FedDG-MoE: Test-Time Mixture-of-Experts Fusion for Federated Domain Generalization

Ahmed Radwan Mahmoud Soliman Omar Abdelaziz Mohamed Shehata
University of British Columbia
3333 University Way, Kelowna, BC
{ahmedm04,mosama97,oabdelaz}@student.ubc.ca, mohamed.sami.shehata@ubc.ca

Abstract

Federated Domain Generalization (FDG) aims to learn a model from multiple distributed source domains that generalizes well to unseen target domains. While transformers have achieved remarkable success in various computer vision tasks, their application to FDG is hindered by their large size, leading to high communication costs and demanding substantial data for effective fine-tuning. Inspired by the recent progress in parameter-efficient fine-tuning, we propose a novel approach for FDG that leverages Mixture of Experts (MoE) within a federated learning framework. Specifically, we employ a frozen, pre-trained vision transformer (ViT) as a backbone and introduce trainable MoE adapters based on Kronecker products at each client. This allows us to train only a small fraction of parameters, significantly reducing both computational and communication overhead. Furthermore, the MoE architecture promotes diverse feature learning, which is crucial for generalization to unseen domains. Crucially, during inference, we dynamically combine the client-specific MoE adapters based on a novel test-time weighting scheme. This weighting is determined by calculating the cosine similarity between the feature statistics of a given test batch and those tracked at each client during training. We demonstrate empirically that our approach, outperforms existing methods that rely on extensive fine-tuning of large pre-trained models, highlighting the efficacy of parameter-efficient fine-tuning and test-time adaptation for FDG. Our results highlight the crucial role of pre-trained features and the advantages of using MoE adapters in the context of federated domain generalization. The code is available at: https://github.com/AhmedMostafaSoliman/FedDG-MoE/

1. Introduction

Federated Learning (FL) [27] has emerged as a powerful paradigm for training machine learning models collaboratively across decentralized datasets residing on multiple client devices (e.g., mobile phones, hospitals). This approach offers significant privacy advantages by enabling model training without direct data exchange. However, a key challenge in FL is the heterogeneity of data distributions across clients, which can significantly degrade the performance of the globally aggregated model. Federated Domain Generalization (FDG) [24] specifically addresses this issue, aiming to train a model that generalizes well to unseen target domains, whose data distributions differ from those seen during training. This is a crucial capability for real-world FL applications where models are deployed in diverse environments.

While most existing FDG methods focus on relatively lightweight convolutional neural networks (CNNs) [23, 40], recent advances in computer vision have demonstrated the superior performance of Vision Transformers (ViTs) [4] across a wide range of tasks. ViTs, however, present significant challenges in the FL context. Their large size leads to substantial communication overhead when transmitting model updates between clients and the server, a critical bottleneck in FL, especially with limited bandwidth. Furthermore, effective fine-tuning of ViTs typically requires massive datasets, which are often unavailable in federated settings where individual clients may have limited data. Simply applying standard FL algorithms like FedAvg [27] or FedProx [22] to ViTs can lead to poor generalization due to overfitting to the specific, and potentially biased, data distributions of the training clients.

Moreover, naively fine-tuning a pretrained ViT can disrupt the valuable knowledge acquired during pretraining. Recent research [3, 38] has shown that the feature representations learned during large-scale pretraining are often more aligned with the distributions of unseen test domains than the distributions of the specific source domains used for subsequent fine-tuning. This suggests that preserving the pretrained feature extractor is crucial for achieving good domain generalization.

Parameter-Efficient Fine-Tuning (PEFT) [11, 14] methods offer a compelling solution to these challenges. PEFT techniques, such as adapters [13], train only a small number of additional parameters while keeping the vast majority of the pretrained model frozen. This dramatically reduces communication costs in FL, mitigates overfitting due to limited client data, and, crucially, preserves the valuable pretrained feature representations. Inspired by the success of Mixture-of-Experts (MoE) models [7, 31] in learning diverse representations, we propose incorporating MoE adapters into the FL framework. MoEs can capture a wider range of features, potentially leading to improved generalization performance in unseen domains.

In this paper, we introduce **FedDG-MoE**, a novel Federated Domain Generalization framework that leverages a pretrained, frozen ViT as a shared feature extractor and client-specific MoE adapters. During training, each client trains its own MoE adapter, capturing domain-specific knowledge, while also tracking statistics (mean and variance) of the feature embeddings produced by the frozen ViT. At test time, when a new client with an unknown domain presents data, our method adaptively combines the learned MoE adapters from the training clients. This combination is based on the similarity between the feature distribution of the new input and the tracked statistics of each training client. Clients whose feature distributions are more similar to the input's distribution contribute more strongly to the final prediction, effectively leveraging the most relevant expertise from the federation. This differs significantly from standard FL aggregation methods that average client model updates without considering input-specific characteristics.

Our contributions are summarized as follows:

- We propose FedDG-MoE, a novel framework for Federated Domain Generalization that combines the strengths of pretrained ViTs, MoE adapters, and input-adaptive model aggregation.
- We leverage PEFT to address the communication and overfitting challenges of using large ViTs in FL, while preserving valuable pretrained knowledge.
- We introduce a novel test-time adapter fusion mechanism that dynamically combines client-specific MoE adapters based on the similarity between the input feature distribution and the tracked statistics of each training client.
- We empirically demonstrate that FedDG-MoE outperforms several strong baselines, including methods that fine-tune entire pretrained models and standard FL algorithms applied to PEFT models, highlighting the benefits of our input-adaptive approach.

2. Related Work

This section reviews existing literature relevant to our proposed FedDG-MoE framework. We structure the discussion by first introducing Federated Learning and its challenges, then progressing through general Domain Generalization, Parameter-Efficient Fine-Tuning, Mixture of Experts, and finally culminating in a focused discussion of Federated Domain Generalization.

2.1. Federated Learning

Federated Learning (FL) [27] enables collaborative model training across decentralized datasets without direct data sharing. This paradigm is crucial for privacy-sensitive applications. Foundational FL algorithms include FedAvg [27], which averages model weights across clients; FedProx [22], which adds a proximal term to mitigate client drift; and SCAFFOLD [17], which uses control variates to correct for client drift. While these methods work well for relatively homogeneous data distributions, they often struggle with significant data heterogeneity, a common challenge in real-world FL deployments.

Recently, there has been interest in leveraging large pretrained models within FL. Raha et al. [29] investigate the impact of different pre-trained architectures on Federated Domain Generalization (FDG) performance, finding that larger models can improve generalization when fully fine-tuned. However, full fine-tuning incurs substantial communication overhead, making it impractical for many FL scenarios. This motivates the need for more parameter-efficient approaches, such as those explored in our work.

2.2. Domain Generalization

Domain Generalization (DG) aims to train a model that performs well on unseen target domains, given data from multiple source domains. Unlike Federated Learning, traditional DG often assumes access to all source domain data simultaneously. Common centralized DG techniques include meta-learning approaches that simulate domain shifts during training [21], methods that encourage learning domain-invariant representations [23], and techniques that align data distributions across domains [9].

While effective in centralized settings, these traditional DG methods are not directly applicable to Federated Domain Generalization. A core principle of FL is that data cannot be shared or mixed across clients. Therefore, techniques that rely on accessing and manipulating data from multiple domains within a single mini-batch are infeasible in the federated context. This fundamental difference necessitates the development of specialized FDG methods.

2.3. Parameter-Efficient Fine-Tuning (PEFT)

Parameter-Efficient Fine-Tuning (PEFT) [11, 13] addresses the challenges of adapting large pre-trained models to new tasks while minimizing computational and memory costs. PEFT methods achieve this by updating only a small fraction of the model's parameters or by introducing a limited number of new trainable parameters. This is particularly beneficial in FL, where communication bandwidth is often limited.

Adapters [13] are a widely used PEFT technique, inserting small, task-specific modules within the layers of a pre-trained model. LoRA [14] achieves parameter efficiency by injecting low-rank trainable matrices. Kronecker adapters [5] have been explored in Natural Language Processing and have achieved great success. The Kronecker Adapter represents weight updates as Kronecker products of smaller matrices. The work in [12] further explores using Kronecker adapters, comparing them to different adapter approaches and finding that they offer stronger performance when used to adapt pre-trained vision transformers for out-of-distribution computer vision tasks. Our work builds upon these results, extending it to a Mixture-of-Experts framework. PEFT methods not only reduce computational and communication costs but also often improve generalization by preserving the valuable knowledge learned during pre-training [3, 38].

2.4. Mixture of Experts (MoE)

Mixture of Experts (MoE) models [7, 31] enhance model capacity and representation diversity by combining the outputs of multiple "expert" networks. A gating network dynamically selects which experts are activated for each input, allowing the model to specialize different parts of its parameter space for different types of data. This input-conditional routing has been shown to improve generalization performance, particularly in settings with heterogeneous data distributions. MoE approaches, while not typically categorized as strict PEFT, still contribute to parameter efficiency as not every expert is needed for a given input. Leveraging a mixture of parameter-efficient adapters has been utilized in work like [35], and [19]. Our work leverages the strengths of MoE by incorporating A mixture of Kronecker based-adapter in the Federate Domain Generalization (FedDG) realm. This challenges the existing state of the art on FedDG [41] and Centralized DG [37] that leverages prompt-based adaptation of pre-trained models.

2.5. Federated Domain Generalization (FDG)

Federated Domain Generalization (FDG) combines the challenges of FL and DG, aiming to learn models that generalize to unseen domains in a decentralized setting. PLAN [16] addresses FDG through prompt learning and aggregation. While effective, this approach is limited by the representational capacity of the prompts. MaPLe [18] proposed a multi-modal prompt learning method. FedCLIP [26] explores generalization and personalization of CLIP models. Our approach, FedDG-MoE, distinguishes itself by incorporating MoE adapters and a novel test-time adapter fusion mechanism. This allows for greater representational power compared to prompt-based methods and enables input-specific adaptation, leading to superior generalization performance, as demonstrated by our empirical results.

3. Proposed Method

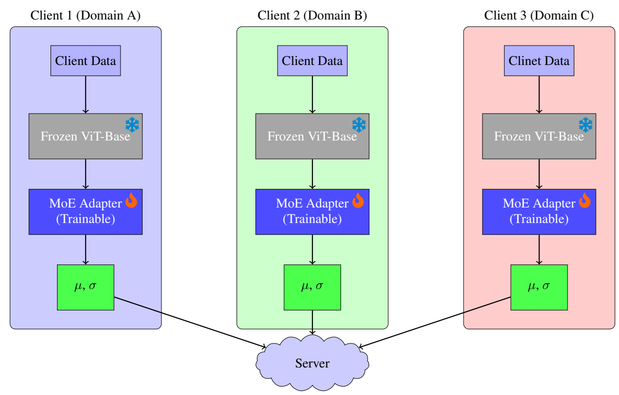

Figure 1. Federated Domain Generalization Architecture. Each client (Domains A, B, C) processes local data through a frozen ViT-Base model followed by a trainable MoE Adapter. The resulting parameters (μ, σ) are sent to the federated server for aggregation.

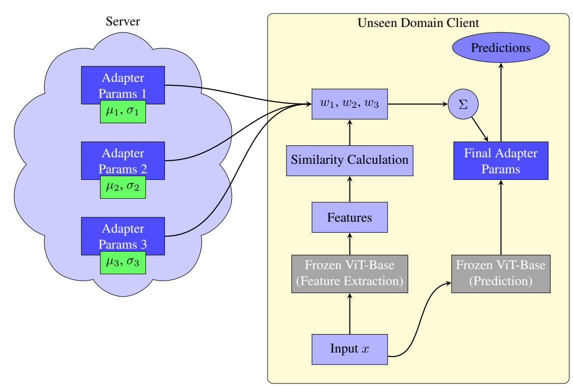

Figure 2. Federated Domain Generalization Inference Architecture. The server stores adapter parameters (μ, σ) from multiple domains. For an unseen domain, the client extracts features, calculates similarity to known domains, and creates weighted adapter parameters for final prediction while maintaining frozen ViT-Base backbones.

3.1. Preliminaries: Federated Domain Generalization

Federated Domain Generalization (FDG) aims to learn a model that performs well on unseen target domains, given training data from multiple, heterogeneous source domains. Formally, consider a federation of K clients, where each client $k \in \{1, \ldots, K\}$ possesses a local dataset $\mathcal{D}_k =$

$\{(x_i^k, y_i^k)\}_{i=1}^{n_k}$. Here, x_i^k represents an input sample (e.g., an image), and y_i^k is its corresponding label. The data distribution $P_k(X, Y)$ on each client k is assumed to be different, i.e., $P_i(X, Y) \neq P_j(X, Y)$ for $i \neq j$. The goal of FDG is to learn a model, parameterized by θ, that minimizes the expected loss on unseen target domains:

$$\min_{\theta} \mathbb{E}_{P_{test}(X,Y)}[\mathcal{L}(f_\theta(X), Y)],$$

where $\mathcal{L}$ is a loss function (e.g., cross-entropy), f_θ is the model, and $P_{test}(X, Y)$ represents the data distribution of an unseen target domain. Importantly, during training, we only have access to the source domain datasets $\{\mathcal{D}_k\}_{k=1}^K$, and we cannot access any data from the target domains. The objective in federated learning is typically achieved by optimizing the following objective:

$$\min_{\theta} \sum_{k=1}^{K} \frac{n_k}{N} \mathbb{E}_{(x,y)\sim P_k(X,Y)}[\mathcal{L}(f_\theta(x), y)].$$

Where $N = \sum_{k=1}^K n_k$. In standard federated learning, the global model weights are obtained as a weighted average of client models:

$$\theta = \sum_{k=1}^{K} \frac{n_k}{N} \theta_k$$

3.2. Mixture of Kronecker Product Expert Adapters

Inspired by the success of Mixture-of-Experts (MoE) in enhancing model capacity and representation diversity, and building upon the efficiency of Kronecker product decomposition for parameter-efficient fine-tuning [12], we propose a novel Mixture of Kronecker Product Expert Adapters (FedDG-MoE) approach.

Our method leverages a pretrained Vision Transformer (ViT) as a shared feature extractor. The ViT parameters are frozen throughout the training process to preserve the valuable knowledge acquired during pretraining and reduce communication overhead. We introduce client-specific MoE adapters that are inserted into the ViT architecture. Specifically, we replace the feedforward network (FFN) within each Transformer block with our MoE adapter layer.

The MoE adapter is designed to approximate the weight updates of a standard FFN layer in a parameter-efficient manner. Let $W \in \mathbb{R}^{d_{out} \times d_{in}}$ be the weight matrix of a standard FFN layer. In our MoE adapter, we approximate the weight update, ΔW, as a sum of n Kronecker products:

$$\Delta W = \sum_{i=1}^{n} A_i \otimes B_i,$$

where $A_i \in \mathbb{R}^{p \times q}$ are shared slow weights across all clients, and $B_i \in \mathbb{R}^{r \times s}$ are client-specific fast weights, with $d_{out} = p \cdot r$ and $d_{in} = q \cdot s$. This decomposition allows for sharing a significant portion of the parameters (A_i) across clients, while still allowing for client-specific adaptation through the B_i matrices. To further reduce the parameter count, each B_i is decomposed into two low-rank matrices $u_i \in \mathbb{R}^{r \times k}$ and $v_i \in \mathbb{R}^{k \times s}$, where k is a small rank, much smaller than r and s:

$$B_i = u_i v_i.$$

This low-rank decomposition significantly reduces the number of trainable parameters per client.

Furthermore, our MoE adapter incorporates a learned, input-dependent routing mechanism. This mechanism dynamically selects a subset of the Kronecker product components (experts) for each input. Given an input x, the router computes a set of n logits. These logits are passed through a softmax function to produce a weight distribution over the n experts. The output of the MoE adapter is then a weighted sum of the outputs of the individual experts:

$$y = \sum_{i=1}^{n} \alpha_i (A_i \otimes (u_i v_i))x + b,$$

where α_i is the softmax-normalized weight for expert i, and b is a shared bias parameter. This input-conditional weighting allows the model to specialize different experts for different types of inputs, promoting diverse representations and improving generalization. An auxiliary loss is added during training to encourage diversity in expert usage. The auxiliry loss ensures a balanced usage of experts, preventing the model from relying solely on a small subset of them.

By combining Kronecker product decomposition with a mixture-of-experts approach and a input-dependant routing, FedDG-MoE achieves a high degree of parameter efficiency while maintaining the capacity to learn diverse and specialized representations for improved domain generalization.

3.3. Test-Time Adapter Fusion

A core contribution of FedDG-MoE is its test-time adaptation capability, enabling generalization to unseen domains. This is accomplished via a dynamic adapter fusion mechanism that leverages client-specific distributional statistics gathered during training.

During training, each client k computes and maintains the mean (μ_k) and variance (σ_k^2) of the feature embeddings produced by the frozen ViT backbone over its local dataset, $\mathcal{D}_k$. These statistics, denoted concisely as $\mathcal{S}_k = \{\mu_k, \sigma_k^2\}$, characterize the client's data distribution. Crucially, before transmission to the server, each element of the mean vector μ_k is divided by the corresponding element of the standard deviation vector σ_k. This element-wise scaling, $\mu'_{k,j} = \frac{\mu_{k,j}}{\sigma_{k,j}}$, down-weights feature dimensions exhibiting high variance within a client's data, enhancing the robustness of subsequent similarity calculations.

At test time, given an input x_{test} from an unseen domain, the following procedure is employed:

Feature Extraction: The frozen ViT backbone extracts a feature embedding, $f(x_{test})$, from the input x_{test}.

Similarity Calculation: A modified cosine similarity is computed between the batch-averaged test features and the precomputed, scaled client statistics. First, features from the test batch, $\mathcal{B}_{test}$, are averaged: $\bar{f}(\mathcal{B}_{test}) = \frac{1}{|\mathcal{B}_{test}|}\sum_{x\in\mathcal{B}_{test}} f(x)$. Both the averaged test features and the stored client means are then normalized by each client's standard deviation:

$$\bar{f}'(\mathcal{B}_{test}) = \frac{\bar{f}(\mathcal{B}_{test})}{\sigma_k + \epsilon}, \quad \mu'_k = \frac{\mu_k}{\sigma_k + \epsilon},$$

where ϵ is a small constant for numerical stability. The cosine similarity, s_k, between the scaled, averaged test features, $\bar{f}'(\mathcal{B}_{test})$, and each scaled client mean, μ'_k, is then calculated. The purpose of this standard deviation scaling is to diminish the contribution of features that exhibit high variability within a given client's domain. This ensures that the similarity measure is more strongly influenced by features that are characteristic and consistent within each domain, rather than those with high intra-domain fluctuations. In our experiments, we ablate whether we should scale both the client mean μ_k, and the averaged test features $\bar{f}(\mathcal{B}_{test})$ or not.

Weight Calculation: The similarities $\{s_k\}_{k=1}^K$ are transformed into a set of normalized weights, $\{w_k\}_{k=1}^K$, via a softmax function with a temperature parameter, τ:

$$w_k = \frac{\exp(s_k/\tau)}{\sum_{j=1}^K \exp(s_j/\tau)}.$$

The temperature τ controls the sharpness of the weighting distribution.

Adapter Fusion: A test-sample-specific MoE adapter is constructed as a weighted linear combination of the trained client-specific MoE adapter parameters. Denoting the parameters of client k's MoE adapter as θ_k^{MoE}, the fused adapter parameters, θ_{test}^{MoE}, are given by:

$$\theta_{test}^{MoE} = \sum_{k=1}^K w_k \theta_k^{MoE}.$$

Prediction: Finally a forward pass with x_{test} is performed using the frozen ViT with θ_{test}^{MoE} to generate model predictions.

This procedure allows the model to adapt to the unseen domain by dynamically weighting the contributions of different clients based on their similarity to the test data distribution, thereby improving generalization performance. The pre-similarity averaging and single softmax application, combined with temperature scaling, provide a robust and efficient mechanism for test-time adaptation.

4. Experiments

This section details the experimental setup and results of evaluating our proposed FedDG-MoE framework. We first describe the datasets and implementation details, followed by a description of the baseline methods used for comparison.

4.1. Datasets and Implementation Details

We evaluate FedDG-MoE on three widely used benchmark datasets for domain generalization: OfficeHome, PACS, and VLCS.

OfficeHome [33] consists of approximately 15,500 images from four distinct domains: Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw). Each domain contains images of 65 object categories found in typical office and home settings.

PACS [20] (Photo, Art Painting, Cartoon, Sketch) is another popular domain generalization dataset with 9,991 images across four domains: Photo (P), Art Painting (A), Cartoon (C), and Sketch (S). It contains seven object categories.

VLCS [6] (VOC2007, LabelMe, Caltech-101, SUN09) is a domain generalization dataset that combines four photographic domains: VOC2007 (V), LabelMe (L), Caltech-101 (C), and SUN09 (S). It contains a total of 10,729 images distributed across five object classes.

For all datasets, we follow the standard federated domain generalization setup. Each domain is treated as a separate client. We perform leave-one-domain-out evaluation: for each dataset, we train on three domains (clients) and evaluate on the held-out fourth domain. This process is repeated for all possible combinations of training and testing domains.

4.2. Experimental Setup: Baseline Methods

To evaluate the performance of our Method, we compared our work against fifteen recent methods, categorized into four groups based on their underlying learning paradigms.

Centralized Learning (CL) for DG methods utilize data from all available domains in a centralized fashion to train a single, well-generalized model. We included SWAD [1], which exploits the relationship between flat minima and generalization using stochastic weight averaging; HCVP [15], which employs a hierarchical prompt generation system and contrastive learning within a Vision Transformer (ViT) to capture both domain- and task-specific features; and Doprompt [37], which generates image-specific prompts by first learning domain-specific prompts and then training a prompt adapter.

Boosting Federated Domain Generalization by Raha *et al.* [30]. This work provides a comprehensive evaluation of various architectures and pre-training strategies in the context of Federated Domain Generalization (FedDG). To ensure a focused and relevant comparison, we selected four key models from their extensive analysis: Swin-B, ViT-B, ConvNeXt-S, and ConvNeXt-B. These models were chosen because they represent a diverse range of state-of-the-art architectures (Transformer and ConvNet-based) and, as demonstrated in [30], exhibit strong performance in FedDG scenarios.

Conventional Federated Domain Generalization (FedDG) methods combine FL and DG principles to achieve both generalization to unseen domains and data privacy. The baselines in this category are FedAvg [27], the foundational FL algorithm, which aggregates a global model by averaging locally trained model updates; and FedProx [22], an extension of FedAvg that incorporates a proximal term to mitigate the effects of statistical heterogeneity across client data; FedSR [32], which employs L2-norm and conditional mutual information regularizations for a simple yet generalizable data representation; FedADG [36], which uses adversarial learning to align source domain distributions with a dynamically generated reference distribution; CCST [8], which transfers domain-specific styles across clients to capture a wider range of domain variations; ELCFS [28], which facilitates privacy-preserving data exchange in the frequency space to enhance generalizability; and GA [2], which dynamically recalibrates local model aggregation weights by minimizing the variance in generalization gaps across clients.

Finally, **Parameter-Efficient Fine-Tuning (PEFT)-based methods** leverage techniques like prompt learning and adapters to enhance the generalization of Vision-Language Models (VLMs) in FL scenarios. We included FedCLIP [25], which uses an attention-based adapter to tailor VLMs to individual clients; PromptFL [41], which trains and averages text prompts using local data;

Our implementation is based on a pre-trained Vision Transformer (ViT) model, specifically the CLIP-pretrained ViT-Base/16. We freeze the ViT backbone and only train the MoE adapter parameters and the classification head, as described in Section 3. We utilize the Adam optimizer with a learning rate of 0.001 and a batch size of 64. The federated learning process consists of 5 communication rounds, with each client performing 10 local epochs of training in each round. The temperature scaling parameter mentioned in section 3 is set to $\tau = 0.5$.

4.3. Main Results

The performance of our method, in comparison to baseline approaches, is documented in Table 1, Table 2, and Table 3 for the OfficeHome, PACS, and VLCS datasets, respec-

Table 1. Leave-one domain out evaluation results on OfficeHome

Algorithm	P	A	C	R	Avg
Centralized Algorithms					
SWAD	86.74	76.26	68.87	87.03	79.73
HCVP	88.01	81.77	69.76	90.62	82.54
Doprompt	88.94	80.95	70.88	90.10	82.72
Federated Algorithms					
Full Fine-Tuning					
FedAvg (ViT-B)	80.15	62.72	71.11	81.13	73.78
FedAvg (Swin-B)	87.88	68.72	80.01	88.17	81.20
FedAvg (ConvNeXt-S)	88.96	69.73	81.66	89.71	82.52
FedAvg (ConvNeXt-B)	89.81	73.31	84.42	90.31	84.46
Full Fine-Tuning (ViT-CLIP)					
FedProx	72.02	70.71	48.69	78.51	67.48
FedSR	72.71	69.12	49.69	79.12	67.66
FedADG	72.89	69.31	48.76	79.13	67.52
CCST	72.09	69.23	51.36	81.27	68.19
ELCFS	71.44	68.17	50.52	80.11	67.56
ELCFS+GA	73.35	68.62	50.60	81.23	68.45
PEFT (ViT-CLIP)					
FedCLIP	87.68	78.45	64.77	87.84	79.69
PromptFL	92.14	82.98	68.98	90.27	83.59
FedDG-MoE (Ours)					
FedDG-MoE (Avg)	94.62	85.62	81.58	92.68	88.63
FedDG-MoE (Scaffold)	94.50	85.17	81.90	**93.07**	88.66
FedDG-MoE (Prox)	94.64	85.29	82.02	92.84	88.70
FedDG-MoE (AM)	**94.84**	85.33	**82.70**	92.56	**88.86**
FedDG-MoE (TTF)	94.44	**85.99**	81.58	92.86	88.72

Table 2. Leave-one domain out evaluation results on PACS

Algorithm	P	A	C	S	Avg
Centralized Algorithms					
SWAD	99.18	93.23	85.93	82.03	90.44
HCVP	99.33	93.17	86.89	81.30	90.17
Doprompt	99.63	95.00	86.35	78.20	89.91
Federated Algorithms					
Full Fine-tuning					
FedAvg (ViT-B)	95.20	81.49	75.85	78.64	82.80
FedAvg (Swin-B)	99.52	93.26	85.36	85.64	90.94
FedAvg (ConvNeXt-S)	99.78	94.97	89.41	84.75	92.23
FedAvg (ConvNeXt-B)	99.76	95.60	88.09	86.73	92.55
Full Fine-tuning (ViT-CLIP)					
FedProx	97.43	83.89	68.39	64.21	78.48
FedSR	95.74	88.19	67.45	65.92	79.33
FedADG	98.09	82.93	65.42	65.36	77.95
CCST	98.29	87.02	74.57	65.84	81.43
ELCFS	98.14	86.77	73.21	65.16	80.82
ELCFS+GA	97.56	87.68	75.19	65.86	81.57
PEFT (ViT-CLIP)					
FedCLIP	99.76	96.19	97.91	85.85	94.93
PromptFL	99.58	96.34	98.46	92.19	96.64
FedDG-MoE (Ours)					
FedDG-MoE (Avg)	99.88	**98.05**	98.29	92.24	97.12
FedDG-MoE (Scaffold)	**99.94**	97.56	98.29	93.15	97.24
FedDG-MoE (Prox)	99.88	97.66	**98.72**	92.87	97.28
FedDG-MoE (AM)	99.88	97.71	98.51	93.03	97.28
FedDG-MoE (TTF)	99.94	98.10	98.46	**93.69**	**97.55**

Table 3. Leave-one-domain-out evaluation results on VLCS

Algorithm	V	L	C	S	Avg
Centralized Algorithms					
SWAD	75.40	68.36	98.49	79.49	79.31
HCVP	80.08	66.26	96.32	81.65	81.08
Doprompt	78.28	66.53	96.70	79.39	80.23
Federated Algorithms					
Full Fine-Tuning (ViT CLIP)					
FedAvg	78.21	65.63	95.48	73.31	78.16
FedProx	76.45	64.91	95.41	77.9	78.70
FedSR	78.51	65.86	95.16	73.49	78.26
FedADG	76.43	65.76	95.21	75.96	78.34
CCST	76.42	65.73	96.49	77.67	79.08
ELCFS	76.55	65.02	95.67	77.96	78.80
ELCFS+GA	78.89	65.16	96.77	78.93	79.18
PEFT (ViT CLIP)					
FedCLIP	73.28	66.98	99.93	87.14	81.83
PromptFL	72.24	68.03	99.65	85.10	81.26
FedDG-MoE (Ours)					
FedDG-MoE (Avg)	84.40	63.86	99.76	81.93	82.49
FedDG-MoE (Prox)	84.21	63.99	**100.00**	82.64	82.71
FedDG-MoE (Scaffold)	84.01	62.36	99.53	81.76	81.92
FedDG-MoE (AM)	**84.90**	61.98	99.76	**82.94**	82.40
FedDG-MoE (TTF)	84.70	**65.12**	**100.00**	81.83	**82.91**

tively. These tables provide a breakdown compared to results of other methods [1, 2, 8, 10, 15, 22, 25, 27, 28, 30, 32, 36, 37, 41]. The results presented in each table encompass five distinct variations of our proposed approach. The initial four rows detail the performance of our Mixture-of-Experts (MoE) method when integrated with various federated learning algorithms: FedAvg [27], SCAFFOLD [17], FedProx [22], and AM [34]. The final row in each table illustrates the results obtained using Test-Time Adapter Fusion (TTF).

4.3.1. OfficeHome Results

An analysis of Table 1 reveals that our methods achieves the highest performance across all domains within the OfficeHome dataset. Our Method outperforms these methods by at least 4.4%, in classification accuracy on OfficeHome.

4.3.2. PACS Results

An analysis of Table 2 reveals that our methods achieves the highest performance across all domains within the PACS dataset. Our Method outperforms these methods by at least 1.08%, in classification accuracy on PACS.

4.3.3. VLCS Results

An analysis of Table 3 reveals that our methods achieves the highest performance across all domains within the VLCS dataset. Our Method outperforms these methods by at least 1.08%, in classification accuracy on VLCS.

Table 4. Effect of the number of feature extraction layers (N) on OfficeHome.

N	P	A	C	R	Avg
1	94.37	85.46	81.72	93.05	88.65
2	94.44	**85.99**	81.58	92.86	**88.72**
3	94.28	85.00	81.95	92.86	88.52
4	94.46	85.17	**82.20**	**93.07**	**88.73**
5	**94.75**	85.00	82.02	92.70	88.62

Table 5. Effect of test batch size (BS) on OfficeHome.

Test BS	P	A	C	R	Avg
1	94.46	85.04	81.60	92.79	87.03
4	94.44	84.92	81.92	**93.14**	88.61
8	93.71	85.50	80.92	93.41	88.39
16	**94.66**	85.21	**82.04**	92.66	88.64
32	94.44	**85.99**	81.58	92.86	**88.72**
64	94.51	85.00	**82.04**	93.07	88.66

Table 6. Effect of softmax temperature scaling factor τ. evaluated on OfficeHome

τ	P	A	C	R	Avg
1	94.35	85.58	81.53	92.77	88.56
0.5	**94.44**	**85.99**	81.58	**92.86**	**88.72**
0.25	94.39	85.58	**81.79**	92.70	88.62

Table 7. Effect of Scaling down the test features $\bar{f}(\mathcal{B}_{test})$ evaluated on OfficeHome.

Scaling	P	A	C	R	Avg
True	94.44	85.99	**81.58**	**92.86**	**88.72**
False	**94.64**	84.92	81.42	92.70	88.42

Table 8. Effect of Scaling down the test features $\bar{f}(\mathcal{B}_{test})$ evaluated on PACS.

Scaling	P	A	C	S	Avg
True	99.88	97.95	**98.85**	92.62	97.33
False	**99.94**	**98.10**	98.46	**93.69**	**97.55**

4.4. Ablation Study

We demonstrate the effect of the different variants of the Test-Time Fusion model by conducting ablation experiments on OfficeHome dataset [33]. All ablations are compliant with the Leave-one-domain-out protocol.

Table 9. Effect of Scaling down the test features $\bar{f}(\mathcal{B}_{test})$ evaluated on VLCS.

Scaling	V	L	C	S	Avg
True	85.00	**63.36**	**100.00**	81.62	82.50
False	**87.96**	62.23	99.53	**81.73**	**82.86**

4.4.1. Number of Feature Extraction Layers

We investigated the impact of incorporating features from multiple layers of the frozen backbone, rather than relying solely on the final layer's output. Prior research suggests that earlier layers in deep neural networks tend to encode more domain-specific and stylistic information, while later layers capture higher-level semantic content [39]. Given that our test-time fusion mechanism relies on measuring the similarity between a test sample's features and pre-computed domain statistics, it is hypothesized that incorporating features containing domain specific information is beneficial.

Table 4 presents the results of this investigation. The best performing configurations are those that combine features from either the penultimate and final layers, or from a combination of four layers. This optimal performance suggests a nuanced relationship between feature layer depth and generalization ability in our framework. While earlier layers provide crucial domain-specific cues, the inclusion of deeper, more semantically rich features from later layers remains important for accurate classification. The results underscore the importance of balancing domain-specific information with higher-level semantic understanding for optimal test-time adaptation.

4.4.2. Test Batch Size

Table 5 presents the results of experiments conducted with varying test batch sizes. Our findings indicate that a test batch size of 32 yields optimal performance. This observation highlights a trade-off inherent in the test-time adapter fusion process. Smaller batch sizes allow for a more specialized fusion of adapters, potentially tailoring the model more precisely to individual or small groups of samples. However, smaller batches also reduce the statistical power for accurately estimating the batch mean, a crucial component of the similarity calculation. Conversely, larger batches provide more robust mean estimations but can dilute the specialization benefits of the adapter fusion. The superior performance at a batch size of 32 suggests an appropriate balance between these competing factors.

4.4.3. Temperature Scaling Factor

We examine the effect of the similarity softmax temperature scaling factor τ and report the results in Table 6. Accordingly, a τ of 0.5 allows for sharp-enough softmax similarities to capture the closest attributes for a test domain from a set of training domains.

4.4.4. Scaling Down Test Features

Scaling down both the test features and the client's mean using the computed standard deviation could lead to over regularization. We examine the effect of turning off the down scaling of test features and report the results on the 3 benchmarks. Table 7, 8, 9 shows the results on OfficeHome, PACS, and VLCS respectively. We do notice that on the easier benchmarks (PACS, and VLCS) over regularization hurts the performance. On the other hand, it is beneficial for more complex benchmarks like OfficeHome.

5. Conclusion

This paper introduces FedDG-MoE, a novel framework for Federated Domain Generalization that leverages the strengths of pre-trained Vision Transformers, Mixture-of-Expert adapters, and a test-time adapter fusion mechanism. By employing parameter-efficient fine-tuning with Kronecker product-based MoE adapters, FedDG-MoE significantly reduces communication overhead and mitigates overfitting, crucial challenges in applying large ViTs in federated learning settings. The MoE architecture promotes diverse feature learning, enhancing generalization to unseen domains. Crucially, our proposed test-time adapter fusion mechanism dynamically combines client-specific MoE adapters based on the similarity between the input feature distribution and tracked client statistics, enabling effective adaptation to new, unseen domains during inference.

Our extensive empirical evaluations on the OfficeHome, PACS, and VLCS datasets demonstrate that FedDG-MoE consistently outperforms a wide range of strong baselines, including methods that fine-tune entire pre-trained models and standard FL algorithms applied to PEFT models. These results highlight the effectiveness of our approach in preserving valuable pre-trained knowledge, learning diverse representations, and adaptively leveraging client-specific expertise for improved generalization. Ablation studies further validate the design choices within our framework, demonstrating the impact of the number of feature extraction layers, test batch size, and temperature scaling factor.

Future work will explore several directions. First, investigating more routing mechanisms within the MoE adapters could further enhance the specialization and diversity of learned representations. Second, exploring methods for dynamically adjusting the number of experts per client based on data availability and complexity could improve resource allocation and model efficiency. Finally, extending FedDG-MoE to other modalities, could broaden the applicability of our approach to a wider range of federated tasks. The intersection between Federated Domain Generalization, and the Mixture of Experts paradigm is a promising research avenue that can bring more efficient, and better performing models.

References

[1] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Gyu Yun, and Sangdoo Lee. Swad: Domain generalization by seeking flat minima. In *Advances in Neural Information Processing Systems*, pages 8856–8868, 2021. 5, 7

[2] Hong-You Chen and Wei-Lun Chao. Model-agnostic round-wise variance reduction for federated learning. *ICLR*, 2021. 6, 7

[3] Zheng Dai, Bolin Ni, Ziyang Jha, Richard Socher, and Caiming Xiong. Why do my representations look like yours? a deep dive into pre-training and fine-tuning. In *International Conference on Learning Representations*, 2022. 1, 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[5] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. Krona: Parameter efficient tuning with kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022. 3

[6] C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE international conference on computer vision*, pages 1657–1664, 2013. 5

[7] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021. 2, 3

[8] Junming Gao, Yuan Wu, Ruinan Zhao, Hualin Fu, Kai Li, and Yizhou Wang. Cross-client style transfer for improved federated domain generalization. In *2022 59th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2022. 6, 7

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Domain generalization for object recognition with multi-task autoencoders. *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 2

[10] Shuai Gong, Chaoran Cui, Chunyun Zhang, Wenna Wang, Xiushan Nie, and Lei Zhu. Federated domain generalization via prompt learning and aggregation. 2024. 7

[11] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Parameter-efficient fine-tuning of large-scale pre-trained language models. *arXiv preprint arXiv:2203.16153*, 2022. 1, 2

[12] Xuehai He, Chuanyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient model adaptation for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 817–825, 2023. 3, 4

[13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 3

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1, 3

[15] Shiping Hu, Kunjie Pan, Pengfei Huang, Jiayu Zhang, Xiaolin Zhang, Jingdong Ren, and Deyi He. Improving domain generalization by hcvp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20083–20092, 2023. 5, 7

[16] Jintao Huang, Zhixuan Zhang, Chun Mao, Jingdong Zhou, Fenglong Tang, Xu Zhang, Min Wei, and Heqing Zhou. Plan: Federated domain generalization via prompt learning and aggregation. *arXiv preprint arXiv:2307.15270*, 2023. 3

[17] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2, 7

[18] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Mubarak Shah. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19998–20008, 2023. 3

[19] Seungryong Kim, Gyuseong Lee, Wooseok Jang, Jinhyeon Kim, and Jaewoo Jung. Domain generalization using large pretrained models with mixture-of-adapters. *Available at SSRN 4786252*, 2025. 3

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 5

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Fedprox: Heterogeneity-aware federated learning with local regularization. *arXiv preprint arXiv:1812.06127*, 2020. 1, 2, 6, 7

[23] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Domain generalization with adversarial feature learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5400–5409, 2018. 1, 2

[24] Quande Liu, Cheng Qin, Ruoxu Cao, Defeng Shen, Qian Wang, and Hengshuang Chen. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1013–1023, 2021. 1

[25] Yu Lu, Mei Chen, Hwee Kuan Yu, Xian Zhang, Joo Hwee Tan, and Choon Seng Lau. Fedclip: Fast generalization and personalization for clip in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 481–489, 2023. 6, 7

[26] Zihan Ma, Fangru Zhou, Yang Zhang, Ruoming Shi, Yansong Chen, Jie Li, Chen Chen, and Meng Zhu. Fedclip: Fast

generalization and personalization for clip in federated learning. *arXiv preprint arXiv:2310.09738*, 2023. 3

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*, pages 1273–1282, 2017. 1, 2, 6, 7

[28] Xuanzhe Qiu, Zhi-Li Zhang, Qilong Gu, Yan Chen, Chao Wang, and Qun Yang. Efficient federated learning via privacy-preserving data synthesis in frequency domain. In *ACM International Conference on Multimedia (MM)*, 2022. 6, 7

[29] Avi Deb Raha, Apurba Adhikary, Mrityunjoy Gain, Yu Qiao, and Choong Seon Hong. Boosting federated domain generalization: Understanding the role of advanced pre-trained architectures. *arXiv preprint arXiv:2310.05497*, 2023. 2

[30] Avi Deb Raha, Apurba Adhikary, Mrityunjoy Gain, Yu Qiao, and Choong Seon Hong. Boosting federated domain generalization: Understanding the role of advanced pre-trained architectures. *arXiv preprint arXiv:2409.13527*, 2024. 6, 7

[31] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2, 3

[32] Blesson Varghese, Maruan Al-Shedivat, S. Seyed Mousavi, and Lovekesh Ramachandran. A federated approach to domain generalization using shannon entropy. *arXiv preprint arXiv:2110.00580*, 2021. 6, 7

[33] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sujoy Sethi. Deep hashing network for unsupervised domain adaptation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 5, 7

[34] Jingye Wang, Ruoyi Du, Dongliang Chang, and Zhanyu Ma. Domain generalization via frequency-based feature disentanglement and interaction. *CoRR*, abs/2201.08029, 2022. 7

[35] Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. *arXiv preprint arXiv:2205.12410*, 1(2): 4, 2022. 3

[36] Pengxin Wu, Shuang Wang, Yuesheng Gu, and Jiming Chen. A federated adversarial domain generalization framework for privacy-preserving smart healthcare. In *ICASSP*, 2021. 6, 7

[37] Qi Xu, Ruipeng Zhang, Ya Zhao, Zixian Chen, Jiangchao Yuan, and Xiangru Wang. Doprompt: Prompt learning for domain generalization. In *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 440–446. IEEE, 2022. 3, 5, 7

[38] Kang You, Yexuan Wang, Wenqi Li, Yuantao Zhou, Shu-Tao Han, Qinghua Jiang, Yuesheng Ye, and Mingsheng Long. Towards better serialization-free federated learning via representation similarity. *arXiv preprint arXiv:2310.06601*, 2023. 1, 3

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 8

[40] Yanbei Zhao, Ruimao Jia, Zhizheng Li, Jingjing Ma, Shuangping Chai, Yuan Wang, and Xiaowei Zheng. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 550–559, 2021. 1

[41] Weiming Zhuang, Yonggang Liu, Xin Chen, Yuan Tan, Xin Gan, Li Zhang, Bo Li, Shutao Han, Jingyang He, et al. Promptfl: Prompt-tuning for federated learning on text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9507–9517, 2023. 3, 6, 7