A MODEL-HETEROGENEOUS FEDERATED LEARNING METHOD FOR MITIGATING DOMAIN SHIFT VIA VIRTUAL FEATURE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

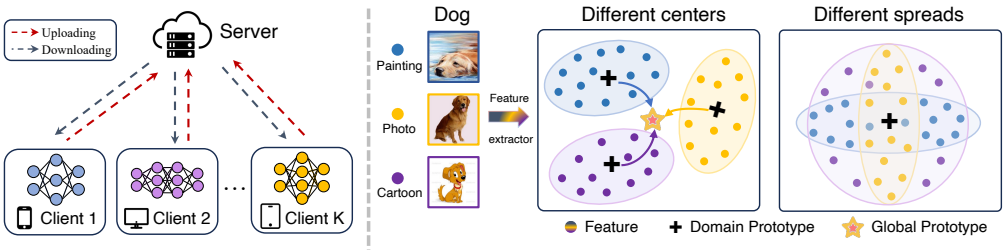
Federated learning (FL) becomes particularly challenging when model heterogeneity and domain shift occur simultaneously: heterogeneous architectures prevent direct parameter aggregation, while domain-dependent feature distributions hinder effective knowledge transfer across clients. Prototype-based approaches alleviate model heterogeneity by exchanging class-level feature summaries, yet a single prototype captures only the distribution center and overlooks domain-dependent variations in feature spread. To address these challenges, we propose **FedVFG**, a model-heterogeneous **Fed**erated Learning method for mitigating domain shift through **V**irtual **F**eature **G**eneration. Each client summarizes its local feature distributions using class prototypes and diagonal variances, which are scale-calibrated and aggregated by domain on the server. During local training, clients interpolate domain-specific class prototypes and standard deviations to construct virtual domain statistics, which guide the transformation of local features into class-conditioned virtual features. The original and virtual features are then jointly used to optimize the local models, enabling them to learn from cross-domain feature variations across heterogeneous client models. Experiments on Digit-5 and PACS under domain shift in the model heterogeneous setting show that FedVFG achieves state-of-the-art performance, outperforming the strongest baselines by 6.54% and 1.66% in average accuracy, respectively.

1 INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train models without directly sharing their raw data McMahan et al. (2017); Li et al. (2020); Ye et al. (2023). In each communication round, clients perform local training and upload model updates to a central server, which aggregates them to update the global model. By keeping data decentralized, FL reduces the exposure of raw data and avoids the cost of centralized data collection. It has therefore been widely studied and applied in scenarios such as mobile devices, healthcare, finance, and autonomous driving Jia et al. (2024); Liu et al. (2024); Chen et al. (2026); Kou et al. (2025).

Despite the widespread adoption of FL, practical deployments often involve two critical challenges: **I) Model heterogeneity.** When clients adopt different architectures with incompatible parameter dimensions, direct model aggregation becomes infeasible Diao et al. (2020), so collaboration must rely on an interface that is independent of the model architecture, as shown in Figure 1(a). Knowledge distillation provides an architecture-agnostic alternative, but often relies on public data or server-side generators Li & Wang (2019); Zhu et al. (2021). Prototype-based methods avoid these requirements by exchanging class prototypes instead of model parameters Tan et al. (2022); Zhang et al. (2024b). However, a single prototype captures only the center of a class distribution. **II) Domain shift.** Data collected from different environments or acquisition conditions can follow distinct distributions, causing models to learn domain-dependent features Zhou et al. (2023). In federated learning, such shifts are distributed across clients and can impede effective cross-domain knowledge sharing Li et al. (2021); Huang et al. (2023); Li et al. (2025); Nguyen et al. (2026). As illustrated in Figure 1(b), the same class can exhibit similar centers but different feature spreads across domains, while distinct domain-specific centers may also be obscured when collapsed into a single global prototype. Recent methods preserve richer domain structure through prototype cluster-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Model heterogeneity (b) Domain shift

Figure 1: **Motivation for model-heterogeneous federated learning under domain shift.** (a) Clients adopt heterogeneous model architectures. (b) The same class can exhibit different feature spreads or centers across domains. Colored dots denote local features, and black crosses denote domain-specific class prototypes. *Best viewed in color.*

ing or domain-aware alignment Wang et al. (2024); Chen et al. (2024); Le et al. (2026), but mainly use the resulting representatives as alignment targets rather than descriptions of cross-domain feature distributions. This raises a key question: *how can heterogeneous clients exploit cross-domain distributional variation through a collaboration interface independent of model parameters?*

To address these challenges, we propose FedVFG, a model-heterogeneous Federated Learning method for mitigating domain shift via virtual feature generation. Our goal is to enable heterogeneous clients to effectively exploit cross-domain knowledge while keeping model parameters local. Specifically, to enable knowledge sharing across heterogeneous model architectures, each client summarizes every observed class using its prototype and diagonal variance, and uploads these class statistics instead of model parameters. The server further calibrates and aggregates the uploaded statistics by domain to reduce feature-scale discrepancies across heterogeneous models while preserving domain-specific distributional information. Furthermore, to mitigate domain shift, each client interpolates domain-specific class prototypes and standard deviations to construct virtual domain statistics, which guide a location–scale transformation of local features into class-conditioned virtual features. These virtual features are jointly optimized with the original local features, allowing the local model to learn from cross-domain variations without requiring a public dataset or a separately trained generator. Our main contributions are as follows:

- We investigate model-heterogeneous federated learning under domain shift and develop a statistical collaboration mechanism that enables heterogeneous clients to share domain-specific distributional information through lightweight class statistics.
- We propose FedVFG, which generates class-conditioned virtual features from interpolated domain-specific class prototypes and standard deviations. This design enables heterogeneous local models to learn from cross-domain variations without requiring a public dataset or a separately trained generator.
- We conduct extensive experiments on Digit-5 and PACS under domain shift in the model-heterogeneous setting. Empirical results demonstrate that FedVFG achieves state-of-the-art performance, outperforming the strongest baselines by 6.54% and 1.66% in average accuracy, respectively. The ablation study further validates the effectiveness of its key design components.

2 Related Work

2.1 Model-Heterogeneous Federated Learning

Model-Heterogeneous Federated Learning (MH-FL) has emerged as a promising research direction for enabling clients with independently designed architectures to collaborate Ye et al. (2023). Existing work in this area can be broadly categorized into: 1) *knowledge distillation (KD)-based methods*, 2) *representation alignment methods*, and 3) *partial model sharing methods*. KD-based methods transfer knowledge through predictions or proxy data rather than directly aggregating model parameters, allowing differently structured models to learn from one another Li & Wang (2019); Zhu et al. (2021); Yao et al. (2023); Wang et al. (2023b); Zhang et al. (2023a); Gong et al. (2024); Wu et al.

(2025b); Yao et al. (2026); Wu et al. (2026). Representation alignment methods exchange feature representations, class prototypes, semantic anchors, or feature statistics to establish an architecture-agnostic interface for local models Tan et al. (2022); Zhang et al. (2024b;a); Zhou et al. (2025a); Niu et al. (2026). Partial model sharing methods retain compatible subnetworks or prediction components while allowing the remaining parts of client models to remain local Diao et al. (2020); Alam et al. (2022); Yi et al. (2023); Wang et al. (2023a); Yi et al. (2024); Wu et al. (2025a). However, existing representation-level methods mainly use shared features or prototypes as alignment or synthesis targets, rather than using domain statistics to transform local features. Our method follows the representation-level collaboration paradigm, but represents each domain–class pair with a domain-specific class prototype and a diagonal variance and uses these domain statistics to generate virtual features through an explicit affine transformation, without requiring a public dataset or a separately trained generator.

2.2 Federated Learning under Domain Shift

Federated learning under domain shift addresses the setting in which clients share a prediction task but observe different feature distributions. Beyond federated learning, feature statistics have been widely used to augment or align domain variations by manipulating feature means, variances, or other embedding statistics Tang et al. (2021); Zhou et al. (2021); Li et al. (2022); Ma et al. (2026). These studies highlight the effectiveness of feature-space statistics for modeling distributional changes across domains. Prior studies on federated learning under domain shift can be broadly categorized into three directions: 1) *domain-invariant representation learning*, 2) *cross-domain data or feature augmentation*, and 3) *domain-aware prototype or distribution sharing*. Domain-invariant methods preserve domain-specific normalization statistics or encourage invariant representations through regularization and cross-client alignment Li et al. (2021); Nguyen et al. (2022); Zhang et al. (2023b); Le et al. (2024); Zhou et al. (2025b); Wang et al. (2025); Nguyen et al. (2025); Liao et al. (2025); Pourpanah et al. (2025). Data- or feature-based augmentation methods increase domain diversity by transforming or mixing samples and feature statistics, including federated approaches that aggregate feature statistics for local diversification Huang & Belongie (2017); Liu et al. (2021); Zhou et al. (2021); Yan et al. (2025); Yang et al. (2024). Domain-aware prototype or distribution sharing methods preserve domain-specific information through prototypes or distribution summaries and use them for cross-domain knowledge transfer Huang et al. (2023); Wang et al. (2024); Le et al. (2025; 2026). However, these methods mainly use shared prototypes or statistics as alignment or augmentation targets. In contrast, FedVFG exploits domain-specific class statistics across heterogeneous clients to construct class-conditioned virtual features, enabling cross-domain knowledge transfer without requiring homogeneous model architectures.

3 Problem Formulation

The Global Objective of Federated Learning. Following the standard FL setting McMahan et al. (2017), we consider a federated system consisting of a central server and K clients distributed across M domains. Client k holds a private labeled dataset $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$, where $n_k = |\mathcal{D}_k|$, and belongs to a known domain $s(k) \in \{1, \dots, M\}$. All clients share the input space $\mathcal{X}$ and label space $\{1, \dots, C\}$. In the conventional FL setting, all clients optimize a common model parameter θ. Let $N = \sum_{k=1}^{K} n_k$ and $p_k = n_k/N$ denote the total number of samples and the sample proportion of client k, respectively. The global objective function can be expressed as:

$$\min_{\theta} \; F(\theta) = \sum_{k=1}^{K} p_k F_k(\theta), \qquad F_k(\theta) = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathcal{L}\big(f(x_{k,i}; \theta), y_{k,i}\big), \tag{1}$$

where $F_k(\theta)$ is the local objective on client k with loss function $\mathcal{L}$. However, this common-parameter assumption no longer holds under model heterogeneity, while domain shift makes client-wise risks distribution-specific.

Model Heterogeneity. Clients can adopt different model architectures to match their computational resources, resulting in potentially different parameter dimensions, *i.e.*, $\dim(\theta_k) \neq \dim(\theta_j), \exists k \neq j \in [K]$. Direct parameter averaging is therefore unavailable in general. Following

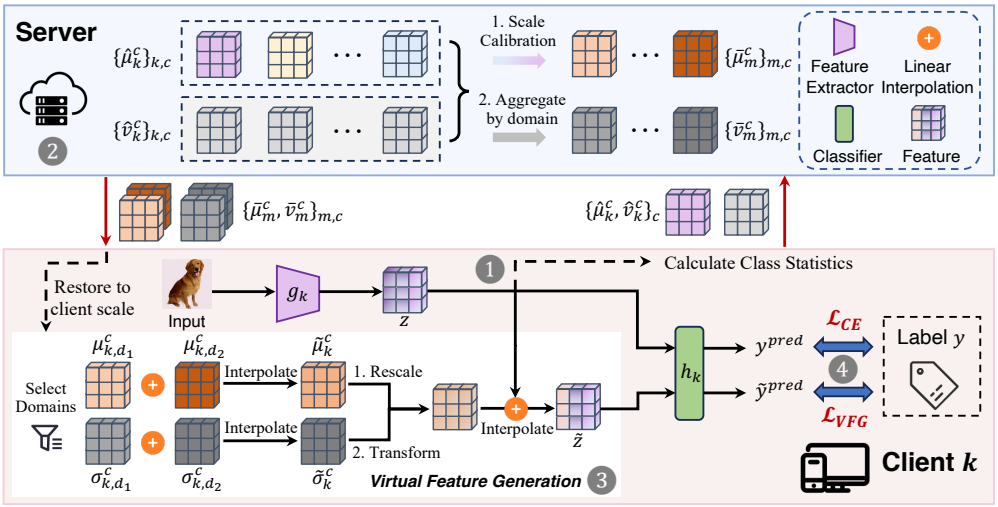


Figure 2: Overview of FedVFG. Domain-specific class statistics are calibrated and aggregated on the server and then used to generate class-conditioned virtual features for local training.

previous works Tan et al. (2022); Zhang et al. (2024b), we represent client k's local model as:

$$f_k(x;\theta_k) = h_k\big(g_k(x;\phi_k);\psi_k\big), \qquad \theta_k = (\phi_k, \psi_k), \tag{2}$$

where $g_k : \mathcal{X} \to \mathbb{R}^d$ and $h_k : \mathbb{R}^d \to \mathbb{R}^C$ denote the local *feature extractor* and *classifier*, respectively. The model heterogeneity problem in FL is thus to enable knowledge transfer among these heterogeneous models across all clients.

Domain Shift. Let $P_m(x \mid y = c)$ denote the class-conditional distribution of class c in domain m. Although all domains share the same label space, domain variations may induce $P_m(x \mid y = c) \neq P_{m'}(x \mid y = c)$ for $m \neq m'$. Clients may also exhibit different label proportions or observe only subsets of the classes. Such discrepancies can lead feature extractors to learn domain-dependent representations with different class prototypes and within-class dispersions. Our goal is to leverage cross-domain information to improve local prediction while keeping model parameters local.

4 Methodology

4.1 Overview of FedVFG

FedVFG enables heterogeneous clients to collaborate under domain shift by exchanging lightweight class statistics rather than model parameters, and further exploits the shared statistics to generate virtual features for local training. Specifically, each private model $f_k = h_k \circ g_k$ retains its own architecture, while cross-client knowledge is communicated through class-wise feature statistics. As shown in Figure 2, the training process in each communication round t can be summarized as follows:[1] ① **Local class statistics extraction.** Each participating client $k \in \mathcal{A}_t$ summarizes its updated representations using class-wise prototypes and diagonal variances $\{\widehat{\boldsymbol{\mu}}_k^c, \widehat{\boldsymbol{v}}_k^c\}_c$, which are uploaded to the server for cross-client collaboration (Section 4.2). ② **Scale calibration and domain aggregation.** To alleviate feature-scale discrepancies caused by heterogeneous models, the server calibrates each client's statistics with its RMS scale τ_k and aggregates them by domain and class, yielding the shared domain statistics $\bar{\mathcal{S}} = \{\bar{\boldsymbol{\mu}}_m^c, \bar{\boldsymbol{v}}_m^c\}_{m,c}$ (Section 4.3). ③ **Virtual feature generation.** After receiving $\bar{\mathcal{S}}$, client k restores the statistics to its own feature scale, selects a valid domain pair (d_1, d_2) with the largest support set $\mathcal{I}$, and interpolates their class-wise statistics to generate virtual features $\widetilde{\boldsymbol{z}}_i$ for supported local samples (Section 4.4). ④ **Local training.** Client k jointly optimizes its private feature extractor g_k and classifier h_k using the local objective $\mathcal{L}_k$. After local training, the client refreshes its class statistics for the next round $t+1$ (Section 4.5).

[1] Algorithm 1 summarizes the complete training procedure of FedVFG.

4.2 Local Class Statistics Extraction

To communicate both class centers and within-class variation, each client summarizes its features using class prototypes and diagonal variances. After local training, let $\mathcal{D}_k^c$ denote the n_k^c examples of class c on client k. For each observed class, the updated feature extractor produces:

$$\widehat{\boldsymbol{\mu}}_k^c = \frac{1}{n_k^c} \sum_{(x,y)\in\mathcal{D}_k^c} g_k(x), \tag{3}$$

$$\widehat{\boldsymbol{v}}_k^c = \frac{1}{n_k^c} \sum_{(x,y)\in\mathcal{D}_k^c} \left(g_k(x) - \widehat{\boldsymbol{\mu}}_k^c\right)^{\odot 2}, \tag{4}$$

where $\odot$ denotes element-wise operations. Class-wise moments are estimated online with a double-precision Welford algorithm Welford (1962), without storing individual features. Each client then uploads its class-wise prototypes and variances, *i.e.*, $\{\widehat{\boldsymbol{\mu}}_k^c, \widehat{\boldsymbol{v}}_k^c\}_c$, to the server for aggregation.

4.3 Scale Calibration and Domain Aggregation

Since heterogeneous feature extractors may produce features at substantially different scales, directly aggregating their statistics can bias the resulting domain representation toward clients with larger feature magnitudes. Before aggregation, the server therefore calibrates each upload using the root-mean-square (RMS) feature scale:

$$\tau_k = \left[\frac{1}{n_k}\sum_{c=1}^{C} n_k^c \left(\|\widehat{\boldsymbol{\mu}}_k^c\|_2^2 + \mathbf{1}^\top \widehat{\boldsymbol{v}}_k^c\right)\right]^{1/2}, \tag{5}$$

where $n_k = \sum_c n_k^c$. The calibrated statistics are defined as $\bar{\boldsymbol{\mu}}_k^c = \widehat{\boldsymbol{\mu}}_k^c/\tau_k$ and $\bar{\boldsymbol{v}}_k^c = \widehat{\boldsymbol{v}}_k^c/\tau_k^2$, while the client retains its original statistics.

The server maintains the latest calibrated statistics received from each client and updates them whenever new statistics are uploaded. Let $\mathcal{K}_m^c$ denote the set of clients in domain m that provide statistics for class c. For each nonempty $\mathcal{K}_m^c$, we define $w_{k,m}^c = n_k^c / \sum_{j\in\mathcal{K}_m^c} n_j^c$ and aggregate the domain-specific statistics as follows:

$$\bar{\boldsymbol{\mu}}_m^c = \sum_{k\in\mathcal{K}_m^c} w_{k,m}^c \bar{\boldsymbol{\mu}}_k^c, \tag{6}$$

$$\bar{\boldsymbol{v}}_m^c = \sum_{k\in\mathcal{K}_m^c} w_{k,m}^c \left[\bar{\boldsymbol{v}}_k^c + \left(\bar{\boldsymbol{\mu}}_k^c - \bar{\boldsymbol{\mu}}_m^c\right)^{\odot 2}\right]. \tag{7}$$

where the domain variance captures both within-client variation and between-client prototype displacement. The server then broadcasts the aggregated domain statistics $\bar{\mathcal{S}} = \{\bar{\boldsymbol{\mu}}_m^c, \bar{\boldsymbol{v}}_m^c\}_{m,c}$ to active clients $k \in \mathcal{A}_t$ for all valid domain–class pairs.

4.4 Virtual Feature Generation

To augment local training with cross-domain variation, we use the received domain statistics to transform local features. Client k first computes τ_k from its retained local statistics using Eq. 5, then restores the domain prototypes and variances as $\boldsymbol{\mu}_{k,m}^c = \tau_k \bar{\boldsymbol{\mu}}_m^c$ and $\boldsymbol{v}_{k,m}^c = \tau_k^2 \bar{\boldsymbol{v}}_m^c$. During local training, each client uses the restored domain statistics to generate virtual features for supported samples. Given a mini-batch $\mathcal{B}_k$, client k first computes local features $\boldsymbol{z}_i = g_k(x_i)$. A domain pair is valid for a sample only if both domains provide statistics for its class. For each pair $d_1 < d_2$, we therefore define the supported index set as:

$$\mathcal{I}_{d_1,d_2} = \left\{ i \in \mathcal{B}_k : (\boldsymbol{\mu}_{k,d_1}^{y_i}, \boldsymbol{v}_{k,d_1}^{y_i}) \text{ and } (\boldsymbol{\mu}_{k,d_2}^{y_i}, \boldsymbol{v}_{k,d_2}^{y_i}) \text{ are available} \right\}. \tag{8}$$

We select the domain pair that maximizes $|\mathcal{I}_{d_1,d_2}|$, breaking ties uniformly at random, and denote the resulting support set as $\mathcal{I}$. This maximizes the number of samples eligible for virtual feature generation in the current mini-batch.

Next, we sample a shared interpolation coefficient $\alpha \sim \mathcal{U}(0,1)$ for all supported examples. Let $\widehat{\boldsymbol{\sigma}}_k^c = \sqrt{\widehat{\boldsymbol{v}}_k^c}$ and $\boldsymbol{\sigma}_{k,m}^c = \sqrt{\boldsymbol{v}_{k,m}^c}$ denote the local and restored domain standard deviations, respectively. For each supported class, we then interpolate the prototypes and standard deviations of the selected domains as:

$$\widetilde{\boldsymbol{\mu}}_k^c = (1-\alpha)\boldsymbol{\mu}_{k,d_1}^c + \alpha\boldsymbol{\mu}_{k,d_2}^c, \tag{9}$$

$$\widetilde{\boldsymbol{\sigma}}_k^c = (1-\alpha)\boldsymbol{\sigma}_{k,d_1}^c + \alpha\boldsymbol{\sigma}_{k,d_2}^c. \tag{10}$$

The virtual prototype $\widetilde{\boldsymbol{\mu}}_k^c$ specifies the target class center, while $\widetilde{\boldsymbol{\sigma}}_k^c$ specifies the target spread. Although these interpolated statistics characterize a virtual domain, they do not directly provide sample-level features. We therefore transform each local feature toward the corresponding virtual distribution while preserving its sample-specific residual. To avoid unstable changes caused by large differences between local and target spreads, we first define the bounded rescaling factor:

$$\boldsymbol{s}_k^c = \mathrm{clip}\left(\frac{\widetilde{\boldsymbol{\sigma}}_k^c}{\widehat{\boldsymbol{\sigma}}_k^c}, s_{\min}, s_{\max}\right), \tag{11}$$

where $s_{\min}$ and $s_{\max}$ bound the coordinate-wise scaling. Each local feature is then rescaled around its class prototype and recentered at the virtual prototype:

$$T_k^c(\boldsymbol{z}_i) = \widetilde{\boldsymbol{\mu}}_k^c + \boldsymbol{s}_k^c \odot \left(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_k^c\right). \tag{12}$$

This location–scale transformation preserves the sample-specific residual while adapting the feature center and spread to the virtual domain statistics, following the principle of adaptive instance normalization Huang & Belongie (2017).

Directly replacing the local feature with $T_k^c(\boldsymbol{z}_i)$ may, however, introduce an overly strong domain shift. We therefore control the transformation strength by independently sampling $\rho_i \sim \mathcal{U}(0, \rho_{\max})$ for each $i \in \mathcal{I}$ and interpolating between the local and transformed features:

$$\widetilde{\boldsymbol{z}}_i = (1-\rho_i)\boldsymbol{z}_i + \rho_i T_k^{y_i}(\boldsymbol{z}_i). \tag{13}$$

Here, $\rho_{\max}$ controls the maximum transformation strength, allowing virtual features to explore different degrees of cross-domain variation while retaining the original label y_i.

4.5 Local Training with Virtual Features

The generated virtual features retain the labels of their corresponding local samples and are fed into the same classifier h_k for joint optimization. Given a mini-batch $\mathcal{B}_k$, we first apply the standard cross-entropy loss to the original local features:

$$\mathcal{L}_{\mathrm{CE}}^k = \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} \ell_{\mathrm{CE}}\big(h_k(\boldsymbol{z}_i), y_i\big). \tag{14}$$

This loss provides the basic supervised signal and anchors the local model to the observed data distribution. For samples supported by the selected domain pair, we further impose classification supervision on their virtual features:

$$\mathcal{L}_{\mathrm{VFG}}^k = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \ell_{\mathrm{CE}}\big(h_k(\widetilde{\boldsymbol{z}}_i), y_i\big). \tag{15}$$

By enforcing the same semantic labels on features generated from cross-domain statistics, this loss encourages the model to remain discriminative under domain-induced feature variation. Finally, the local objective combines the two terms as:

$$\mathcal{L}_k = \mathcal{L}_{\mathrm{CE}}^k + \lambda\mathcal{L}_{\mathrm{VFG}}^k, \tag{16}$$

where $\lambda \geq 0$ controls the strength of virtual-feature supervision. If $\mathcal{I} = \varnothing$, we set $\mathcal{L}_{\mathrm{VFG}}^k = 0$, reducing the objective to standard local cross-entropy training. During training, the virtual prototypes, target standard deviations, and rescaling factors are treated as constants, while the generated virtual features remain differentiable with respect to the local representations:

$$\frac{\partial\widetilde{\boldsymbol{z}}_i}{\partial\boldsymbol{z}_i} = \mathrm{diag}\left((1-\rho_i)\mathbf{1} + \rho_i\boldsymbol{s}_k^{y_i}\right). \tag{17}$$

Therefore, gradients from $\mathcal{L}_{\mathrm{VFG}}^k$ propagate through both the classifier and the feature extractor, allowing cross-domain variation to directly shape the learned representation. After local training, each client recomputes its class statistics and uploads them for the next server update.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate our proposed FedVFG on two multi-domain datasets: Digit-5 Zhou et al. (2020b) and PACS Zhou et al. (2020a). Digit-5 comprises five digit domains: MNIST, MNIST-M, SVHN, USPS, and Synthetic Digits, while PACS consists of four style domains: Photo, Art Painting, Cartoon, and Sketch. Each domain contains four clients, giving 20 and 16 clients in total. For both datasets, we adopt a 4:1 ratio to split samples into training and test sets within each domain. Following the conventional data partition setting Huang et al. (2023), each client trains on 1% of its domain's training pool for Digit-5 and 10% for PACS based on the complexity and scale of the datasets. For each class c in domain m, the allocation vector over its clients is sampled as $\mathbf{p}_m^c \sim \mathrm{Dir}(\boldsymbol{\beta})$, where a smaller $\boldsymbol{\beta}$ implies more extreme heterogeneity. We set $\boldsymbol{\beta} = 1.0$ for Digit-5 and $\boldsymbol{\beta} = 0.5$ for PACS, respectively.

Models. Following previous studies Zhang et al. (2024b); Zhou et al. (2025a), we adopt four heterogeneous architectures: 4-layer CNN, GoogLeNet Szegedy et al. (2015), MobileNetV2 Sandler et al. (2018), and ResNet-18 He et al. (2016). These architectures are distributed across clients in a round-robin manner, so that client k takes the one indexed by $k \bmod 4$. More results with homogeneous model architectures are reported in Appendix G.

Comparison Baselines. In the heterogeneous model experiments, we comprehensively evaluate our method against seven FL baselines that require no public data, including FedProto Tan et al. (2022), FPL Huang et al. (2023), FedTGP Zhang et al. (2024b), FedPLVM Wang et al. (2024), FedSA Zhou et al. (2025a), FedVTC Niu et al. (2026), FedDAP Le et al. (2026). Moreover, we also compare a standalone baseline where clients train locally without any communication. Methods that do not natively support model heterogeneity are adapted by removing model-parameter aggregation. More details are provided in Appendix B.

Implementation Details. All methods are implemented in PyTorch Paszke et al. (2019) with NVIDIA RTX 3090 GPU. We optimize all models with SGD using a learning rate of 0.001, momentum of 0.9 and a weight decay of $1e-5$. The batch size is set at 64 for all datasets. Training runs for 100 communication rounds with 100% client participation and 5 local epochs per round. The standardized feature dimension d is set to 512. For our method, $\lambda = 10$, $\rho_{\max} = 1$, and $[s_{\min}, s_{\max}] = [0.5, 5.0]$ on both datasets. Details regarding hyperparameter settings are provided in Appendix D.

Evaluation Metrics. We use the Top-1 accuracy averaged over all domains as the evaluation metric. Each experiment is run three times with different random seeds, and we report the mean $\pm$ standard deviation of the results.

5.2 PERFORMANCE COMPARISON

Tables 1 and 2 compare FedVFG with state-of-the-art methods in the model-heterogeneous setting. Overall, FedVFG consistently achieves the best average accuracy on both datasets. On Digit-5, FedVFG reaches 50.19%, outperforming the strongest baseline FedVTC by 6.54%, and achieves the best performance across all five domains. On PACS, FedVFG obtains an average accuracy of 35.00%, exceeding the strongest baseline FedVTC by 1.66%, while ranking first on Cartoon and Sketch and second on Photo and Art Painting. Compared with standalone training, FedVFG further improves the average accuracy by 22.30% and 8.15% on Digit-5 and PACS, respectively. These results demonstrate the effectiveness of leveraging domain-specific class statistics and virtual feature generation to facilitate knowledge collaboration among heterogeneous clients under domain shift.

5.3 ABLATION STUDY

We investigate four variants of FedVFG to evaluate the contribution of its key designs, as summarized in Table 3. Removing the VFG loss leads to the largest performance degradation, reducing the average accuracy from 50.19% to 34.21% on Digit-5 and from 35.00% to 30.57% on PACS,

Table 1: Performance comparison (%) on Digit-5 in the heterogeneous setting. Avg reports the mean accuracy across all domains, and Δ denotes its improvement over Standalone. Best in **bold** and second with underline.

Methods	Digit-5						
	MNIST-M	MNIST	SYN	USPS	SVHN	Avg	Δ
Standalone	14.92 ± 0.09	46.96 ± 0.41	12.99 ± 0.47	51.42 ± 1.32	13.17 ± 0.16	27.89	–
FedProto [AAAI22]	17.33 ± 0.47	43.32 ± 1.67	18.59 ± 1.15	51.24 ± 0.90	12.10 ± 0.34	28.52	+0.63
FPL [CVPR23]	23.72 ± 0.22	56.17 ± 0.46	25.28 ± 0.08	59.19 ± 1.17	14.85 ± 0.80	35.84	+7.95
FedTGP [AAAI24]	22.95 ± 0.18	63.31 ± 1.42	26.56 ± 0.50	61.64 ± 1.38	14.20 ± 0.32	37.73	+9.84
FedPLVM [NIPS24]	24.35 ± 0.18	64.97 ± 1.37	24.73 ± 1.45	63.57 ± 1.07	15.83 ± 0.90	38.69	+10.80
FedSA [AAAI25]	28.57 ± 0.37	70.29 ± 0.77	29.17 ± 0.16	67.44 ± 0.85	21.78 ± 0.68	43.45	+15.56
FedDAP [CVPR26]	25.30 ± 0.89	62.88 ± 0.24	28.16 ± 0.26	61.96 ± 1.28	17.30 ± 0.83	39.12	+11.23
FedVTC [ICLR26]	29.80 ± 0.54	70.34 ± 1.24	28.81 ± 0.89	71.79 ± 0.59	17.51 ± 0.54	43.65	+15.76
FedVFG (Ours)	**36.71 ± 3.41**	**79.16 ± 0.66**	**37.56 ± 0.86**	**74.23 ± 1.71**	**23.28 ± 1.49**	**50.19**	**+22.30**

Table 2: Performance comparison (%) on PACS in the heterogeneous setting. Avg reports the mean accuracy across domains, and Δ denotes its improvement over Standalone. Best in **bold** and second with underline.

Methods	PACS					
	Photo	Art painting	Cartoon	Sketch	Avg	Δ
Standalone	26.97 ± 0.85	20.35 ± 0.42	27.01 ± 0.60	33.07 ± 0.14	26.85	–
FedProto [AAAI22]	26.82 ± 0.78	18.46 ± 1.08	28.18 ± 0.13	36.71 ± 1.08	27.54	+0.69
FPL [CVPR23]	29.54 ± 0.66	20.98 ± 0.13	32.21 ± 0.52	42.74 ± 0.60	31.37	+4.52
FedTGP [AAAI24]	31.36 ± 1.11	20.35 ± 0.80	33.28 ± 0.74	44.10 ± 0.79	32.27	+5.42
FedPLVM [NIPS24]	29.49 ± 2.51	**21.75 ± 0.75**	29.67 ± 1.14	38.93 ± 0.40	29.96	+3.11
FedSA [AAAI25]	31.61 ± 0.49	20.59 ± 0.40	34.74 ± 0.26	45.48 ± 0.31	33.11	+6.26
FedDAP [CVPR26]	29.72 ± 0.53	20.79 ± 0.40	33.07 ± 0.24	44.38 ± 0.34	31.99	+5.14
FedVTC [ICLR26]	**34.18 ± 1.26**	20.48 ± 0.56	35.27 ± 0.40	43.42 ± 0.52	33.34	+6.49
FedVFG (Ours)	34.08 ± 0.93	21.61 ± 1.00	**35.93 ± 0.57**	**48.37 ± 0.08**	**35.00**	**+8.15**

demonstrating the importance of virtual-feature supervision. Replacing the statistic-guided transformation with random perturbations also causes clear drops to 44.84% and 33.45%, indicating that the improvement does not simply come from generic feature augmentation.

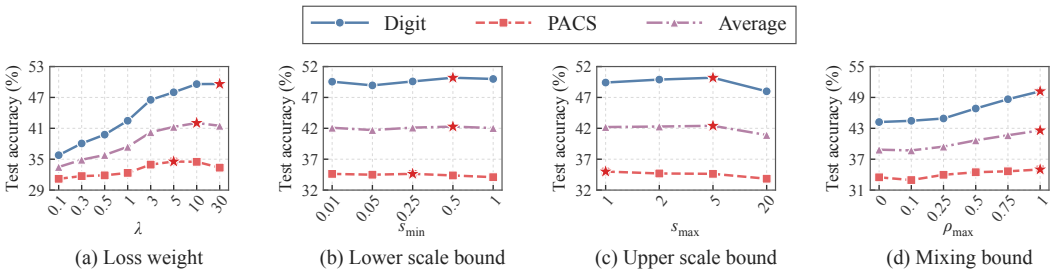
When domain-specific statistics are replaced with globally pooled statistics, the performance decreases to 48.95% and 34.47%, confirming the benefit of preserving domain-dependent class distributions. Finally, using only mean shifts achieves 49.04% and 34.57%, while the full method improves the results to 50.19% and 35.00%, showing that modeling both class centers and feature spreads provides additional gains over the mean-shift-only transformation.

Table 3: Ablation study of FedVFG in the heterogeneous setting.

Variant	Digit-5	PACS
w/o VFG loss	34.21 ± 0.42	30.57 ± 0.08
Random perturbation	44.84 ± 0.38	33.45 ± 0.44
Global class statistics	48.95 ± 0.74	34.47 ± 0.17
Mean shift only	49.04 ± 0.08	34.57 ± 0.38
FedVFG (Ours)	**50.19 ± 0.93**	**35.00 ± 0.03**

5.4 Hyperparameter Sensitivity

To investigate the sensitivity of FedVFG to different hyperparameters, we vary the virtual-feature loss weight λ, the lower and upper scale bounds $s_{\min}$ and $s_{\max}$, and the mixing bound $\rho_{\max}$ on Digit-5 and PACS. As shown in Figure 3, increasing λ generally improves performance at first, while an excessively large value leads to degradation, especially on PACS. In comparison, FedVFG is relatively insensitive to $s_{\min}$ over a broad range. For $s_{\max}$, both the Digit-5 accuracy and the average



(a) Loss weight (b) Lower scale bound (c) Upper scale bound (d) Mixing bound

Figure 3: Hyperparameter sensitivity on Digit-5 and PACS: (a) virtual-feature loss weight λ, (b) lower rescaling bound $s_{\min}$, (c) upper rescaling bound $s_{\max}$, and (d) mixing bound $\rho_{\max}$. "Average" denotes the mean accuracy across the two datasets.

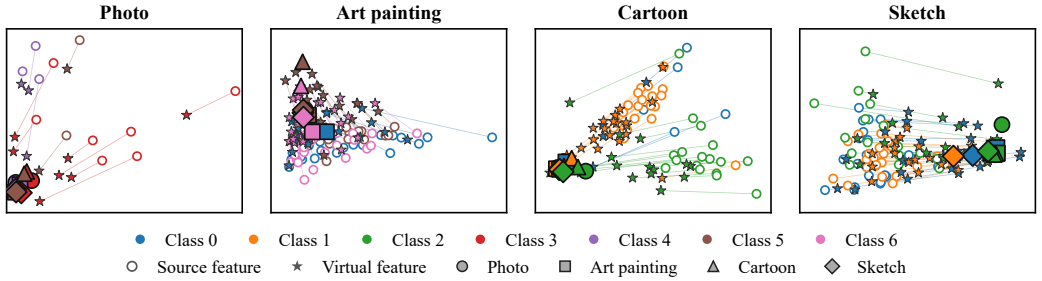


Figure 4: Visualization of virtual feature generation on PACS. (○) denote source features and (★) denote their generated virtual features. Colors indicate classes, and line segments connect corresponding feature pairs. *Best viewed in color.*

performance peak around $s_{\max} = 5$, whereas a much larger bound degrades the results. Finally, increasing $\rho_{\max}$ consistently improves the overall performance, with the best results obtained at $\rho_{\max} = 1$, demonstrating the benefit of sufficiently incorporating the generated virtual features. Overall, FedVFG remains stable over moderate variations of the scale bounds, while λ and $\rho_{\max}$ have a more pronounced influence on performance.

5.5 Visualization of Virtual Feature Generation

Figure 4 visualizes the source features and their generated virtual features on PACS. For visualization, we select the trained 4-layer CNN client from each domain, sample its local source features, regenerate the corresponding virtual features using the final domain statistics, and project both into a client-specific 2D PCA space Jolliffe & Cadima (2016). Within each client-specific PCA space, the source-to-virtual trajectories vary in both direction and magnitude across classes, indicating that FedVFG does not apply a uniform perturbation to local representations. Instead, the transformation is guided by class-specific cross-domain prototypes and variances, introducing structured and class-conditioned feature variation while preserving semantic labels. The corresponding visualization on Digit-5 is provided in Appendix F.

6 Conclusion

In this paper, we propose FedVFG, a model-heterogeneous federated learning method for mitigating domain shift via virtual feature generation. FedVFG enables heterogeneous clients to share domain-specific distributional information through lightweight class statistics while keeping model parameters local. By calibrating and aggregating class prototypes and diagonal variances by domain, FedVFG constructs domain-specific statistics and uses their interpolation to generate class-conditioned virtual features for local training, without requiring a public dataset or a separately trained generator. Experiments on Digit-5 and PACS under domain shift in the model-heterogeneous setting demonstrate state-of-the-art average accuracy, outperforming the strongest baselines by 6.54% and 1.66%, respectively. Ablation studies further validate the effectiveness of the key design components.

AI USE STATEMENT

In this work, we used generative AI tools for research ideation, language polishing, and code assistance. These tools were used to support brainstorming of research directions and implementation alternatives, improve the clarity and grammar of author-written text, and assist with code drafting and debugging. The authors independently evaluated and selected the research ideas, finalized the problem formulation and method design, conducted the experiments, and analyzed the results. All AI-assisted content was carefully reviewed and verified by the authors, including technical descriptions, experimental implementations, and reported results. We take full responsibility for the final content of this work, including any text, claims, code, or other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including PACS and Digit-5, were sourced in accordance with their relevant usage guidelines. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and the study does not involve additional collection or sharing of raw user data. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We describe the datasets, client partitions, model architectures, training setting, and hyperparameters in Section 5.1 and Appendix D. The complete training procedure is given in Appendix A. We provide anonymized source code and instructions for reproducing the experiments as supplementary material.

REFERENCES

Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35:29677–29690, 2022.

Dongbin Chen, Shan Chang, Minghui Dai, Denghui Li, and Haoliang Zhao. Bridging data silos in finance via federated learning. *IEEE Network*, 40(3):246–253, 2026. doi: 10.1109/MNET.2025.3584806.

Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12077–12086, 2024.

Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

Xuan Gong, Shanglin Li, Yuxiang Bao, Barry Yao, Yawen Huang, Ziyan Wu, Baochang Zhang, Yefeng Zheng, and David Doermann. Federated learning via input-output collaborative distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 22058–22066, 2024.

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

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

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1501–1510, 2017.

Yongzhe Jia, Xuyun Zhang, Hongsheng Hu, Kim-Kwang Raymond Choo, Lianyong Qi, Xiaolong Xu, Amin Beheshti, and Wanchun Dou. Dapperfl: Domain adaptive federated learning with model fusion pruning for edge devices. *Advances in Neural Information Processing Systems*, 37: 13099–13123, 2024.

Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 374(2065):20150202, 2016.

Wei-Bin Kou, Qingfeng Lin, Ming Tang, Sheng Xu, Rongguang Ye, Yang Leng, Shuai Wang, Guofa Li, Zhenyu Chen, Guangxu Zhu, and Yik-Chung Wu. pfedlvm: A large vision model (lvm)-driven and latent feature-based personalized federated learning framework in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 26(10):15915–15931, 2025. doi: 10.1109/TITS.2025.3581939.

Huy Q Le, Ye Lin Tun, Yu Qiao, Minh NH Nguyen, Keon Oh Kim, Eui-Nam Huh, and Choong Seon Hong. Mitigating domain shift in federated learning via intra-and inter-domain prototypes. *arXiv preprint arXiv:2501.08521*, 2025.

Huy Q. Le, Loc X. Nguyen, Yu Qiao, Seong Tae Kim, Eui-Nam Huh, and Choong Seon Hong. Feddap: Domain-aware prototype learning for federated learning under domain shift. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3390–3399, June 2026.

Khiem Le, Long Ho, Cuong Do, Danh Le-Phuoc, and Kok-Seng Wong. Efficiently assemble normalization layers and regularization for federated domain generalization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6027–6036. IEEE, 2024.

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

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling-Yu Duan. Uncertainty modeling for out-of-distribution generalization. *arXiv preprint arXiv:2202.03958*, 2022.

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

Ying Li, Xingwei Wang, Rongfei Zeng, Praveen Kumar Donta, Ilir Murturi, Min Huang, and Schahram Dustdar. Federated domain generalization: A survey. *Proceedings of the IEEE*, 113 (4):370–410, 2025.

Tianchi Liao, Binghui Xie, Lele Fu, Sheng Huang, Bowen Deng, Chuan Chen, and Zibin Zheng. Federated domain generalization with decision insight matrix. In *IJCAI*, pp. 5689–5697, 2025.

Jingwei Liu, Yating Li, Mengjiao Zhao, Lei Liu, and Neeraj Kumar. Epffl: enhancing privacy and fairness in federated learning for distributed e-healthcare data sharing services. *IEEE Transactions on Dependable and Secure Computing*, 2024.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1013–1023, 2021.

Xiao Ma, Young D Kwon, Pan Zhou, and Dong Ma. Architecture-agnostic test-time adaptation via backprop-free embedding alignment. In *International Conference on Learning Representations*, volume 2026, pp. 127222–127244, 2026.

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

A Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022.

Binh Nguyen, Minh-Duong Nguyen, Jinsun Park, Viet Pham, and Won-Joo Hwang. Federated domain generalization with data-free on-server matching gradient. In *International Conference on Learning Representations*, volume 2025, pp. 77488–77513, 2025.

Thinh Nguyen, Trung Phan, Binh Nguyen, Khoa D Doan, and Kok-Seng Wong. Hfedatm: Hierarchical federated domain generalization via optimal transport and regularized mean aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39435–39444, 2026.

Ziru Niu, Hai Dong, and AK Qin. Bridging generalization gap of heterogeneous federated clients using generative models. In *International Conference on Learning Representations*, volume 2026, pp. 130041–130059, 2026.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Farhad Pourpanah, Mahdiyar Molahasani, Milad Soltany, Michael Greenspan, and Ali Etemad. Federated unsupervised domain generalization using global and local alignment of gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19948–19958, 2025.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 4510–4520. Ieee, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022.

Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 52–61, 2021.

Jiaqi Wang, Xingyi Yang, Suhan Cui, Liwei Che, Lingjuan Lyu, Dongkuan (DK) Xu, and Fenglong Ma. Towards personalized federated learning via heterogeneous model reassembly. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 29515–29531. Curran Associates, Inc., 2023a.

Lei Wang, Jieming Bian, Letian Zhang, Chen Chen, and Jie Xu. Taming cross-domain representation variance in federated prototype learning with heterogeneous data domains. *Advances in Neural Information Processing Systems*, 37:88348–88372, 2024.

Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, Ming Gao, et al. Dfrd: Data-free robustness distillation for heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 36:17854–17866, 2023b.

Zheng Wang, Zihui Wang, Xiaoliang Fan, and Cheng Wang. Federated learning with domain shift eraser. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4978–4987. IEEE, 2025.

Barry Payne Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Siyuan Wu, Yongzhe Jia, Bowen Liu, Haolong Xiang, Xiaolong Xu, and Wanchun Dou. Pfedcs: A personalized federated learning method for enhancing collaboration among similar classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21572–21580, 2025a.

Siyuan Wu, Yongzhe Jia, Haolong Xiang, Xiaolong Xu, Xuyun Zhang, Lianyong Qi, and Wanchun Dou. A fair federated learning method for handling client participation probability inconsistencies in heterogeneous environments. In *Advances in Neural Information Processing Systems*, volume 38, pp. 150102–150131, 2025b. doi: 10.52202/085713-5019.

Siyuan Wu, Sheng Lu, Hao Tian, Zheng Li, Guoqiang Liu, Xuyun Zhang, and Wanchun Dou. A model-heterogeneous federated unlearning method via negative knowledge distillation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5631–5635. IEEE, 2026.

Yunlu Yan, Huazhu Fu, Yuexiang Li, Jinheng Xie, Jun Ma, Guang Yang, and Lei Zhu. A simple data augmentation for feature distribution skewed federated learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25749–25758. IEEE, 2025.

Seunghan Yang, Seokeon Choi, Hyunsin Park, Sungha Choi, Simyung Chang, and Sungrack Yun. Feature diversification and adaptation for federated domain generalization. In *European Conference on Computer Vision*, pp. 52–70. Springer, 2024.

Dezhong Yao, Wanning Pan, Yutong Dai, Yao Wan, Xiaofeng Ding, Chen Yu, Hai Jin, Zheng Xu, and Lichao Sun. Fedgkd: Towards heterogeneous federated learning via global knowledge distillation. *IEEE Transactions on Computers*, 2023.

Dezhong Yao, Tongtong Liu, Yuexin Shi, and Zhiqiang Xu. Fedmho: Heterogeneous one-shot federated learning towards resource-constrained clients. In *Proceedings of the ACM Web Conference 2026*, pp. 5686–5697, 2026.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8686–8696, 2023.

Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoxiao Li, et al. Federated model heterogeneous matryoshka representation learning. *Advances in Neural Information Processing Systems*, 37: 66431–66454, 2024.

Jianqiao Zhang, Caifeng Shan, and Jungong Han. Fedgmkd: An efficient prototype federated learning framework through knowledge distillation and discrepancy-aware aggregation. *Advances in Neural Information Processing Systems*, 37:118326–118356, 2024a.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16768–16776, 2024b.

Jie Zhang, Song Guo, Jingcai Guo, Deze Zeng, Jingren Zhou, and Albert Y Zomaya. Towards data-independent knowledge transfer in model-heterogeneous federated learning. *IEEE Transactions on Computers*, 72(10):2888–2901, 2023a.

Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated domain generalization with generalization adjustment. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3954–3963. IEEE, 2023b.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13025–13032, 2020a.

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

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

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2023.

Yanbing Zhou, Xiangmou Qu, Chenlong You, Jiyang Zhou, Jingyue Tang, Xin Zheng, Chunmao Cai, and Yingbo Wu. Fedsa: A unified representation learning via semantic anchors for prototype-based federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 23009–23017, 2025a.

Zikun Zhou, Wen Huang, Xingyi Wang, Zhishuo Zhang, Zhun Zhang, Jian Peng, and Feihu Huang. Improving federated domain generalization through dynamical weights calculated from data influences on global model update. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23045–23053, 2025b.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pp. 12878–12889. PMLR, 2021.

Appendix

A PSEUDOCODE OF FEDVFG

The overall training procedure of FedVFG is summarized in Algorithm 1. Please refer to Section 4 for more details.

Algorithm 1 Training process of FedVFG

Input: Client datasets $\{\mathcal{D}_k\}_{k=1}^K$, domain identities $\{s(k)\}_{k=1}^K$, local models $\{f_k = h_k \circ g_k\}_{k=1}^K$, communication rounds T, local epochs E.
Output: Optimized local models $\{f_k\}_{k=1}^K$.

Initialization:
for all clients $k = 1, \ldots, K$ **in parallel do**
    Train f_k for E epochs using local cross-entropy loss $\mathcal{L}_{\text{CE}}$;
    Compute and upload local class statistics by Eqs. 3–4;
end for
Server calibrates the uploaded statistics by Eq. 5 and constructs domain statistics by Eqs. 6–7;
for round $t = 1, \ldots, T$ **do**
    Server Side:
    Select active clients $\mathcal{A}_t$ and broadcast the domain statistics $\bar{\mathcal{S}}$;
    Client Update:
    for all $k \in \mathcal{A}_t$ **in parallel do**
        Restore the received domain statistics to client k's feature scale;
        for local epoch $e = 1, \ldots, E$ **do**
            for all mini-batches $\mathcal{B}_k \subset \mathcal{D}_k$ **do**
                Compute local features $\boldsymbol{z}_i = g_k(x_i)$;
                Select the domain pair with maximum support by Eq. 8;
                if a valid domain pair exists **then**
                    Generate virtual features by Eqs. 9–13;
                    Compute $\mathcal{L}_{\text{VFG}}^k$ by Eq. 15;
                else
                    $\mathcal{L}_{\text{VFG}}^k \leftarrow 0$;
                end if
                Update f_k using $\mathcal{L}_k$ in Eq. 16;
            end for
        end for
        Recompute and upload local class statistics;
    end for
    Server updates the cached statistics and recomputes $\bar{\mathcal{S}}$;
end for
return $\{f_k\}_{k=1}^K$.

B DETAILS OF COMPARED METHODS

We compare FedVFG with seven representative FL baselines considered in the main experiments. For FPL, FedPLVM, and FedDAP, which do not natively support model-heterogeneous training, we remove their internal model-parameter aggregation components and retain only prototype transmission and aggregation to ensure compatibility with heterogeneous client models. We briefly describe the compared methods below:

FedProto Tan et al. (2022). FedProto communicates class prototypes instead of model parameters. The server aggregates local class prototypes into global class prototypes and sends them back to the clients, which jointly optimize the classification loss and a prototype-alignment loss.

FPL Huang et al. (2023). FPL addresses domain shift with two complementary types of prototypes. Cluster prototypes preserve diverse domain knowledge by encouraging samples to approach prototypes of the same class, while unbiased prototypes provide a stable cross-domain target through consistency regularization.

FedTGP Zhang et al. (2024b). FedTGP replaces naive weighted averaging of local class prototypes with trainable global class prototypes optimized on the server. Its adaptive-margin-enhanced contrastive learning objective improves inter-class separation while preserving the semantic meaning of the prototypes.

FedPLVM Wang et al. (2024). FedPLVM models cross-domain representation variance through variance-aware prototype clustering at both local and global levels. It further introduces an α-sparsity prototype loss to improve intra-class compactness and reduce inter-class confusion, especially for underrepresented domains.

FedSA Zhou et al. (2025a). FedSA introduces semantic anchors that decouple prototype generation from biased local representations. Anchor-based regularization, margin-enhanced contrastive learning, and classifier calibration are used together to learn consistent representations and more separable decision boundaries across clients.

FedVTC Niu et al. (2026). FedVTC transfers knowledge through generative models rather than directly aligning local features from heterogeneous clients. Each client learns a variational transposed-convolutional generator, distribution matching regularizes the synthesis process, and the aggregated generators produce synthetic data for local fine-tuning.

FedDAP Le et al. (2026). FedDAP preserves domain information by constructing domain-specific class prototypes through similarity-weighted aggregation within each domain. Local features are aligned with prototypes from the same domain and separated from prototypes of different domains, yielding domain-aware prototype alignment.

C Model Architectures Used in Experiments

To construct the model-heterogeneous setting, we employ four different client architectures: 4-layer CNN, GoogLeNet Szegedy et al. (2015), MobileNetV2 Sandler et al. (2018), and ResNet-18 He et al. (2016). These models differ substantially in network structure and parameter size, providing a diverse heterogeneous model pool for evaluation. Table 4 summarizes their parameter counts on PACS with seven output classes. Following Wang et al. (2024), all clients use ResNet-10 He et al. (2016) in the homogeneous setting.

Table 4: Parameter counts of the evaluated models. "M" denotes one million.

Model	Parameter Count
4-layer CNN	0.88M
GoogLeNet	5.60M
MobileNetV2	2.23M
ResNet-10	4.91M
ResNet-18	11.18M

D Hyperparameter Settings

Table 5 summarizes the hyperparameter settings used in our experiments. Unless otherwise specified, all methods follow the same settings. [2] FedVTC uses 90 federated rounds with 5 local epochs per round, followed by 50 fine-tuning rounds with one epoch each, resulting in the same 500 local training epochs as the other methods. For FedVFG, we set $\lambda = 10$ and $\rho_{\max} = 1.0$ on both datasets. The scale bounds are set as $[s_{\min}, s_{\max}] = [0.5, 5.0]$. The remaining method-specific hyperparameters are reported in Table 5.

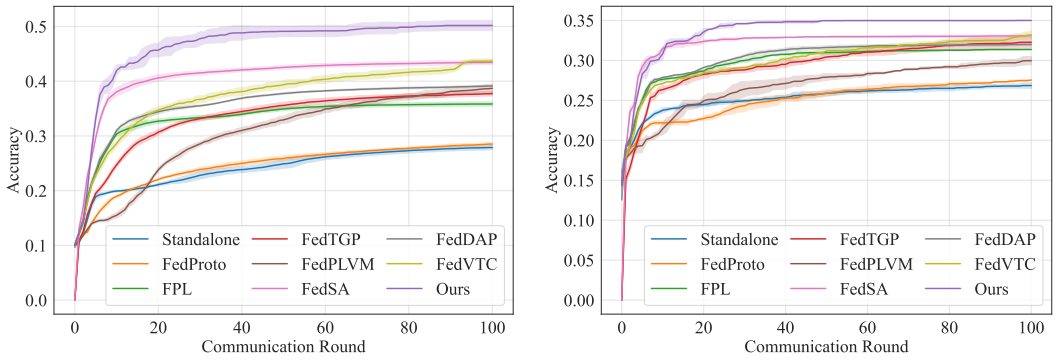
E Comparison of Training Curves

To evaluate the convergence behavior of FedVFG, we compare the average accuracy curves across communication rounds on Digit-5 and PACS. As shown in Figure 5, FedVFG converges rapidly on

[2]Note that these parameter names in different methods are consistent with the original references and are independent of the notation used in our work.

Table 5: Hyperparameter settings used in the experiments.

Type	**Hyperparameters**	**Value**
FL training settings	Optimizer	SGD
	Local learning rate η	0.001
	SGD momentum	0.9
	Weight decay	$1e-5$
	Batch size	64
	Local epochs per round E	5
	Total rounds T	100
	Client participation	100%
	Feature dimension d	512
	Number of clients K	20 for Digit-5 and 16 for PACS
	Sampling ratio r	1% for Digit-5 and 10% for PACS
Framework-specific	λ in FedProto	1
	λ in FPL	1
	τ in FPL	0.02
	λ in FedTGP	0.1
	Server epochs in FedTGP	100
	τ in FedTGP	100
	λ in FedPLVM	100
	α in FedPLVM	0.25
	τ in FedPLVM	0.07
	λ_1 in FedSA	0.1
	λ_2 in FedSA	1
	λ_3 in FedSA	1
	α in FedSA	0.9999
	λ_{DPA} in FedDAP	10
	λ_{CPCL} in FedDAP	1
	τ_{cross} in FedDAP	0.04
	τ_{agg} in FedDAP	0.001
	λ in FedVTC	0.1
	N_s in FedVTC	500
	T_{ft} in FedVTC	50
	E_{ft} in FedVTC	1



(a) Digit-5 (b) PACS

Figure 5: Comparison of training curves in terms of average accuracy on Digit-5 and PACS in the heterogeneous setting.

both datasets and consistently maintains higher accuracy than the compared methods throughout most of the training process. In particular, the performance gap becomes evident in the early rounds and remains stable after convergence, demonstrating that the proposed virtual feature generation effectively facilitates cross-client knowledge transfer under model heterogeneity and domain shift.

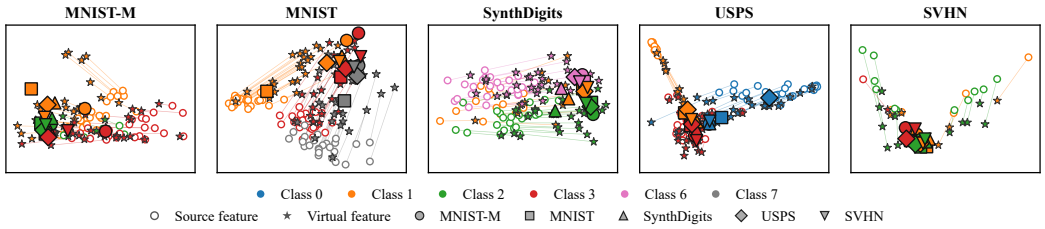


Figure 6: Visualization of virtual feature generation on Digit-5. (○) denote source features and (★) denote their generated virtual features. Colors indicate classes, and line segments connect corresponding feature pairs. *Best viewed in color.*

F Virtual Feature Visualization on Digit-5

Figure 6 further visualizes virtual feature generation on Digit-5 following the same setting as Section 5.5, where source features are sampled from the local training set of the trained 4-layer CNN client in each domain and their virtual counterparts are regenerated using the final domain statistics before PCA projection. Similar to PACS, within each client-specific PCA space, the transformation trajectories differ across classes, showing that the generated features follow structured variations rather than a fixed perturbation pattern. This further illustrates how cross-domain statistics guide local features toward diverse feature configurations while retaining their original labels.

G Performance Comparison in the Homogeneous Setting

Following Wang et al. (2024), we assign ResNet-10 He et al. (2016) to all clients to further evaluate FedVFG without model heterogeneity, while keeping the other experimental settings unchanged. As shown in Tables 6 and 7, FedVFG achieves the best average accuracy on both datasets. On Digit-5, FedVFG reaches 51.13%, outperforming the strongest baseline FedSA by 1.43%, and achieves the highest accuracy across all five domains. On PACS, FedVFG obtains 38.86% average accuracy, exceeding the strongest baseline FedTGP by 0.72%. It achieves the best performance on Photo and Sketch and the second-best result on Art Painting. Compared with standalone training, FedVFG improves the average accuracy by 11.19% and 6.90% on Digit-5 and PACS, respectively. These results demonstrate that FedVFG remains effective under domain shift even when all clients share the same model architecture.

Table 6: Performance comparison (%) on Digit-5 in the homogeneous setting. Avg reports the mean accuracy across domains, and Δ denotes its improvement over Standalone. Best in **bold** and second with underline.

Methods	Digit5						
	MNIST-M	MNIST	SYN	USPS	SVHN	Avg	Δ
Standalone	23.79 ± 0.46	67.57 ± 0.37	21.33 ± 0.30	69.76 ± 0.23	17.24 ± 0.37	39.94	–
FedProto [AAAI22]	20.65 ± 0.35	67.22 ± 1.15	18.52 ± 0.80	70.23 ± 1.00	15.37 ± 0.30	38.40	-1.54
FPL [CVPR23]	25.83 ± 0.51	69.84 ± 0.24	23.43 ± 0.32	71.63 ± 0.18	19.00 ± 0.35	41.95	+2.01
FedTGP [AAAI24]	33.33 ± 0.28	79.34 ± 0.03	33.41 ± 0.28	75.90 ± 0.41	22.97 ± 0.40	48.99	+9.05
FedPLVM [NIPS24]	32.78 ± 0.63	78.38 ± 0.18	32.44 ± 0.20	75.89 ± 0.07	22.44 ± 0.55	48.39	+8.45
FedSA [AAAI25]	35.11 ± 0.31	79.37 ± 0.21	33.95 ± 1.03	76.25 ± 0.51	23.82 ± 0.81	49.70	+9.76
FedDAP [CVPR26]	30.57 ± 0.27	74.91 ± 0.46	28.95 ± 0.25	73.96 ± 0.16	21.64 ± 0.44	46.01	+6.07
FedVTC [ICLR26]	32.01 ± 0.42	77.35 ± 0.23	31.31 ± 0.10	76.35 ± 0.86	22.89 ± 0.69	47.99	+8.05
FedVFG (Ours)	**36.56 ± 0.16**	**81.73 ± 0.40**	**34.86 ± 0.25**	**77.37 ± 0.17**	**25.16 ± 0.42**	**51.13**	**+11.19**

Table 7: Performance comparison (%) on PACS in the homogeneous setting. Avg reports the mean accuracy across domains, and Δ denotes its improvement over Standalone. Best in **bold** and second with underline.

Methods	PACS					
	Photo	Art painting	Cartoon	Sketch	Avg	Δ
Standalone	34.16 ± 0.77	21.55 ± 0.40	31.63 ± 0.13	40.49 ± 0.12	31.96	–
FedProto [AAAI22]	30.47 ± 1.16	18.39 ± 0.97	30.44 ± 0.72	42.05 ± 0.67	30.34	-1.62
FPL [CVPR23]	34.31 ± 0.37	21.30 ± 0.33	31.91 ± 0.18	43.09 ± 0.29	32.65	+0.69
FedTGP [AAAI24]	40.45 ± 0.46	23.11 ± 0.57	40.44 ± 0.65	48.55 ± 0.39	38.14	+6.18
FedPLVM [NIPS24]	37.87 ± 0.80	22.58 ± 0.06	38.90 ± 0.74	46.22 ± 0.23	36.39	+4.43
FedSA [AAAI25]	39.62 ± 0.55	23.03 ± 0.50	**40.49 ± 0.78**	48.38 ± 0.38	37.88	+5.92
FedDAP [CVPR26]	37.33 ± 0.56	22.07 ± 0.48	36.09 ± 0.45	45.27 ± 0.31	35.19	+3.23
FedVTC [ICLR26]	37.92 ± 0.86	**24.04 ± 0.70**	39.93 ± 0.95	46.56 ± 0.94	37.11	+5.15
FedVFG (Ours)	**40.54 ± 0.47**	23.74 ± 0.93	40.07 ± 0.38	**51.07 ± 0.25**	**38.86**	**+6.90**

H Cost and Efficiency Analysis

Communication Cost. Table 8 compares the average bidirectional communication cost per client per round on Digit-5, measured by the number of transmitted parameters. Here, $\mathcal{A}_t$ denotes the set of participating clients at round t, C_k and C denote the number of classes observed by client k and the total number of classes, respectively, d is the feature dimension, and D is the number of domains. For FPL and FedPLVM, G_t denotes the number of downloaded cluster prototypes and C_t^g the number of represented classes in the global prototypes, whose values depend on the clustering results. For FedVTC, $|\Theta|$ denotes the number of VTC parameters and T the total number of communication rounds, with its one-time VTC transmission amortized over all rounds. For simplicity, class-count scalars are omitted as they contribute less than 1% of the total communication when $d = 512$. As shown in Table 8, FedVFG communicates both class prototypes and diagonal variances for local and domain-specific statistics, resulting in 60.31K transmitted parameters per client per round. Although this is higher than prototype-only methods, its communication cost depends only on C, D, and d, rather than the parameter sizes of heterogeneous client models.

Table 8: The average communication cost per client per round on Digit-5 in the heterogeneous setting. The cost is measured by the number of transmitted parameters, where 'K' denotes one thousand.

Method	Theory	Practice
FedProto	$\frac{d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + C)$	9.68K
FPL	$\frac{d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + G_t + C_t^g)$	14.80K
FedTGP	$\frac{d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + C)$	9.68K
FedPLVM	$\frac{d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + G_t + C_t^g)$	14.80K
FedSA	$\frac{d+1}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t} C_k + Cd$	9.69K
FedVTC	$2d\left(\frac{\sum_{k\in\mathcal{A}_t} C_k}{\lvert\mathcal{A}_t\rvert} + 1\right) + \frac{2\lvert\Theta\rvert + d(C+1)}{T}$	11.71K
FedDAP	$\frac{d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + CD)$	30.16K
FedVFG	$\frac{2d}{\lvert\mathcal{A}_t\rvert}\sum_{k\in\mathcal{A}_t}(C_k + CD)$	60.31K

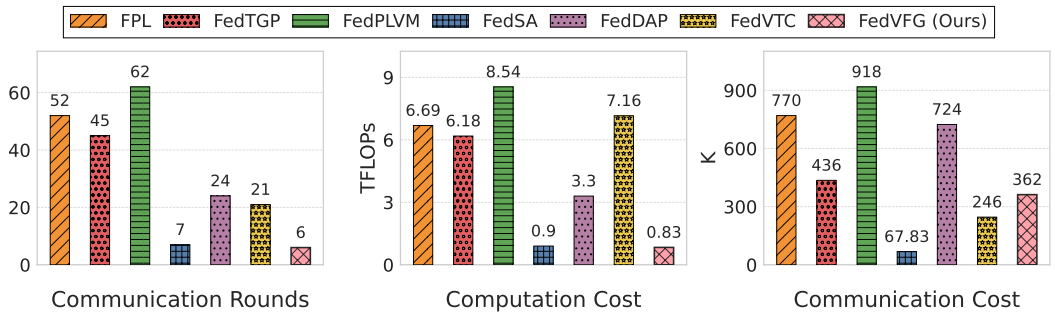


Figure 7: From left to right: Communication rounds, computation FLOPs, and total number of transmitted parameters required to achieve 35% accuracy on Digit-5 in the heterogeneous setting.

Computation Cost. We also compare the computation cost per client per round on Digit-5 in terms of FLOPs (floating-point operations)[3]. Following Zhang et al. (2024b), other operations such as data preprocessing are not included in the FLOPs calculation. All other configurations remain aligned with the main experiments. According to Table 9, FedVFG requires 139.104 GFLOPs, corresponding to an 8.45% increase over standalone training. This overhead mainly comes from virtual feature generation and the additional classification on virtual features. Despite this increase, the computation remains close to other prototype-based methods and is substantially lower than FedVTC, which requires 340.816 GFLOPs per client per round due to the additional VTC training. These results show that FedVFG introduces only moderate computational overhead while avoiding the cost of training an auxiliary generative model.

Table 9: Computation cost per client per round on Digit-5 in the heterogeneous setting. The overhead is computed relative to standalone local training, where 'G' denotes one billion.

Method	**FLOPs**	**Overhead** (↑)
Standalone	128.27G	–
FedProto	128.59G	0.25%
FPL	128.62G	0.28%
FedTGP	137.39G	7.11%
FedPLVM	137.77G	7.41%
FedSA	128.62G	0.27%
FedVTC	340.82G	165.71%
FedDAP	137.51G	7.21%
FedVFG	139.10G	8.45%

Efficiency Analysis. We further evaluate the overall efficiency of different methods by measuring the communication rounds, cumulative computation cost, and total communication cost required to reach 35% accuracy on Digit-5[4]. As shown in Figure 7, FedVFG reaches the target accuracy in only 6 communication rounds, while all compared baselines require at least 7 rounds. Benefiting from its faster convergence, FedVFG incurs only 0.83 TFLOPs of cumulative computation, which is the lowest among all compared methods. Moreover, despite communicating richer class statistics in each round, its total communication cost is reduced to approximately 362K transmitted parameters, lower than FPL, FedTGP, FedPLVM, and FedDAP. These results demonstrate that FedVFG provides a favorable overall efficiency trade-off for reaching the target accuracy in the heterogeneous setting.

[3]We calculate FLOPs using the thop library: `https://github.com/Lyken17/pytorch-OpCounter`.

[4]FedProto is excluded because it does not reach the target accuracy within the training budget.

I LEAVE-ONE-DOMAIN-OUT GENERALIZATION

To further evaluate the cross-domain generalization ability under model heterogeneity, we conduct leave-one-domain-out experiments on Digit-5, where one domain is held out as the target domain and the remaining domains are used for federated training. Table 10 reports the accuracy of the final-round models on each held-out target domain. FedVFG achieves the best performance on every target domain and improves the average accuracy from 21.46% of the strongest baseline FedSA to 27.22%. The gains are particularly pronounced on MNIST and USPS, where FedVFG reaches 37.90% and 40.32%, respectively. These results further demonstrate that the domain-specific statistics and virtual feature generation in FedVFG improve the transferability of heterogeneous client models to unseen domains.

Table 10: Leave-one-domain-out generalization performance (%) on Digit-5 in the heterogeneous setting. Δ denotes its improvement over Standalone. Best in **bold** and second with underline.

Methods	Digit-5						
	→MNIST-M	→MNIST	→SYN	→USPS	→SVHN	Avg	Δ
Standalone	14.02	17.49	11.88	18.21	10.71	14.46	–
FedProto [AAAI22]	13.44	18.09	12.12	19.60	13.43	15.34	+0.88
FPL [CVPR23]	16.93	24.51	14.02	25.08	14.18	18.94	+4.48
FedTGP [AAAI24]	17.69	23.68	14.45	28.04	14.72	19.72	+5.26
FedPLVM [NIPS24]	17.59	25.58	15.16	26.99	15.15	20.09	+5.63
FedSA [AAAI25]	18.48	27.36	16.46	30.13	14.89	21.46	+7.00
FedDAP [CVPR26]	18.36	25.50	15.26	29.46	15.05	20.72	+6.26
FedVTC [ICLR26]	17.33	25.00	14.19	28.65	14.75	19.98	+5.52
FedVFG (Ours)	**21.49**	**37.90**	**18.48**	**40.32**	**17.91**	**27.22**	**+12.76**

J LIMITATIONS

Despite the promising results, FedVFG has two main limitations:

First, FedVFG represents each domain-specific class distribution using a prototype and diagonal variance. While this lightweight representation is effective and communication-efficient, it may not fully capture complex feature correlations or multimodal class distributions. More expressive yet efficient distribution representations could be explored in future work.

Second, our evaluation is conducted on two multi-domain datasets with a limited set of heterogeneous model architectures. Although FedVFG demonstrates consistent improvements in these settings, its generalizability should be further verified on larger-scale datasets, more diverse model architectures, and practical participation settings.