PARDON: Privacy-Aware and Robust Federated Domain Generalization

***Abstract*—While Federated Learning (FL) shows promise in preserving privacy and enabling collaborative learning, most current solutions concentrate on private data collected from a single domain. Yet, a substantial performance degradation on unseen domains arises when data among clients is drawn from diverse domains (i.e., domain shift). However, existing Federated Domain Generalization (FedDG) methods are typically designed under the assumption that each client has access to the complete dataset of a single domain. This assumption hinders their performance in real-world FL scenarios, which are characterized by domain-based heterogeneity—where data from a single domain is distributed heterogeneously across clients—and client sampling, where only a subset of clients participate in each training round.**

In addition, certain methods enable information sharing among clients, raising privacy concerns as this information could be used to reconstruct sensitive private data. To overcome this limitation, we present PARDON, a novel FedDG paradigm designed to robustly handle more complicated domain distributions between clients while ensuring security. PARDON facilitates client learning across domains by extracting an interpolative style from abstracted local styles obtained from each client and using contrastive learning. This approach provides each client with a multi-domain representation and an unbiased convergent target. Empirical results on multiple datasets, including PACS, Office-Home, and IWildCam, demonstrate PARDON's superiority over state-of-the-art methods. Notably, our method outperforms state-of-the-art techniques by a margin ranging from 3.64 to 57.22% in terms of accuracy on unseen domains. Our code is available at https://anonymous.4open.science/r/Fed-Domain-Generalization.

***Index Terms*—Federated Learning, Domain Generalization, Robustness and Privacy, Style Transfer, Contrastive Learning**

I. Introduction

Federated learning (FL) [24] is a distributed machine learning paradigm that facilitates training a single unified model from multiple disjoint contributors, each of whom may own or control their private data. The security aggregation mechanism and its distinctive distributed training mode render it highly compatible with a wide range of practical applications that have strict privacy demands [29, 39]. Data heterogeneity [14, 19, 20, 25] presents a substantial challenge in FL due to the diverse origins of data, each with a distinct distribution. In this scenario, each client optimizes towards the local minimum of empirical loss, deviating from the global direction. Therefore, the averaged global model unavoidably faces a slow convergence speed [19] and achieves limited performance improvement [45]. Many techniques have been developed to handle data heterogeneity in FL [8, 18, 30]. However, these techniques primarily address label skew — i.e., the variation in label distributions among client's data within the same domain.

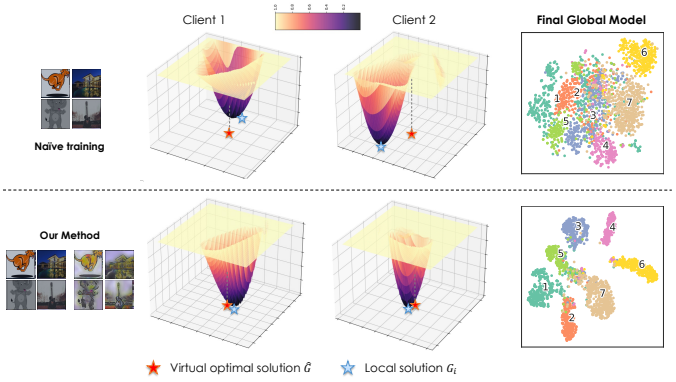


Fig. 1: Loss landscape visualization of two clients under domain-based heterogeneity, with normal training (first row) and our method using interpolative style-transferred data (second row). The vertical axis shows the loss (i.e., solution for the global optimum G and each local objective G_i). The horizontal plane represents a parameter space centered at the global model weight. The feature visualization in the last column demonstrates that our method achieves superior classification results on unseen domains.

However, real-world data heterogeneity in practical applications extends beyond class imbalances, introducing complexities such as domain shifts due to geographical dispersion of data collectors, like cameras and sensors across different hospitals in medical settings [9, 11, 32]. This problem is widely-addressed in centralized machine learning, where all training data is collected in central storage. Most existing DG methods for centralized ML [28, 42, 46] require sharing the representation of the data across domains or access to data from a pool of multiple source domains (in a central server) [16]. However, this violates FL's privacy-preserving nature and raises a critical need for specialized DG methods in FL.

Recognizing this issue, several Federated Domain Generalization (FedDG) methods have attempted to address domain shift in FL [23, 27, 43, 51], but their methods and evaluations often reveal several limitations. Firstly, current evaluations are confined to testing on datasets with limited domain diversity and a small number of domains. Secondly, existing approaches are highly sensitive to the number of clients involved in training, and *domain-based client heterogeneity* [2] where each client distribution is a (different) mixture of train domain distributions. In addition, *client sampling* scenarios where only a subset of clients engage are often overlooked, limiting the generalizability of these methods to popular FL setups [5, 7].

We argue that existing techniques rely on local signals, such as training loss and gradient sign, to mitigate overfitting by penalizing model complexity. However, by treating each client signal as a complete domain indicator, these methods may struggle to generalize when a single client lacks comprehensive domain data. In addition, other methods using cross-sharing local signals, such as amplitude, style statistics, and class-level prototypes, among clients can potentially result in privacy breaches, as discussed in prior studies [22, 27, 51].

To address these challenges, we propose shifting from relying on individual local information in each round to using an aggregated *interpolation style*, which fuses shared domain knowledge from all clients from the beginning to mitigate the limitations caused by *client sampling*. This global *interpolation style* is formed through two-level clustering and unbiased blending of local style statistics, creating a representative style that captures inter-client domain diversity to address the limitations of *domain-based client heterogeneity*. By sharing this global style instead of specific local signals, we reduce privacy risks while maintaining superior performance, as the global style conceals client-specific data and class-level information. Given the *interpolation style*, each client can generate new data containing the global interpolative style, leveraging domain diversity from all clients. We enforce the model to represent the original data in a manner that is more similar to the style-transferred data from the same class while pushing it away from the style-transferred data from other classes. This approach enhances generalization by encouraging the model to focus on domain-invariant features, gradually aligning it with the global model and preventing bias towards local data. Fig. 1 intuitively illustrates local models converging towards their local optima via loss landscape visualization. Our method aims at guiding a local model to optimize towards a converged optimal solution for inter-domain data. This results in better discrimination of unseen domains, as illustrated in the TSNE plot in the rightmost column of Fig. 1.

We claim the following major contributions in this work.

- We are the first to investigate FedDG methods under both *domain-based client heterogeneity* and *client sampling* scenarios, pinpointing the underlying limitation of current methods: the client's signals in each round are insufficient for representing the complete global domain knowledge, resulting in a model distorted by partial observation.
- We present PARDON, a FedDG method that enhances robustness and privacy under client sampling and domain heterogeneity. PARDON employs multi-level unsupervised clustering to identify an unbiased interpolative style vector that represents styles across client domains. Then, we introduce a multi-domain contrastive learning mechanism that guides each local model to learn a multi-domain representation aligned with the global representation, reducing local bias from individual datasets.
- We rigorously validate the effectiveness of our resulting model with a wide range of datasets (i.e., both small domain and large domain), varied domain distribution, and number of client settings. Our results demonstrate that PARDON outperforms existing FedDG methods, maintaining comparable computational overhead while effectively preserving client privacy and sensitive information.

II. Related Work

While FedDG shares a common objective with centralized Domain Generalization (DG), i.e., generalizing from multi-source domains to unseen domains, FedDG is distinct from traditional DG in prohibiting direct data sharing among clients. This section first examines how domain generalization is tackled in centralized ML approaches, then delves into existing FedDG methodologies closely related to ours.

Domain Generalization. Domain generalization aims to learn a model from multiple source domains so that the model can generalize to an unseen target domain. Existing work has explored domain shifts from various directions in a centralized data setting. These methods can be divided into three categories [47], including *data manipulation* to enrich data diversity [38, 52], *domain-invariant feature* distribution across domains for the robustness of conditional distribution to enhance the generalization ability of the model [26, 28], and exploiting general learning strategies such as *meta-learning-based* [31] and transfer learning [50] to promote generalizability. However, many of these methods require centralized data from different domains, violating local data preservation in federated learning. Specifically, access for more than one domain is needed to augment data or generate new data in [41]; domain-invariant representation learning or decomposing features are performed under the comparison across domains, and some learning strategy-based methods use an extra domain for meta-update [1, 34]. There exists some methods that do not explicitly require centralized domains and can be adapted for federated learning with minor adjustments. However, these methods such as MixStyle [52],[48] and JiGen [3] offer minimal improvement in addressing domain shift in FL due to constrained intra-client and differing inter-client distributions, as noted by [2, 4]. This highlights the preference for a specialized FedDG method that preserves the distributed nature of FL while efficiently enhancing generalizability.

Federated Domain Generalization. Existing works tackling heterogeneity among clients in FL [10, 14, 35] have not considered model performance under the domain shift between training and testing data. Recently, works tackling DG in FL [4, 11, 23, 27, 43, 51] have been proposed. First, *cross-information sharing* lets other clients see information abstracted from one client's domain — for example, the amplitude spectrum [23, 40] and the representation statistics [4, 33]. Other clients then use this information to produce synthetic data that contains domain information to enrich the local data. In these methods, clients in FL are encouraged to share their sample-level or class-level information. Thus, these methods cause additional costs and risks of data privacy leakage since the style information of each client is publicly available to other clients or the server. Second, *multi-object optimization* aligns the generalization gaps among clients. For example, Zhang et al. [51] create a new optimization goal with a regularizer to

lower variance and get a tighter generalization bound. Nguyen et al. [27] proposed restricting the representation's complexity to help minimize the distribution distance between specific and reference domains. In Tenison et al. [43], the authors suggested using a gradient-masked averaging aggregation method based on the signed agreement of the local updates to eliminate weight parameters with much disagreement. Besides these two approaches, third, *unbiased learning* [11] improves domain generalization by creating unbiased by-class prototypes and using regularization to line up the local models with the unbiased prototypes. However, previous studies on FedDG have often focused on domain-isolated settings where the number of clients equals the number of training domains. In addition, these methods show limited effectiveness in more complicated scenarios such as domain-based heterogeneity and datasets with a large number of domains [2]. Moreover, client sampling, a common setting in FL [5, 7], is typically overlooked, where only a portion of clients participate in each training round. Considering client sampling is essential as it reflects real-world scenarios where involving all clients in every round is impractical due to constraints such as communication bandwidth, computational resources, and energy limitations [7, 14]. Unlike prior methods, we relax the assumption of isolated domain distribution by considering various client-domain heterogeneity and client sampling scenarios. We argue that efficient FedDG methods must perform robustly and securely under these settings with reasonable computation cost.

III. Methodology

In this section, we formally introduce the domain-related definitions used in this work and then present PARDON, our approach for domain generalization in FL.

A. Preliminaries

First, we extend domain and domain generalization definitions in [47] into domain shift and domain-based client heterogeneity [2] within the context of FL.

Definition 1 (Domain)**.** *Let $\mathcal{X}$ denote a nonempty input space and $\mathcal{Y}$ an output space. A* domain *comprises data sampled from a distribution. We denote it as $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \sim P_{XY}$, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d, y \in \mathcal{Y} \subset \mathbb{R}$ denotes the label, and P_{XY} denotes the joint distribution of the input sample and output label. X and Y denote the corresponding random variables.*

Definition 2 (Domain Generalization)**.** *We are given M training (source) domains $\mathcal{S}_{train} = \left\{\mathcal{S}^i \mid i = 1, \cdots, M\right\}$ where $\mathcal{S}^i = \left\{\left(\mathbf{x}_j^i, y_j^i\right)\right\}_{j=1}^{n_i}$ denotes the i-th domain. The joint distributions between each pair of domains are different: $P_{XY}^i \neq P_{XY}^j, 1 \leq i \neq j \leq M$.*
The goal of domain generalization *is to learn a robust and generalizable predictive function $h : \mathcal{X} \to \mathcal{Y}$ from the M training domains to achieve a minimum prediction error on an unseen test domain $\mathcal{S}_{test}$ (i.e., $\mathcal{S}_{test}$ cannot be accessed in training and $P_{XY}^{test} \neq P_{XY}^i$ for $i \in \{1, \cdots, M\}$), i.e., $\min_h \mathbb{E}_{(\mathbf{x},y)\in\mathcal{S}_{test}}[\ell(h(\mathbf{x}), y)]$, where $\mathbb{E}$ is the expectation and $\ell(\cdot, \cdot)$ is the loss function.*

Definition 3 (Domain Shift in FL)**.** *We consider a scenario where M training domains S_{train} are distributed among K participants (indexed by k) with respective private datasets $\mathcal{D}_k = \left\{(\mathbf{x}_i^k, y_i^k)\right\}_{i=1}^{N_k}$, where N_k denotes the local data scale of participant k. In heterogeneous FL, the conditional feature distribution $P(\mathbf{x}|y)$ may vary across participants, even if $P(y)$ remains consistent, leading to domain shift: $P_k(\mathbf{x} \mid y) \neq P_l(\mathbf{x} \mid y)$, where $P_k(y) = P_l(y)$.*

Definition 4 (Domain-based Client Heterogeneity)**.** *We consider domain-based client heterogeneity as an instance of domain shift in FL, such that each client distribution $\mathcal{D}_i$ is a mixture of training domain distributions, which is: $\mathcal{D}_i(\mathbf{x}, y) = \sum_{d\in\mathcal{S}_{train}} w_{i,d} \cdot S_d(\mathbf{x}, y)$, where $w_{i,d}$ represents the proportion of samples from domain d within client i.*

B. PARDON: Architecture

Fig. 2 presents the overall architecture of PARDON. Our method includes four main processes: ① local style calculation, ② interpolation style extraction, ③ contrastive-based local training, and, ④ aggregation. In this subsection, we present how each step is performed. In this work, we adopt the FL scheme formalized from previous works [14, 24], in which all clients agree on sharing a model with the same architecture. We regard the model as having two modules: a feature extractor and a unified classifier. The feature extractor $f : \mathcal{X} \to \mathcal{Z}$, encodes sample x into a compact d dimensional feature vector $z = f(x) \in \mathbb{R}^d$ in the feature space $\mathcal{Z}$. A unified classifier $g : \mathcal{Z} \to \mathbb{R}^{|I|}$ maps feature z into logits output $y = g(z)$, where I is the classification category.

Local style calculation. Initially, each client must compute their unique style and upload it to the central server. The style information extracted from each local client is carefully abstracted to ensure that it cannot be employed to reconstruct the original dataset. For this purpose, we have chosen to represent the style of each client by statistics of its local data's feature embedding. Importantly, it has been demonstrated to be highly challenging to reverse-engineer the original dataset solely from this style information (Chen et al. [4]). Firstly, to extract the feature representation of each image, we use a pre-trained VGG encoder Φ of real-time style transfer model AdaIN [12], which will be used to generate interpolative style-transferred data at the next step.

Assume the feature $\Phi(x)$ has the shape of $d \times h \times w$ given channel dimension x, in which $h \times w$ is the feature map resolution, and d is the number of dimensions. Then, denote the representation set of a client $\mathcal{C}_k$ as $\Phi(\mathcal{C}_k)$, where $\Phi(\mathcal{C}_k) := \{\Phi(x_i) | \ \forall x_i \in \mathcal{D}_k\}$. Since each client k may contain data from more than one domain, we cannot estimate precisely the ratio of each domain, i.e., $w_{i,d}$ in Definition 4. Still, we must prevent the local style from being biased by the dominant domains. Therefore, we categorize samples based on different styles using FINCH [36]. This hierarchical clustering method identifies groupings in the data based on its neighborhood relationship without needing hyper-parameters or thresholds. It enables us to discover inherent clustering

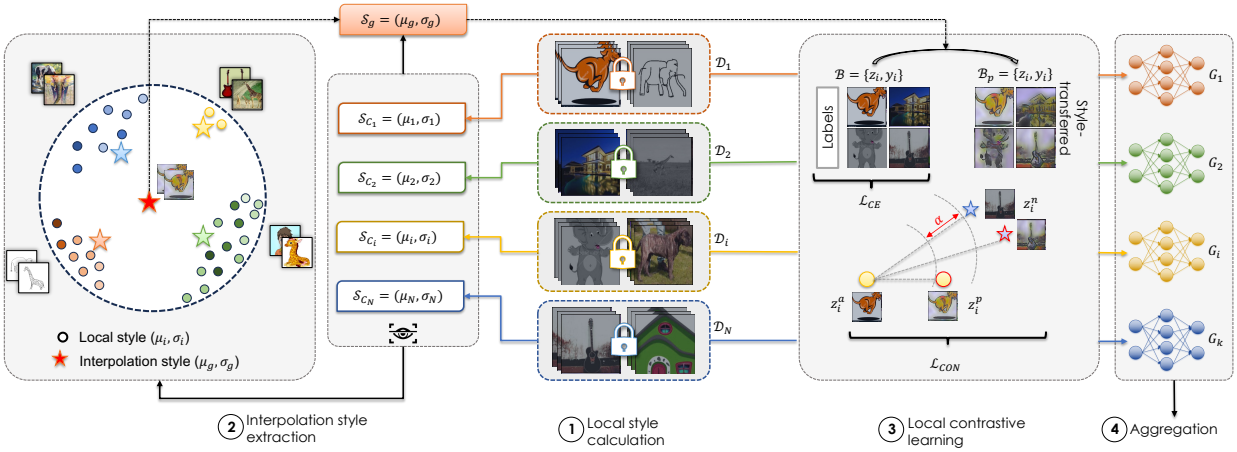


Fig. 2: The PARDON framework we present in this paper. Clients calculate their style information using their local data and a pre-trained encoder in Step ①. The server extracts interpolation style by weighted averaging the local style information in Step ②. Then, in Step ③, local clients exploit the interpolation style to obtain style-transferred data and update their models via contrastive learning. The aggregated global model in Step ④ runs inference on unseen data.

tendencies inside each client's samples. Compared to traditional clustering methods, FINCH is more suitable for scenarios where each client contains an uncertain number of clusters. Mathematically, given a set of features $\mathcal{F}$, FINCH outputs a set of partitions $\mathcal{L} = \{\Gamma_1, \Gamma_2, \cdots, \Gamma_L\}$ where each partition $\Gamma_i = \{C_1, C_2, \cdots, C_{\Gamma_i} | \ C_{\Gamma_i} > C_{\Gamma_{i+1}} \forall i \in \mathcal{L}\}$ is a valid clustering of $\mathcal{F}$, and the last clustering Γ_L has the smallest number of classes. Here, we use FINCH to group local data points based on their representation $\Phi(\mathcal{C}_k)$. Specifically,

$$\Phi(\mathcal{C}_k) \xrightarrow{\text{FINCH}} \Gamma_{L_k} = \{\epsilon_j \subset \Phi(\mathcal{C}_k) \mid \epsilon_j \cap \epsilon_k = \emptyset, \ \forall j \neq k\}_{j=1}^{L_k} \tag{1}$$

This technique implicitly clusters samples with similar characteristic styles, as styles from other domains are less likely to be adjacent. Consequently, samples from distinct domains are unlikely to merge, while those from similar domains are naturally grouped. Specifically, we employ cosine similarity to quantify the closeness between two image styles, identifying the style with the smallest distance as its "neighbor."

Given a cluster ϵ_j, the concatenation of all its elements, $\Phi(\epsilon_j)$, has the shape $\mathbb{R}^{|\epsilon_j| \times d \times w \times h}$. The style of the cluster, denoted as $\mathcal{S}(\epsilon_j)$, is derived as the pixel-level channel-wise mean and standard deviation of the feature representations across all elements. Formally, $\mathcal{S}(\epsilon_j)$ is computed as follows:

$$\mathcal{S}(\epsilon_j) = \left(\mu\left(\Phi(\epsilon_j)\right), \sigma\left(\Phi(\epsilon_j)\right)\right) \in \mathbb{R}^{2d} \tag{2}$$

The local style information for each client k is now the set of all clusters' styles, i.e., $\mathcal{S}_{\mathcal{C}_k} := \{\mathcal{S}(\epsilon_j)\}_{j=1}^{L_k}$ and the client's style statistic is then determined by the average of all cluster styles, i.e., $S_{\mathcal{C}_k} = \frac{1}{L_k} \sum_{j=1}^{L_k} \mathcal{S}(\epsilon_j) \in \mathbb{R}^{2d}$.

Interpolation Style Extraction. Since we consider FL settings with many participants and domain-based client heterogeneity among them, given a set of local styles $\mathbf{S} = \{\mathcal{S}_{\mathcal{C}_i}\}_{i=1}^{N}$, our objective is to find an optimal combination of these local styles, denoted as $\mathcal{S}_g$. Specifically, we employ FINCH to identify grouping relationships among client styles, forming clusters that capture shared characteristics and account for the inherent heterogeneity among participants.

$$\mathbf{S} \xrightarrow{\text{FINCH}} \Gamma_L = \{\varepsilon_j \subset \mathbf{S} \mid \varepsilon_j \cap \varepsilon_k = \emptyset, \ \forall j \neq k\}_{j=1}^{L} \tag{3}$$

The output of this process is a set of L non-overlapping subsets of client styles from $\mathbf{S}$. Since different clients may share similar domains or multiple clients may belong to the same domain, we reduce the number of local style representations to L cluster styles, as described in Eq. 3. Each cluster j is represented by the average style of all $S_{\mathcal{C}_i} \in \varepsilon_j$, computed as:

$$\mathcal{S}(\varepsilon_j) = \frac{1}{|\varepsilon_j|} \sum_{S_{\mathcal{C}_i} \in \varepsilon_j} S_{\mathcal{C}_i} \in \mathbb{R}^{2d} \tag{4}$$

After this step, the clustering method groups clients with close styles together, and we consider each cluster style equally rather than treating each client equally. Finally, the global interpolation style $\mathcal{S}_g$ is defined as the median of all cluster style statistics, i.e.,

$$\mathcal{S}_g = median\left(\mathcal{S}(\varepsilon_j) \mid \forall \varepsilon_j \in \Gamma_L\right) \in \mathbb{R}^{2d} \tag{5}$$

This enables domains with low cardinality to engage in the interpolation style, hence facilitating the dominant domain's knowledge acquisition from these smaller populations. The median is robust to outliers (i.e., extreme values) and skewed distributions in style statistics, ensuring no single dominant style skews the global interpolation style. This fairness in representing the central tendency makes the median ideal for capturing diverse client styles without bias, promoting equitable and comprehensive knowledge transfer across all domains. This strategy for calculating the *interpolation style* effectively addresses the bias introduced by domain-based client heterogeneity. By incorporating information from all clients prior to the commencement of training, it circumvents the limitations associated with partial observations due to client sampling during individual training rounds, thereby enhancing

the performance compared to other methods reliant on round-specific client participation.

Local Contrastive Learning. This component demonstrates how each client leverages the global interpolation style, $\mathcal{S}_g$, sent from the server, to learn an unbiased local model G_i. Motivated by the success of contrastive learning in FL, we introduce a multi-domain contrastive learning mechanism specialized for our method, which balances between *inter- and intra-client domain diversity learning*. The key idea is to use the feature representation of style-transferred data, with the global interpolation style as positive anchors for each data point in $\mathcal{D}_i$. Through a contrastive manner, the local client learns a feature extractor that aligns its local data and the style-transferred data. Specifically, this learning mechanism improves generalizability by treating samples from the same class with original and transferred styles as positive pairs and samples from different classes with both styles as negative pairs. The global interpolative style, denoted as $\mathcal{S}_g$, serves as an unbiased aggregation of the statistical information across all client domains. By encompassing multi-domain characteristics, $\mathcal{S}_g$ provides a balanced and inclusive representation that acts as a guiding framework for achieving a fair and convergent learning objective. This strategy enhances the model's ability to capture more refined and discriminative class-level features while simultaneously encouraging the learning of domain-invariant features. As a result, it improves generalization performance and mitigates biases caused by domain heterogeneity. To achieve style-transferred data $\mathcal{D}_i'$ given $\mathcal{D}_i$ and $\mathcal{S}_g$, we use pre-trained AdaIN model, in which:

$$\mathrm{AdaIN}\left(F_{\mathcal{D}_i}, \mathcal{S}_g\right) = \sigma\left(\mathcal{S}_g\right)\left(\frac{f(\mathcal{D}_i) - \mu\left(f(\mathcal{D}_i)\right)}{\sigma\left(f(\mathcal{D}_i)\right)}\right) + \mu\left(\mathcal{S}_g\right), \tag{6}$$

where $f(\mathcal{D}_i)$ is the representation of $\mathcal{D}_i$ in feature space, and $\mu(.)$ and $\sigma(.)$ are the channel-wise mean and standard variance of image features, respectively. Given a training batch $\mathcal{B} = \{z_i, y_i\}$ with z_i, y_i are the feature representation and label of i^{th} sample, respectively, we denote the corresponding style-transferred batch as $\mathcal{B}_p$ such that $\mathcal{B}_p = \{\mathrm{AdaIN}\left(z_i, \mathcal{S}_g\right), y_i| \ \forall z_i \in \mathcal{B}\}$.

To leverage this style-transferred data, we use triplet loss [37] to guide clients to learn from embedding following a globally agreed style. In action, for each training batch, PARDON would treat all representations of the ground-truth samples as the set of anchors and build up their corresponding positive and negative sets. Specifically, the positive sample for anchor z_i is defined as its corresponding style-transferred embedding, i.e., $z_i^p := z_i' \in \mathcal{B}_p$, whereas the negative sample is style-transferred embedding from other classes. Consider the set of negative anchors as: $\mathcal{N}_i = \{z_j | z_j \in \mathcal{B}_p \wedge y_j \neq y_i\}$. Then, the triplet loss function is expressed as follows:

$$\mathcal{L}_T = \sum_{i=1}^{|\mathcal{B}|}\left[\|z_i^a - z_i^p\|_2^2 - \frac{1}{|\mathcal{N}_i|}\sum_{z_i^n \in \mathcal{N}_i}\|z_i^a - z_i^n\|_2^2 + \alpha\right], \tag{7}$$

Here, z_i^a, z_i^p, z_i^n are feature representations of the current sample x_i, its positive and negative anchors, respectively, and α is the margin value.

Intra-client domain diversity learning. In addition to distinguishing inter-domain representations, each client should have the capability to effectively learn from its local data, i.e., intra-client domain diversity. To achieve this, we incorporate the CrossEntropy [6] loss, utilizing the logits output along with the original annotation signal (y_i) to maintain local domain discriminatory power. This is formulated as: $\mathcal{L}_{CE} = -\mathbf{1}_{y_i}\log\left(\sigma\left(g\left(z_i\right)\right)\right)$, where σ denotes the softmax function. Since the cross-entropy loss term already encourages the model to leverage intra-client domain diversity, we use an additional mechanism to ensure the model can effectively learn from data across various domains without overfitting its local data. To address this, we impose restrictions on the amount of information the model can encode by applying L2 regularization directly to the representations, rather than the network parameters. This approach limits the complexity of the learned embeddings and enhances their generalizability [27, 49].

Specifically, we incorporate L2 regularization, denoted as $\mathcal{L}_{reg}$, to prevent overfitting and encourage bounded embeddings in contrastive learning. The regularization term is defined as:

$$\mathcal{L}_{reg} = \sum_{i=1}^{|\mathcal{B}|}\|z_i\|_2^2 + \|z_i^p\|_2^2, \tag{8}$$

where $\mathcal{B}$ represents the batch, and z_i is the representation of the i-th sample. This regularization encourages embeddings to remain within a bounded range, thereby improving the model's ability to generalize across diverse domains.

Finally, each participant C_i optimizes its local data during local updates by minimizing the following objective:

$$\mathcal{L} = \mathcal{L}_{CE} + \gamma_1\mathcal{L}_T + \gamma_2\mathcal{L}_{reg}, \tag{9}$$

where γ_1 and γ_2 are coefficient terms that control the strength of the inter-domain learning and intra-domain regularization.

Aggregation. We adopt the original average aggregator from most FL frameworks, in which the global model G is calculated by $G = \frac{1}{N}\sum_{i=1}^{K} n_i G_i$, where $n_i = |\mathcal{D}_i|$ is the data size of client $\mathcal{C}_i$ and $N = \sum_i^K n_i$.

To this end, we hypothesize that PARDON can achieve state-of-the-art generalizability in complex and practical scenarios with domain-based client heterogeneity and client sampling, while preserving local data privacy.

IV. Experiments

In this section, we compare our method with five FedDG baselines under FL settings concerning varied training domains, number of clients, and domain-based client heterogeneity, then demonstrate PARDON can achieve improved generalizability and security in complicated scenarios involving domain-based client heterogeneity and client sampling. We conduct all the experiments using PyTorch 2.1.0 [13] and the framework provided by Bai et al. [2] to simulate domain-based client

heterogeneity. All experiments are run on a computer with an Intel Xeon Gold 6330N CPU and an NVIDIA A6000 GPU.

A. Experimental Setup

Datasets. We evaluate our methods on three diverse classification tasks, each represented by a specific dataset. The first dataset is PACS [17], encompassing four distinct domains: Photo (P), Art (A), Cartoon (C), and Sketch (S). PACS comprises a total of seven classes. The second dataset, Office-Home [44], involves four domains with 65 classes. The four domains are Art, Clipart, Product, and Real-World. The third dataset is IWildCam [15], a real-world image classification dataset based on wild animal camera traps aroudn the world, where each camera represents a domain. It contains 243 training, 32 validation, and 48 test domains with a total of 182 classes.
FL Simulation. Based on the framework provided by [2], we simulate an FL system with a total of N clients; in each training round, the server will randomly select $k\%$ of clients to participate, which is known as *client sampling*. The data distribution of each client is simulated by *domain-based client heterogeneity* by λ level. We establish a rigorous FL scenario by adjusting the ratio of participating clients k to total clients (k/N) and heterogeneity degree (λ). Unless otherwise noted, our default parameters are $N = 100$ and $k = 20\%$ for PACS/Office-Home, and $N = 243$ and $k = 10\%$ for IWildCam, with a heterogeneity level of $\lambda = 0.1$. We set the number of communication iterations to 50 rounds for PACS/OfficeHome and 100 for IWildCam, and the number of local training epochs to 1. The batch size is set to 32 for all three datasets.
Baselines. We compare ours against five SOTA methods focusing on addressing domain generalization in FL, representative of three FedDG approaches in related works.

B. Experimental Results

1) Comparison with State-of-the-Art.: First, we evaluate FedDG methods under different validation schemes by varying the domain(s) used for training. Specifically, we consider two evaluation methods: Leave-One-Domain-Out (LODO) [11, 27] and Leave-Two-Domains-Out (LTDO) [2] with PACS and OfficeHome datasets.

LTDO Scenarios. Table I shows the final accuracy measurements with popular SOTA methods by the end of the FL process. We leave out two domains in each scheme alternately for validation and test accuracy. These results suggest that FedSR is not well-suited in scenarios where each client contains a small amount of data (e.g., with a large number of clients), which aligns with observations in previous work [2]. Other methods, such as FedGMA, FPL, and CCST, can somewhat generalize the FL model, but show degradation when training on domains less representative of the unseen domain, such as training on *Photo* and testing on *Cartoon*. On the other hand, our method performs substantially better than other methods, confirming that it generalizes well and thus effectively boosts performance on different unseen domains. Significantly, in the best case, our method outperforms with a gap of 3.64% compared to the second-best method for the PACS dataset.

LODO Scenarios. Table II presents the accuracy of compared methods with LODO experiments, i.e., we leave one domain out and use three domains for training. Generally, all methods achieve higher accuracy when more domains are used for training. Our method consistently performs best with PACS and OfficeHome datasets by achieving an average of 80.37% and 64.35% accuracy. Especially with domains such as Cartoon and Sketch of PACS dataset, we observe a large gap between our method and other baselines; our method outperforms the second-best method (i.e., FPL) by 4.63% and 4.12%, respectively.

2) Robustness Evaluation: We then study the robustness of PARDON w.r.t different domain heterogeneity, i.e., λ and number of clients K/N.
Impact of Domain Heterogeneity. This experiment aims to study the effectiveness of different methods with different domain distribution settings. As shown in Figure. 3, the larger the value of λ is, the more heterogeneous domain distribution is among clients. We observe the accuracy when varying the domain heterogeneity using $\lambda = \{0.0, 0.1, 0.5, 1.0\}$ values. Figure. 3 shows that our method consistently outperforms other baselines under varied domain heterogeneity. Interestingly, our method also achieves the highest accuracy early in training, creating a substantial gap over other baselines.

Next, we evaluate FedDG methods on IWildCam, which is a challenging setting with 323 domains and 243 clients, and present the results in Table III. As observed in real-world data, the performance of FedDG methods significantly degrades as λ decreases, consistent with findings by Bai et al. [2]. This degradation is particularly evident in baselines such as FedGMA, CCST, and FedDG-GA, where a lack of domain overlap ($\lambda = 0.0$) leads local models to converge to biased solutions. Under these conditions, these baselines suffer performance drops ranging from 35% to 50%. Notably, CCST, the second-best method on small-domain datasets, struggles remarkably in this challenging scenario. In contrast, our method exhibits robustness, experiencing only a 19% performance drop in such scenarios. It achieves the best average accuracy across varying levels of heterogeneity on both validation and testing sets, demonstrating its effectiveness and stability under diverse conditions. We argue that stability across varied levels of domain heterogeneity is a crucial requirement for practical FedDG methods.
Impact of Number of Clients. The stability of different methods in federated learning (FL) depends on the number of clients, which can range from small (around ten) to large (hundreds or thousands). In many FL schemes, the server randomly selects K participants from a total of N in each training round, especially with many participants. We consider five cases: $N \in \{5, 10, 50, 100, 200\}$ and $K = 5$, which correspond to 100%, 50%, 10%, 5%, 2.5% clients participating in each round, respectively. We present the comparison of different methods in Figure. 5. The higher the ratio of K/N is, the larger the amount of data participating in each training round is. The scenario of $5/5$ is the most similar setting to previous work, i.e., the number of clients is small, and all clients join each training round. While other methods, such

TABLE I: Comparison of inter-domain performance with SOTA methods under different split LTDO schemes, i.e., two domains are used for validation. AVG denotes the average accuracy calculated from each domain. (The best average accuracy is marked in **bold-underline**. The second-best average accuracy is marked in underline.)

Dataset	Methods	Validation Accuracy					Test Accuracy				
		A	P	C	S	AVG	P	S	A	C	AVG
PACS	FedSR	14.80%	14.67%	13.39%	13.36%	14.06%	13.80%	13.97%	14.55%	12.87%	13.80%
	FedGMA	39.31%	94.13%	63.95%	36.22%	58.40%	73.83%	64.85%	73.10%	52.73%	66.13%
	FPL	77.93%	94.49%	64.97%	31.61%	67.25%	93.53%	55.97%	62.01%	51.83%	65.84%
	FedDG-GA	64.99%	92.46%	63.18%	32.73%	63.34%	84.19%	63.55%	61.87%	48.08%	64.42%
	CCST	68.51%	96.41%	59.26%	35.68%	64.97%	86.89%	59.91%	71.78%	50.94%	67.38%
	Ours	73.63%	95.57%	69.41%	35.91%	**68.63%**	93.05%	66.20%	71.73%	53.11%	**71.02%**
		C	A	R	P	AVG	A	P	C	R	AVG
OfficeHome	FedSR	1.40%	1.24%	1.36%	1.31%	1.33%	1.15%	1.14%	1.34%	1.33%	1.24%
	FedGMA	43.18%	54.92%	66.81%	54.29%	54.80%	55.71%	66.43%	39.91%	56.83%	54.72%
	FPL	45.72%	56.82%	69.45%	46.18%	54.54%	59.95%	65.22%	43.99%	52.54%	55.43%
	FedDG-GA	38.99%	51.38%	63.85%	48.07%	50.57%	51.63%	62.38%	36.68%	54.79%	51.37%
	CCST	44.81%	52.48%	62.29%	49.85%	52.36%	52.20%	62.79%	38.37%	54.88%	52.06%
	Ours	46.74%	58.84%	71.13%	55.31%	**58.01%**	60.09%	67.54%	45.41%	61.62%	**58.67%**

PACS: *A: Art-Painting, P: Photo, C: Cartoon, S: Sketch*

OfficeHome: *C: Clipart, A: Art, R: Real World, P: Product*

TABLE II: Performance (%) comparisons with the start-of-the-art FedDG methods with the PACS dataset under LODO schemes, i.e., one domain is left for validation.

Methods	PACS					OfficeHome				
	P	A	C	S	AVG	P	A	C	R	AVG
FedSR	14.01%	13.27%	15.66%	13.49%	14.11%	1.14%	1.27%	1.34%	1.68%	1.36%
FedGMA	91.08%	81.25%	66.04%	61.72%	75.02%	69.99%	61.60%	49.87%	71.93%	63.35%
FPL	98.20%	81.98%	69.41%	62.64%	78.06%	68.98%	63.00%	48.98%	71.24%	63.05%
FedDG-GA	97.19%	76.17%	58.67%	53.42%	71.36%	66.03%	41.49%	45.98%	64.22%	54.34%
CCST	97.07%	84.18%	72.44%	59.43%	78.28%	66.93%	56.82%	47.35%	69.11%	60.05%
Ours	97.78%	83.89%	73.04%	66.76%	**80.37%**	71.14%	62.22%	50.70%	73.35%	**64.35%**

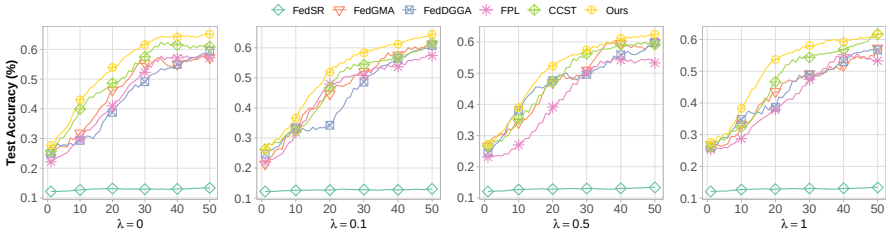


Fig. 3: Convergence curve when the training round increases on PACS's *Sketch*, and the training domains are *Art-Painting* and *Cartoon*; decreasing domain heterogeneity from left to right. $\lambda = 0$ means domain separation while $\lambda = 1.0$ means homogeneous domain distribution.

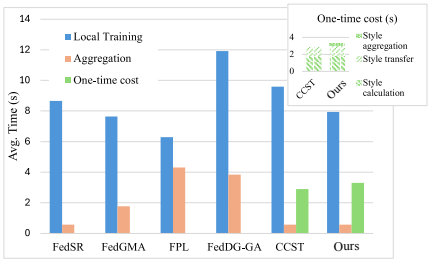


Fig. 4: Computational overhead of FedDG methods.

as FPL and CCST, exhibit strong performance with a small number of clients, their efficacy diminishes with an increase in the total number of clients. In contrast, our method outperforms in terms of stability and efficiency.

3) Scalability Study: In this section, we study the scalability of PARDON under two dimensions, which are computational overhead and security preservation.

Computational Overhead. For the computation time measurement, we used consistent settings for all baselines, including the number of clients, each client's local data, and the indices of clients selected in each round. Then, for the local training time, we averaged the training time for all clients in each round for a fair comparison of each baseline. We break down the computational time into three sub-components, which are (i) local training at each client, (ii) aggregation time at the server, and (iii) the remaining one-time cost. As presented in Fig. 4, the average computation time PARDON is comparable to or even smaller than other baselines. First, PARDON does not introduce additional overhead during aggregation, which linearly increases concerning training rounds as in FedDGGA, FedGMA, or FPL. The overhead of average local training is comparable to other methods. The cost for interpolation style calculation is a one-time cost, happening before the training and only taking 3.3s, while the general local training takes 8.67s on

TABLE III: Accuracy (%) comparisons after 100 training rounds of the different methods with the IWildCam dataset, whose 243 training, 32 validation, and 48 test domains.

Methods	Validation Accuracy				Testing Accuracy			
	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 1.0$	AVG	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 1.0$	AVG
FedSR	3.12%	32.97%	5.57%	13.89%	1.87%	16.04%	2.76%	6.89%
FedGMA	39.26%	71.17%	74.65%	61.69%	16.14%	58.75%	59.45%	44.87%
FPL	45.61%	70.45%	72.44%	62.83%	38.71%	54.29%	60.07%	51.02%
FedDG-GA	37.33%	67.98%	71.72%	59.01%	34.97%	59.29%	56.85%	50.37%
CCST	20.89%	66.20%	70.62%	52.57%	38.01%	56.58%	58.95%	51.18%
Ours	54.26%	67.69%	73.55%	**65.17%**	44.03%	59.34%	60.58%	**54.65%**

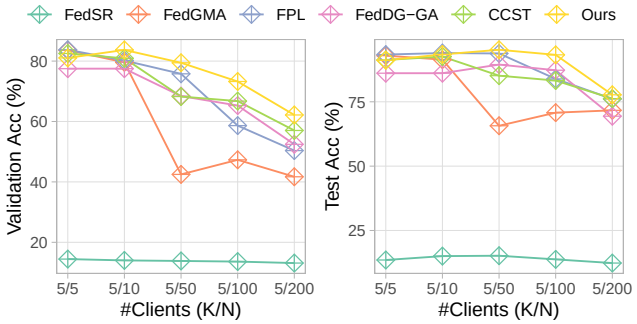
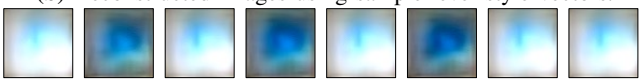


Fig. 5: Accuracy comparison with different settings of selected clients (K) per total number of clients (N).

(a) Original images were sampled from clients' training data.

(b) Reconstructed images using sample-level style vectors.



(c) Reconstructed clients' images using client-level style vectors.

Fig. 6: Image reconstruction from the GAN's models trained on the public dataset Tiny-ImageNet. The first row (Fig. 6a) is the real images sampled from clients' training data. The second row (Fig. 6b) shows the reconstructed images given the sample-level style vectors, i.e., vectors corresponding to each image in the first row. The last row (Fig. 6c) presents images reconstructed given client's style vectors from different clients.

(a) Reconstructed images using sample-level style vectors.

(b) Reconstructed images using client-level style vectors.

Fig. 7: Image reconstruction from the GAN's models trained on the malicious client's dataset. The first row (Fig. 6b) shows the reconstructed images given the sample-level style vectors, i.e., vectors extracted from the original images. The last row (Fig. 6c) presents images reconstructed given clients' style vectors from different clients.

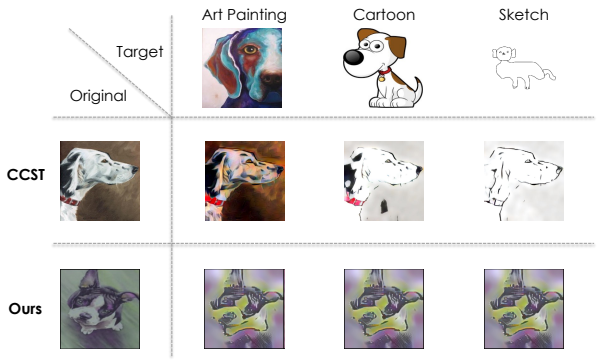


Fig. 8: Comparison of style-transferred images from ours and cross-client style transfer.

average for all methods. This one-time cost does not increase linearly with the number of clients since all clients can conduct local style extraction simultaneously, and the clustering on the server is fast (only 0.3s). Therefore, PARDON introduces low overhead when applied to large-scale systems. We conclude that we PARDON can ensure scalability on a large scale, where the number of clients can reach thousands.

Security Analysis. In our settings, there are two primary potential privacy breaches: (i) third-party reconstruction, where an external attacker or even the server compromises the shared style vectors to reconstruct private client datasets, and (ii) inter-client reconstruction, where a malicious client uses its data to reconstruct the images of other clients from their style vectors. First, we conduct experiments to assess the potential of third-party/server reconstruction attacks, i.e., if an adversary compromises the style vectors and wants to reconstruct private training images from the shared style vectors using generative models [21]. This analysis is crucial as local data includes sensitive medical records like X-rays and MRIs with identifiable details, where exposure risks stigma, discrimination, and job or

TABLE IV: Comparison metrics for the quality of reconstructed images from generative models.

	Domain	FID (↑)		Inception Score (↓)	
		Sample Style	Client Style	Sample Style	Client Style
Attack (i)	P	203.24	422.75	0.29 ± 0.0025	0.16 ± 0.0022
	A	200.71	501.72	0.30 ± 0.0023	0.16 ± 0.0022
	C	224.50	475.19	0.30 ± 0.0016	0.16 ± 0.0022
	S	254.17	488.51	0.27 ± 0.0008	0.16 ± 0.0022
Attack (ii)	P	180.61	447.95	0.33 ± 0.0017	0.23 ± 0.0121
	A	193.51	483.31	0.35 ± 0.0016	0.23 ± 0.0121
	C	146.94	470.02	0.32 ± 0.0011	0.23 ± 0.0121
	S	203.43	502.01	0.28 ± 0.0008	0.23 ± 0.0121

insurance loss. We train the generator until the validation loss converges sufficiently, i.e., the loss is approximately reaching zero and barely changes. The best model with the highest validation average PSNR was selected. We visualized the reconstructed images for different clients in Fig. 6c. The result shows that the reconstructed images are far different from the real images, i.e., compared to images from Fig. 6a. In addition, in our approach, each client uses only one vector to represent their data style, which leads to only one image that can be reconstructed for one client, i.e., no distinguishable images reflecting different classes are available. Indeed, compared to the case where the style of samples is allowed to be shared, adversaries can exploit it to reconstruct more distinguishable images as in Fig. 6b. We conclude that the design of PARDON is more secure than other mechanisms allowing sample-level information exchange.

Regarding inter-client reconstruction, our method does not allow any cross-sharing mechanism between two arbitrary clients, which minimizes the privacy breaches between two clients compared to previous works [4, 23, 33]. We visualized the reconstructed images using sample-level styles and client-level styles (i.e., PARDON) in Fig. 7a and Fig. 7b, respectively. From the results, this attack is extreme because if the adversary can acquire a sample-level style vector (i.e., single style as in CCST), they can reconstruct the content of the image (cf. Fig. 7a). However, with PARDON method, the reconstructed images do not contain content information of the private data. We provide quantitative results in Table IV to further support our conclusion, where the images generated using PARDON's style vectors are much less informative and have lower quality than strategies using sample-level style. We conclude that reconstructing a client's data solely from its style vector is nontrivial and requires specialized methods tailored to the data reconstruction tasks given style vectors utilized in our approach.

Lastly, we present the visualization of style-transferred images generated by our method and CCST, a baseline that uses style transfer for data augmentation. As shown in Fig. 8, the images produced by PARDON across different target domains are visually indistinguishable. In contrast, the images generated by CCST exhibit distinguishable differences among domains, as sample-level styles are communicated. This results in transferred images that closely resemble the private datasets of other clients, unlike those generated by our method. By comparing the images, we conclude that interpolation-style transfer offers stronger privacy protection than cross-local style transfer, as demonstrated by the comparison with CCST [4].

TABLE V: Performance of PARDON with different components present – removed components (✗) / retained (✓).

Methods	Components			Performance	
	Local Clustering	Global Clustering	Contrastive Learning	Validation Accuracy	Test Accuracy
PARDON-v1	✗	✓	✓	72.90%	92.22%
PARDON-v2	✓	✗	✓	72.80%	92.18%
PARDON-v3	✓	✓	✗	64.89%	85.69%
PARDON-v4	✗	✗	✓	59.42%	82.33%
PARDON-v5	✓	✓	✓	**73.63%**	**93.05%**

In conclusion, the local style statistics component of PARDON does not introduce additional security risks; instead, it mitigates attack threats compared to the sample-level and cross-client sharing employed in other works.

4) Ablation Study: In Table V, we conduct an ablation study to investigate the impacts of the main components proposed in our PARDON approach. We gradually remove each component from PARDON architecture to construct four incomplete versions of PARDON. In the PARDON-v1 and PARDON-v2 versions, the ✗ cells indicate that simple averaging was used to calculate the local and global styles, rather than the clustering methods described earlier. The results demonstrate that clustering at both the local and global levels is essential in mitigating the bias introduced by domain heterogeneity, proving to be more effective than simple averaging. The performance limitations of the simple averaging approach underscore the necessity for clustering, with FINCH outperforming simple averaging in both versions. The contrastive learning component in PARDON-v3 is identified as the second-most critical element. Excluding this component leads to a significant 9% drop in accuracy, even when style-transferred data is added to the training. This result underscores the importance of the carefully designed combination of interpolation style and contrastive learning. However, it is important to note that the effectiveness of the contrastive learning component hinges on the use of interpolation style-transferred data. This is evident from the comparison between PARDON-v4 and PARDON-v5, where PARDON-v4—relying on standard contrastive learning with augmentation and close samples as positive anchors—fails to address domain shifts effectively under this scenario, unlike PARDON-v5, which utilizes interpolation style-transferred data. In conclusion, the PARDON-v5 version, which incorporates all components, demonstrates superior performance in handling unseen domain accuracy, reinforcing the value of the well-designed components of PARDON.

V. Conclusions

This work presents PARDON, a domain generalization method in FL. PARDON extracts an unbiased interpolative style from all clients to facilitate style transfer. Each client then performs style transfer using this information through contrastive learning, aiming to enhance generalizability and mitigate local bias. Consequently, the aggregated model achieves improved accuracy for unseen domains. Extensive experiments demonstrate PARDON's improved performance over existing FedDG baselines across various settings, from small to large domain datasets, and in scenarios with high

domain heterogeneity. In addition, using an interpolation style helps enhance the local privacy of clients compared to a cross-client style's information exchange mechanism.

REFERENCES

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] R. Bai, S. Bagchi, and D. I. Inouye. Benchmarking algorithms for federated domain generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[4] J. Chen, M. Jiang, Q. Dou, and Q. Chen. Federated domain generalization for image recognition via cross-client style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[5] M. Crawshaw, Y. Bao, and M. Liu. Federated learning with client subsampling, data heterogeneity, and unbounded smoothness: A new algorithm and lower bounds. *Advances in Neural Information Processing Systems*, 36, 2024.

[6] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134:19–67, 2005.

[7] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu. Client selection in federated learning: Principles, challenges, and opportunities. *IEEE Internet of Things Journal*, 2023.

[8] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. F. Xu. Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA*, pages 18–24, 2022.

[9] J. Guo, L. Qi, and Y. Shi. Domaindrop: Suppressing domain-sensitive channels for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19114–19124, 2023.

[10] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020.

[11] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[12] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[13] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.

[14] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[15] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[16] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[17] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[18] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[19] 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*, 2:429–450, 2020.

[20] J. H. Lim, S. Ha, and S. W. Yoon. Metavers: Meta-learned versatile representations for personalized federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

[21] B. Liu, Y. Zhu, K. Song, and A. Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International conference on learning representations*, 2020.

[22] L. Liu, Z. Liu, and N. Ruan. From the perspective of prototypes: A privacy-preserving personalized federated learning framework. In *International Conference on Information Security Practice and Experience*, pages 222–239. Springer, 2024.

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[24] 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*, pages 1273–1282. PMLR, 2017.

[25] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

[26] J. Mitrovic, B. McWilliams, J. C. Walker, L. H. Buesing, and C. Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021.

[27] A. T. Nguyen, P. Torr, and S. N. Lim. Fedsr: A simple

and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022.

[28] A. T. Nguyen, T. Tran, Y. Gal, and A. G. Baydin. Domain invariant representation learning with domain density transformations. *Advances in Neural Information Processing Systems*, 34:5264–5275, 2021.

[29] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3), 2021.

[30] N. H. Nguyen, P. L. Nguyen, T. D. Nguyen, T. T. Nguyen, D. L. Nguyen, T. H. Nguyen, H. H. Pham, and T. N. Truong. Feddrl: Deep reinforcement learning-based adaptive aggregation for non-iid data in federated learning. In *Proceedings of the 51st International Conference on Parallel Processing*, pages 1–11, 2022.

[31] Z. Niu, J. Yuan, X. Ma, Y. Xu, J. Liu, Y.-W. Chen, R. Tong, and L. Lin. Knowledge distillation-based domain-invariant representation learning for domain generalization. *IEEE Transactions on Multimedia*, 2023.

[32] J. I. Orlando, H. Fu, J. B. Breda, K. Van Keer, D. R. Bathula, A. Diaz-Pinto, R. Fang, P.-A. Heng, J. Kim, J. Lee, et al. Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. *Medical image analysis*, 59:101570, 2020.

[33] J. Park, D.-J. Han, J. Kim, S. Wang, C. Brinton, and J. Moon. Stablefdg: style and attention based learning for federated domain generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] V. Piratla, P. Netrapalli, and S. Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, pages 7728–7738. PMLR, 2020.

[35] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[36] S. Sarfraz, V. Sharma, and R. Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8934–8943, 2019.

[37] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[38] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[39] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598, 2020.

[40] D. Shenaj, E. Fanì, M. Toldo, D. Caldarola, A. Tavera, U. Michieli, M. Ciccone, P. Zanuttigh, and B. Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 444–454, 2023.

[41] Y. Shi, J. Seely, P. Torr, S. N, A. Hannun, N. Usunier, and G. Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

[42] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016.

[43] I. Tenison, S. A. Sreeramadas, V. Mugunthan, E. Oyallon, I. Rish, and E. Belilovsky. Gradient masked averaging for federated learning. *Transactions on Machine Learning Research*, 2023.

[44] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

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

[46] J. Wang, J. Chen, J. Lin, L. Sigal, and C. W. de Silva. Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by gaussian-guided latent alignment. *Pattern Recognition*, 116:107943, 2021.

[47] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[48] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[49] Y. Xu, Z. Zhong, J. Yang, J. You, and D. Zhang. A new discriminative sparse representation method for robust face recognition via $l\_\{2\}$ regularization. *IEEE transactions on neural networks and learning systems*, 28(10):2233–2242, 2016.

[50] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10174–10183, 2022.

[51] R. Zhang, Q. Xu, J. Yao, Y. Zhang, Q. Tian, and Y. Wang. Federated domain generalization with generalization adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[52] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. Domain generalization with mixstyle. In *International Conference*

on Learning Representations, 2021.