Bai, "Towards understanding and mitigating dimensional collapse in heterogeneous federated learning," in *Int. Conf. Learn. Representations*, 2023.

[107] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022.

[108] S. Lee, J. Bae, and H. Y. Kim, "Decompose, adjust, compose: Effective normalization by playing with frequency for domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11 776–11 785.

[109] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[110] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Int. Conf. Mach. Learn.*, 2020, pp. 9929–9939.

[111] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 713–10 722.

[112] T. H. Rafi, F. A. Noor, T. Hussain, and D.-K. Chae, "Fairness and privacy-preserving in federated learning: A survey," *arXiv preprint arXiv:2306.08402*, 2023.

[113] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE symposium on security and privacy (SP)*, 2019, pp. 739–753.

[114] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances Neural Inf. Process. Syst*, 2019.

[115] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

Tianfei Zhou is currently a Professor with Department of Computer Science, Beijing Institute of Technology, China. Prior to that, he was a research fellow with Computer Vision Lab, ETH Zurich, Switzerland. He obtained his Ph.D. degree from Beijing Institute of Technology in 2017. His current research interests are mainly in the areas of computer vision, medical image analysis and machine learning. He was the recipient of MICCAI MEDIA Best Paper Award in 2022.

Ye Yuan received the B.S., M.S., and Ph.D. degrees in computer science from Northeastern University in 2004, 2007, and 2011, respectively. He is currently a Professor with the Department of Computer Science, Beijing Institute of Technology, China. He has more than 100 refereed publications in international journals and conferences, including VLDB Journal, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Knowledge and Data Engineering, SIGMOD, PVLDB, ICDE, IJCAI, WWW, and KDD. His research interests include graph embedding, graph neural networks, and social network analysis.

Binglu Wang received the Ph.D. degree in School of Automation at Northwestern Polytechnic University, Xi'an, China, in 2021. He is currently a Postdoc with the Radar Research Laboratory, Beijing Institute of Technology, China. His research interests include Computer Vision, Digital Signal Processing and Deep Learning.

Ender Konukoglu received the PhD degree in computer science specializing in medical image analysis from the Université de Nice and INRIA Sophia Antipolis, France in 2009. He was a research fellow at the Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital/Harvard Medical School. He is currently an Associate Professor at ETH Zurich, Switzerland. His research interests include medical image analysis, biophysical models, and machine learning.