Scaling Vision Foundation Models with Federated Adapter Generalization

Jiaqi Wang[1] , **Jingtao Li**[2] , **Weiming Zhuang**[2] , **Chen Chen**[2] , **Fenglong Ma**[1] , **Lingjuan Lyu**[2]
[1]Pennsylvania State University, [2]Sony AI
{jqwang, fenglong}@psu.edu,
{Jingtao.Li, Weiming.Zhuang, chena.chen, lingjuan.lv}@sony.com

Abstract

Vision foundation models (FMs) like CLIP have exhibited exceptional capabilities in visual and linguistic understanding, particularly in zero-shot inference tasks. However, these models struggle with data that significantly deviates from their training samples, necessitating fine-tuning, which is often infeasible in centralized settings due to data privacy concerns. Federated learning (FL) combined with parameter-efficient fine-tuning (PEFT) offers a potential solution, yet existing methods face issues with domain-specific characteristics and out-of-domain generalization. We propose Federated Adapter Generalization (`FedAG`), a novel federated fine-tuning approach that leverages multiple fine-grained adapters to capture domain-specific knowledge while enhancing out-of-domain generalization. Our method uses quality-aware in-domain mutual learning and attention-regularized cross-domain learning to integrate domain-specific insights effectively. Experiments on the CLIP model with three domain-shifting datasets, ImageCLEF-DA, Office-Home, and DomainNet, demonstrate the superior performance of `FedAG` in both in-domain and out-of-domain scenarios.

1 Introduction

Vision foundation models (FMs), such as pretrained CLIP [Radford *et al.*, 2021] and its variants [Li *et al.*, 2023], have demonstrated superior capabilities in understanding visual concepts and their linguistic descriptions. They have been employed in a wide range of vision tasks, including image classification, especially for zero-shot inference, thanks to their large number of parameters and the extensive training data they leverage. However, these models still face challenges when confronted with input data significantly different from their training samples. Therefore, fine-tuning becomes essential. Traditional fine-tuning strategies are typically conducted in a centralized manner. However, this approach is often impractical, particularly for sensitive data like medical information, which is often distributed among different clients and cannot be shared. This distributed scenario significantly complicates the fine-tuning process for vision foundation models.

Recent studies have focused on addressing this challenge by combining federated learning (FL) with fine-tuning of vision foundation models, a technique known as **federated fine-tuning**. Existing approaches [Xiao *et al.*, 2023; Marchisio *et al.*, 2023; Chua *et al.*, 2023; Khalid *et al.*, 2023] typically aim to fine-tune these models without utilizing the entire model, often employing layer-drop techniques [Sajjad *et al.*, 2023] to compress a full model into a sub-model. The sub-model and an emulator are distributed to clients. Clients then update this compressed sub-model with their private data with the help of the emulator iteratively. The resulting sub-model is eventually incorporated back into the full model to complete the fine-tuning process. However, these compression techniques fail to maintain alignment between the fine-tuned compressed layers and the remaining layers, leading to performance degradation in the fine-tuned models.

Federated parameter-efficient fine-tuning (PEFT) techniques, such as FedCLIP [Lu *et al.*, 2023] and FedPETuning [Zhang *et al.*, 2023], have emerged to address the aforementioned problem. These approaches involve deploying the foundation model with an additional adapter on each client, which is then collaboratively trained like FedAvg [McMahan *et al.*, 2017]. The aggregated adapter is subsequently integrated into the foundation model to achieve fine-tuning. Despite their straightforward and effective nature, federated PEFT models still have several issues:

Indistinguishable in domain-specific charateristics. In real-world applications, the data collected by clients may exhibit different characteristics even for the same task. For instance, the stylistic realism of an image can vary across different forms of visual art, such as painting, photography, and digital art, leading to unique artistic expressions. However, existing models typically employ a single adapter to capture knowledge from mixed domains, resulting in a performance gap compared to domain-specific adapters. Figure 1 (a) illustrates the performance comparison between using a single adapter for fine-tuning and employing separate adapters for each domain on the CLIP model in a centralized manner, using the DomainNet dataset with three domains: "*clipart*", "*painting*", and "*real*". It can be observed that despite using data from all three domains to fine-tune the adapter, CLIP_{one} still performs worse than CLIP_{multi}, which fine-tunes each

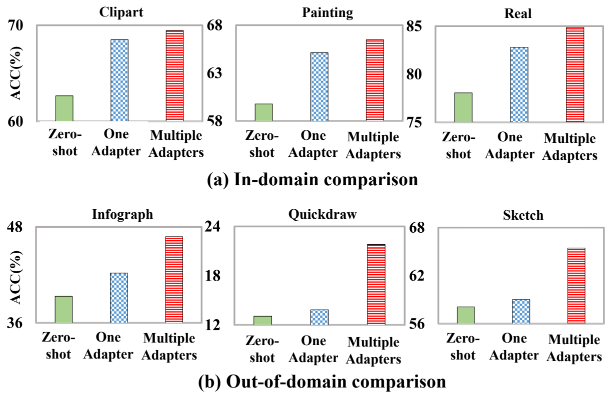

Figure 1: In-domain and out-of-domain preliminary results.

adapter using only domain-specific data. This issue is expected to exacerbate in the federated fine-tuning setting due to the heterogeneity of clients, leading to an aggregated adapter inferior to centralized fine-tuning. These initial findings motivate us to develop domain-specific adapters for use in federated PEFT.

Incapable to out-of-domain generalization. While existing federated fine-tuning approaches can improve performance compared to zero-shot inference on the original models, they still struggle when faced with new or out-of-domain data. To illustrate, consider the centralized fine-tuning on the DomainNet dataset, where we evaluate the original CLIP model (referred to as CLIP$_{zero}$) and the fine-tuned CLIP$_{one}$ on three new domains: "*infograph*", "*quickdraw*", and "*sketch*". Figure 1 (b) presents the results, with the performance of CLIP$_{multi}$ representing the upper bound. It can be observed that, while fine-tuning with a shared adapter (CLIP$_{one}$) does improve performance compared to CLIP$_{zero}$, the degree of improvement is limited, as the results are far from the performance achieved by CLIP$_{multi}$. Therefore, it is crucial to enhance the adapters' capability for out-of-domain generalization, especially in the federated fine-tuning setting.

However, addressing the aforementioned issue is challenging. On the one hand, it is hard to directly extend existing work to model domain-specific characteristics. Sub-model fine-tuning approaches encounter difficulties in compressing multiple domain-specific sub-models and aggregating them. Similarly, PEFT approaches face challenges in aggregating adapters with diverse knowledge. On the other hand, equipping the capability of out-of-domain generalization with federated fine-tuning is an open challenge in this domain and is still underexplored by existing studies. Thus, it is urgent to develop a new method to tackle these challenges simultaneously.

In this paper, we propose a novel federated fine-tuning approach named **Fed**erated **A**dapter **G**eneralization (`FedAG`). This approach employs multiple fine-grained adapters, allowing the injection of domain-specific knowledge into corresponding adapters while enhancing the capability of out-of-domain knowledge generalization by jointly combining these adapters. Unlike existing work, which either compresses a sub-model for each client or deploys a foundation model, we enable clients to have their domain-specific models representing the characteristics of their data. These client models are trained with private data and uploaded to the server to inject their domain-specific knowledge into the foundation model.

Specifically, each domain-specific client C_n with model parameters $\mathbf{W}_n^t$ has a corresponding adapter $\mathbf{A}_n^t$ at each communication round t. Domain-specific knowledge is aggregated into the adapter through a **quality-aware in-domain mutual learning** module, aided by a set of domain-specific synthetic data generated by Stable Diffusion [Rombach *et al.*, 2022]. To equip `FedAG` with the ability for out-of-domain generalization, we develop a novel **attention-regularized cross-domain learning** module, which attentively aggregates all domain-specific adapters with a novel regularizer controlling the domain weights. The updated client models are then distributed to the corresponding domains again for learning in the next communication round.

We conduct experiments in the cross-silo federated fine-tuning setting on the CLIP vision foundation model with three domain-shifting datasets: ImageCLEF-DA, Office-Home, and DomainNet. Experimental results demonstrate the effectiveness of `FedAG` on both in-domain and out-of-domain validations, performing close to or slightly better than the centralized fine-tuning baselines. Ablation studies and model insight analysis validate the reasonableness of our model design.

2 Related Work

2.1 Foundation Model in Federated Learning

Foundation models (FMs) [Bommasani *et al.*, 2021] have demonstrated strong capabilities across various domains, such as computer vision. However, the effectiveness of FMs is heavily dependent on large amounts of publicly available training data and the extensive size of model parameters. In real-world applications, this dependency raises several practical challenges: (1) suboptimal performance in specific domains due to limited access to relevant data, often restricted by privacy concerns; (2) the substantial size of the models necessitates significant computational resources, thereby limiting their applicability in various scenarios. Federated learning (FL)[McMahan *et al.*, 2017] presents a collaborative machine learning framework wherein clients can jointly train models without sharing their data, utilizing distributed computational resources. Several research efforts have explored the integration of FMs within FL [Chen *et al.*, 2024; Guo *et al.*, 2023; Lu *et al.*, 2023; Su *et al.*, 2024]. Additionally, multiple surveys[Zhuang *et al.*, 2023; Ren *et al.*, 2024; Woisetschläger *et al.*, 2024] have reviewed the advancements, open challenges, and future directions in this field.

2.2 Federated Fine-tuning of Foundation Models

To achieve better performance in specific domains, fine-tuning FMs with domain-specific data is essential. FL facilitates this fine-tuning process by allowing the use of locally stored data through distributed computational resources. Existing related research can be categorized into full FMFL tuning [Deng *et al.*, 2023; Fan *et al.*, 2023], partial FMFL tuning [Peng *et al.*, 2024; Marchisio *et al.*, 2022; Khalid *et al.*, 2023], and parameter-efficient FMFL fine-tuning [Lu *et al.*, 2023; Zhang *et al.*, 2023; Chua *et al.*, 2023]. Our work falls

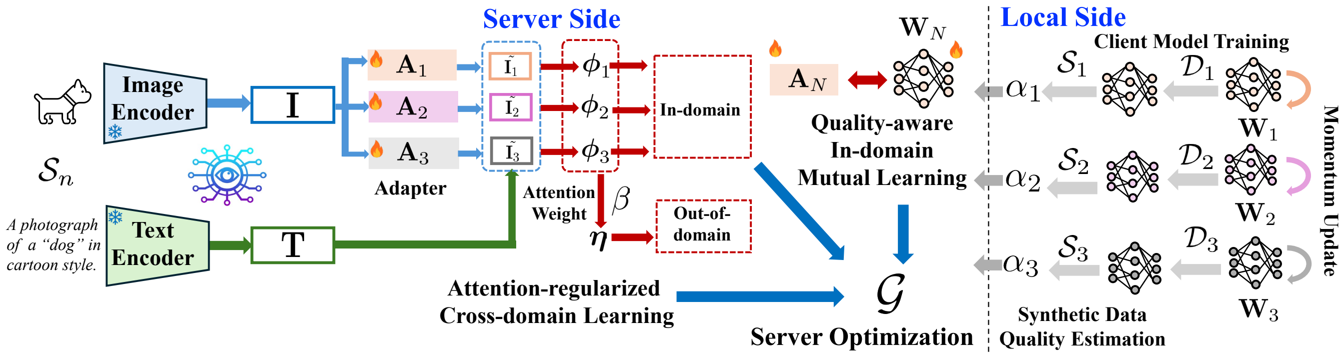

Figure 2: Overview of the proposed `FedAG` framework.

into the parameter-efficient fine-tuning (PEFT) in FMFL. The aforementioned studies typically require clients to possess FMs, with the aim of mutual benefit. In contrast, our approach places the FM on the server side, representing a more practical setting. Moreover, our objective is to enable clients to collaboratively contribute to the FM model learning with their specific domain knowledge without accessing local data.

3 Methodology

3.1 Model Input

The proposed model `FedAG` aims to iteratively inject domain knowledge into the vision foundation model CLIP deployed on the server through collaboration with N *mutually exclusive and independent domain-specific* clients $\{C_1, \cdots, C_N\}$ without sharing their private data $\{\mathcal{D}_1, \cdots, \mathcal{D}_N\}$.

To facilitate knowledge transfer while safeguarding clients' data privacy, the conventional approach involves data-free knowledge transfer, where often random Gaussian noise is utilized to distill knowledge from one model to another [Chen *et al.*, 2019]. Despite recent advancements [Raikwar and Mishra, 2022], noise-based knowledge transfer still encounters performance degradation compared to using real data. To conduct effective knowledge transfer, we leverage the open-source text-to-image model, Stable Diffusion 2.0 [Rombach *et al.*, 2022], to generate domain-specific data $\mathcal{S}_n$ for each client C_n. The details of synthetic data generation can be found in §Sec. 4.1.

In practice, clients will share the style information (text prompt or the generated textual inversion token) so that domain-specific synthetic data $\{\mathcal{S}_1, \cdots, \mathcal{S}_N\}$ can be generated on the server. Once synthetic data is generated, they will be transferred to the corresponding clients to perform the quality estimation. The communication of the synthetic data is only a one-time cost and is often negligible.

3.2 Model Overview

The proposed `FedAG` model comprises two main updates: the client update and the server update. The **client update** module (§Sec. 3.3) is designed to train a local model f_n for each client C_n using their respective data $\mathcal{D}_n$, where the parameters of f_n (i.e., $\mathbf{W}_n^t$ at the t-th communication round) encapsulate the domain-specific knowledge. Additionally, it estimates a data-quality score $\alpha_n^{i,t} \in \boldsymbol{\alpha}_n^t$ for each synthetic data instance $\mathbf{s}_n^i \in \mathcal{S}_n$. The client model parameters $\mathbf{W}_n^t$ and the estimated quality scores $\boldsymbol{\alpha}_n^t$ are then uploaded to the central server for further processing.

During the **server update** (§Sec. 3.4) at the t-th communication round, `FedAG` first learn the logits of synthetic data using the CLIP framework. It then integrates the domain knowledge from $\mathbf{W}_n^t$ into the corresponding domain-specific attention-based adapter $\mathbf{A}_n^t$ based on the learned logits through a quality-aware *in-domain* mutual learning module. Furthermore, it extends the model's capability to out-of-domain knowledge using an attention-regularized *cross-domain* learning module. Afterward, the updated client models (denoted as $\{\widehat{\mathbf{W}}_1^t, \cdots, \widehat{\mathbf{W}}_N^t\}$) are redistributed to their respective clients for another round of the client update. The updates continue iteratively until `FedAG` achieves convergence.

3.3 Client Update

Client Model Training At the t-th communication round, client C_n will receive an updated model $\widehat{\mathbf{W}}_n^{t-1}$ from the server, which is trained using the synthetic data $\mathcal{S}_n$ in the server update. Since the generated synthetic data $\mathcal{S}_n$ are different from the real domain data $\mathcal{D}_n$, directly using $\widehat{\mathbf{W}}_n^{t-1}$ as the initialized client model at the t-the communication round (i.e., $\mathbf{W}_n^t = \widehat{\mathbf{W}}_n^{t-1}$) will be unsuitable.

Figure 3 displays the empirical experiment results of models trained with real and synthetic data on the DomainNet dataset in a centralized manner, where the model is TinyViT [Wu *et al.*, 2022]. It is evident from Figure 3 that models trained with real data outperform those trained with synthetic data by a significant margin. Therefore, replacing the well-trained client model $\mathbf{W}_n^{t-1}$ with the distributed $\widehat{\mathbf{W}}_n^{t-1}$ arbitrarily would disrupt the clients' training. To mitigate this issue, we propose the use of momentum update for the client model as follows:

$$\mathbf{W}_n^t = \gamma \mathbf{W}_n^{t-1} + (1-\gamma)\widehat{\mathbf{W}}_n^{t-1}, \tag{1}$$

where γ is the hyperparameter. We then use the traditional cross-entropy (CE) loss to train the client model's parameters $\mathbf{W}_n^t$ for the n-th client using $\mathcal{D}_n$ as follows:

$$\min_{\mathbf{W}_n^t} \mathcal{L}_n^t := \frac{1}{|\mathcal{D}_n|} \sum_{(\mathbf{x}_n^i, \mathbf{y}_n^i) \in \mathcal{D}_n} \mathrm{CE}(f_n(\mathbf{x}_n^i; \mathbf{W}_n^t), \mathbf{y}_n^i), \tag{2}$$

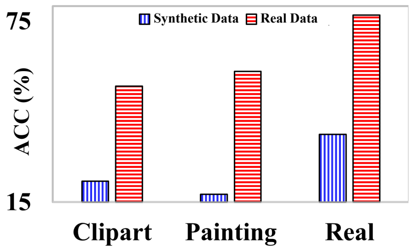

Figure 3: Performance comparison with synthetic and real data.

where f_n is a TinyViT model [Wu *et al.*, 2022], $|\mathcal{D}_n|$ is the total number of private training data, $\mathbf{x}_n^i$ is the i-th data feature, $\mathbf{y}_n^i \in \{0,1\}^{|\mathcal{Y}|}$ is the corresponding label, and $\mathcal{Y}$ is the set of distinct labels, which is shared by all domains. The trained model $\mathbf{W}_n^t$ via Eq. (2) contains the knowledge of the m-th domain.

Quality Estimation for Domain-specific Synthetic Data The synthetic dataset $\mathcal{S}_n$, generated through stable diffusion, is essential for the server update but presents an *unknown* quality challenge. To address this, we propose estimating data quality using a prototype-based similarity measurement for each domain-specific set of generated data $\mathcal{S}_n$, utilizing the trained local model $\mathbf{W}_n^t$.

Label-aware Prototype Representation Learning. Let $\mathcal{D}_n^y$ denote the subset of training data with labels $y \in \mathcal{Y}_n$. For each data instance $\mathbf{x}_n^i$ within $\mathcal{D}_n^y$, we first derive its feature representation $\mathbf{r}_n^{i,t}$ using the layers of $\mathbf{W}_n^t$ before the prediction layer. We then compute a prototype representation $\mathbf{p}_n^{y,t}$ for each label category y by averaging the representations of all data in $\mathcal{D}_n^y$, specifically, $\mathbf{p}_n^{y,t} = \frac{1}{|\mathcal{D}_n^y|}\sum_{\mathbf{x}_n^i \in \mathcal{D}_n^y} \mathbf{r}_n^{i,t}$.

Similarity-based Data Quality Estimation For the generated data subset $\mathcal{S}_n^y$ labeled y in the n-th domain, each instance $\mathbf{s}_n^i \in \mathcal{S}_n^y$ also receives a feature representation $\mathbf{q}_n^{i,t}$ through $\mathbf{W}_n^t$. We then calculate the cosine similarity $\alpha_n^{i,t}$ between $\mathbf{q}_n^{i,t}$ and the corresponding prototype $\mathbf{p}_n^{y,t}$, represented as $\alpha_n^{i,t} = \cos(\mathbf{q}_n^{i,t}, \mathbf{p}_n^{y,t})$. The vector of these similarity scores, $\boldsymbol{\alpha}_n^t$, for all generated data in $\mathcal{S}_n$ on the n-th client, is compiled and prepared for upload to the server alongside $\mathbf{W}_n^t$.

This methodology offers significant advantages: it ensures that uploading synthetic data quality scores does not compromise the confidentiality of client data, and it allows each client model to provide specific data-quality scores, thus enhancing the precision of the mutual learning process.

3.4 Server Update

Upon receiving the uploaded client models $\{\mathbf{W}_1^t, \cdots, \mathbf{W}_N^t\}$ and their corresponding estimated data-quality scores $\{\boldsymbol{\alpha}_1^t, \cdots, \boldsymbol{\alpha}_N^t\}$, the server integrates domain-specific knowledge into the basic foundation model. This is achieved by incorporating domain-specific attention-based adapters $\{\mathbf{A}_1^t, \cdots, \mathbf{A}_N^t\}$, each consisting of an identical multi-layer block positioned after the feature extractor of the vision foundation model CLIP.

CLIP-based Logit Learning The goal of `FedAG` is to inject domain knowledge included in client model parameters into the CLIP model in a parameter-efficient fine-tuning way. Let $\text{Enc}_{img}()$ represent the forzon image encoder and $\text{Enc}_{txt}()$ be the forzon text encoder of CLIP. Let $\mathbf{L}_y$ denote the description of class label y, i.e., "a photo of a $[class]$". To learn the logit for an image $\mathbf{s}_n^i \in \mathcal{S}_n$, we follow the CLIP pre-training framework and take the image $\mathbf{s}_n^i$ and all the label descriptions $\{\mathbf{L}_y\}_{y=1}^{|\mathcal{Y}|}$ as the input. In particular, we first obtain the representations of $\mathbf{s}_n^i$ and $\mathbf{L}_y$ using the corresponding encoders as follows:

$$\mathbf{I}_n^i = \text{Enc}_{img}(\mathbf{s}_n^i), \mathbf{T}_y = \text{Enc}_{txt}(\mathbf{L}_y). \tag{3}$$

Following FedCLIP [Lu *et al.*, 2023], the image representation $\mathbf{I}_n^i \in \mathbb{R}^d$ will pass an attention-based adapter $\mathbf{A}_n$ to obtain a fine-tuned domain-specific representation as follows:

$$\tilde{\mathbf{I}}_n^{i,t} = \mathbf{A}_n^t(\mathbf{I}_n^i) \odot \mathbf{I}_n^i = \text{Softmax}(\text{MLP}_n^{1,t}(\text{Tanh}(\text{MLP}_n^{2,t}(\mathbf{I}_n^i)))) \odot \mathbf{I}_n^i. \tag{4}$$

where $\tilde{\mathbf{I}}_n^{i,t} \in \mathbb{R}^d$, d is the dimension size, and $\odot$ denotes the element-wise dot product. MLP is the multi-layer perception.

Finally, we can obtain the domain-specific logit for the input image as follows:

$$\boldsymbol{\phi}_n^{i,t} = [\tilde{\mathbf{I}}_n^{i,t} \cdot \mathbf{T}_1^\top, \cdots, \tilde{\mathbf{I}}_n^{i,t} \cdot \mathbf{T}_{|\mathcal{Y}|}^\top]. \tag{5}$$

Quality-aware In-domain Mutual Learning To transfer domain-specific knowledge from the client model $\mathbf{W}_n^t$ to the CLIP model (i.e., the corresponding adapter $\mathbf{A}_n^t$), an intuitive way is to conduct knowledge distillation [Hinton *et al.*, 2015] by treating $\mathbf{W}_n^t$ as the teacher network and the adapter-based CLIP as the student network. However, this simple strategy presents several limitations: it overlooks the quality of domain-specific synthetic data $\mathcal{S}_n$ involved in the distillation process and only allows unidirectional knowledge transfer, which does not update the local model $\mathbf{W}_n^t$, thus underutilizing the potential of the federated learning framework.

To overcome these shortcomings, we introduce a quality-aware in-domain mutual learning strategy. This approach not only ensures effective integration of domain-specific knowledge into $\mathbf{A}_n^t$ but also facilitates dynamic updates of the local model, leveraging the quality assessments of the synthetic data to enhance the overall learning process. Note that we use $\widehat{\mathbf{W}}_n^t$ to distinguish the updates of the client model $\mathbf{W}_n^t$ on the server. The loss function is defined as follows:

$$\min_{\mathbf{A}_n^t, \widehat{\mathbf{W}}_n^t} \mathcal{J}_n^t := \frac{1}{2\sum_{j=1}^{|\mathcal{S}_n|} \alpha_n^{j,t}} \sum_{\mathbf{s}_n^i \in \mathcal{S}_n} \alpha_n^{k,t} \Big\{ \text{KL}(\boldsymbol{\theta}_n^{i,t} || \boldsymbol{\varphi}_n^{i,t}) + \text{KL}(\boldsymbol{\varphi}_n^{i,t} || \boldsymbol{\theta}_n^{i,t}) \Big\}, \tag{6}$$

where

$$\boldsymbol{\theta}_n^{i,t} = f_n(\mathbf{s}_n^i; \widehat{\mathbf{W}}_n^t), \boldsymbol{\varphi}_n^{i,t} = \text{softmax}(\boldsymbol{\phi}_n^{i,t})), \tag{7}$$

$\boldsymbol{\theta}_n^{i,t}$ is the predicted probabilities by the client model $\widehat{\mathcal{W}}_n^t$ on each data instance $\mathbf{s}_n^i$ on the server, and $\boldsymbol{\varphi}_n^{i,t}$ is probabilities ouputed by the CLIP model using Eq. (5). $\text{KL}(\cdot||\cdot)$ is the Kullback–Leibler divergence.

Attention-regularized Cross-domain Learning Using Eq. (6), we can update the adapters and client models simultaneously. However, such a design may only work for data belonging to existing domains, i.e., there is a lack of generalization ability for out-of-domain data. We propose a novel attention-regularized cross-domain learning strategy to equip the proposed FedAG with the capability for dealing with out-of-domain data.

In particular, for a synthetic data instance $\mathbf{s}_n^i \in \mathcal{S}_n$, we not only generate its logit $\boldsymbol{\phi}_n^{i,t}$ via Eq. (5) with the domain-specifc adaptor $\mathbf{A}_n^t$ but also from other adaptors $\{\mathbf{A}_1^t, \cdots, \mathbf{A}_{n-1}^t, \mathbf{A}_{n+1}^t, \cdots, \mathbf{A}_N^t\}$. We calculate the attention score $\beta_k^{i,t} \in \mathbb{R}$ ($k \in [1, N]$) for each adaptor using a softmax function on top of an MLP layer and then obtained the aggregated logit for each data as follows:

$$\boldsymbol{\eta}_n^{i,t} = \sum_{k=1}^{N} \beta_k^{i,t} \boldsymbol{\phi}_k^{i,t},$$
$$[\beta_1^{i,t}, \cdots, \beta_N^{i,t}] = \text{softmax}([\text{MLP}(\boldsymbol{\phi}_1^{i,t}), \cdots, \text{MLP}(\boldsymbol{\phi}_N^{i,t})]). \tag{8}$$

The domain index n is known for each training data during the training. Thus, the attention weight $\beta_n^{i,t}$ should be larger than those obtained from the other adapters. We use this intuition as prior knowledge to guide the model learning via an attention-based regularize as follows:

$$\mathcal{R}_n^{i,t} = \max(0, \delta + \max([\beta_1^{i,t}, \cdots, \beta_{n-1}^{i,t}, \beta_{n+1}^{i,t}, \cdots, \beta_N^{i,t}]) - \beta_n^{i,t})), \tag{9}$$

where δ is the margin hyperparameter.

Server Optimization Based on Eqs. (6), (7), (8), and (9), we obtain the final loss function for the server update as follows:

$$\min_{\mathcal{A}^t, \mathcal{W}^t} \mathcal{G}^t := \frac{1}{N} \sum_{n=1}^{N} \Bigg[\mathcal{J}_n^t + \sum_{(\mathbf{s}_n^i, \mathbf{y}_n^i) \in \mathcal{S}_n} \Big[\underbrace{\text{CE}(\boldsymbol{\varphi}_n^{i,t}, \mathbf{y}_n^i)}_{\text{In-domain Prediction}} + \underbrace{\text{CE}(\boldsymbol{\kappa}_n^{i,t}, \mathbf{y}_n^i)}_{\text{Cross-domain Prediction}} + \lambda \mathcal{R}_n^{i,t} \Big] \Bigg], \tag{10}$$

where $\mathcal{A}^t = \{\mathbf{A}_1^t, \cdots, \mathbf{A}_N^t\}$, $\mathcal{W}^t = \{\widehat{\mathbf{W}}_1^t, \cdots, \widehat{\mathbf{W}}_N^t\}$, $\boldsymbol{\kappa}_n^{i,t} = \text{softmax}(\boldsymbol{\eta}_n^{i,t})$, and λ is the hyperparameter. The updated client models $\mathcal{W}^t = \{\widehat{\mathbf{W}}_1^t, \cdots, \widehat{\mathbf{W}}_N^t\}$ will be redistributed to the corresponding domain-specific clients for the next communication round update.

3.5 Inference

FedAG will be trained iteratively using Eqs. (2) and (10) until converge. We then conduct the inference on the testing data. For the **in-domain** scenario, where the domain index n is *known*, we use the label index with the maximum value in $\boldsymbol{\phi}_n^i$ as the predicted label, i.e., $\hat{y}_n^i = \arg\max_{\{1,\cdots,|\mathcal{Y}|\}}(\boldsymbol{\phi}_n^i)$ via Eq. (5). For the **out-of-domain** testing where the domain is *unknown*, we use the label index with the maximum value in $\boldsymbol{\eta}^i$ as the predicted label, i.e., $\hat{y}^i = \arg\max_{\{1,\cdots,|\mathcal{Y}|\}}(\boldsymbol{\eta}^i)$ via Eq. (8).

4 Experimental Setups

4.1 Datasets

Real Data To fairly validate the proposed model FedAG in our experiments, we focus on the image classification task on three commonly domain-shifting datasets. (1) **DomainNet**[1]. It totally has 569,010 images from 6 domains, including clipart, infographics, painting, quickdraw, real, and sketch. Each domain contains 48K to 172K images, categorized into 345 classes. (2) **Office-Home dataset**[2]. It has 15,500 images from 4 different dimensions: artistic images, clip art, product images, and real-world images. Each domain has 65 object classes. (3) **ImageCLEF-DA**[3]. It is a benchmark for the ImageCLEF 2014 domain adaption challenge, including Caltech-256, ImageNet ILSVRC 2012, and Pascal VOC 2012. There are 12 categories and 50 images in each domain.

Since we are addressing both "in-domain" and "out-of-domain" scenarios, we partition the domains in each dataset into training and testing domains. The data in the testing domains are exclusively used for evaluating **out-of-domain** performance. For the training domains, we distribute each domain's data to each client. Specifically, we randomly select 90% of the data for client model training, reserving the remaining 10% for **in-domain** validation.

Synthetic Data When training the proposed FedAG, we also incorporate domain-level synthetic data generated by Stable Diffusion V2[4]. The number of synthetic data for each training domain equals 10% of the real domain data. For the style-distinctive datasets, **DomainNet** and **OfficeHome**, synthetic data can be readily generated using text prompts following the template "a photograph/drawing of $class in $style style". However, for **ImageCLEF-DA**, where the style information is implicit and challenging to articulate using text prompts, we resort to generating synthetic data using textual inversion [Gal *et al.*, 2022]. Textual inversion entails deriving an appropriate text token corresponding to the implicit style. We sampled 10 instances from each of the 12 classes within the real ImageCLEF dataset and employed the Diffuser library to perform textual inversion. Once the style token is derived, the server utilizes a similar template, "a $class in $style_token style", to generate synthetic images for **ImageCLEF-DA**.

4.2 Baselines

We compare the proposed FedAG with several baselines in different settings, including zero-shot inference, centralized training, and federated learning.

Zero-Shot Inference We directly use the original CLIP model to predict the labels for given images in the testing data. This zero-shot inference baseline is denoted as CLIP_{zero}.

Centralized Learning Since FedAG uses private domain data $\{\mathcal{D}_1, \cdots, \mathcal{D}_N\}$ for client training and synthetic data $\{\mathcal{S}_1, \cdots, \mathcal{S}_N\}$ for server training, for a fair comparison, we also use them together for the centralized training baselines. This setting involves two kinds of centralized training: classical centralized training and fine-tuning on CLIP.

[1] https://ai.bu.edu/M3SDA/

[2] https://www.hemanthdv.org/officeHomeDataset.html

[3] https://www.imageclef.org/2014

[4] https://huggingface.co/stabilityai/stable-diffusion-2

Table 1: In-domain evaluation results. "Centra." means the centralized learning, "FLFM" means federated learning with foundation models.

Setting		Method	ImageCLEF-DA		Office-Home			DomainNet		
			Caltech	ImageNet	Art	Product	Real	Clipart	Painting	Real
Zero-shot		CLIP_{zero}	97.25	96.87	78.12	85.14	86.33	62.67	59.77	78.07
Centra.	Classical	TinyViT_{cen}	85.41	82.06	62.81	83.97	76.32	53.44	58.32	77.01
	PEFT	CLIP_{LoRA}	98.49	95.45	85.01	87.92	88.44	68.15	65.66	83.28
		$\text{CLIP}_{adapter}$	98.11	95.52	84.17	88.02	88.26	68.50	65.13	82.79
Federated	Classical	FedAvg	95.11	83.33	75.62	86.85	82.07	51.66	53.02	69.34
		FedAvg_{ft}	90.06	80.25	61.33	75.51	74.68	48.27	43.87	62.06
		FedProx	95.75	84.16	76.98	87.26	83.15	50.40	53.45	69.87
		FedProx_{ft}	91.30	80.54	62.47	75.65	74.98	48.89	44.92	63.77
	FLFM	FedClip	97.34	97.89	82.14	84.33	87.62	67.96	65.78	82.93
		FedOT	97.26	97.91	82.56	85.47	86.61	67.68	65.85	83.20
		`FedAG`	**98.62**	**98.56**	**84.97**	**88.69**	**88.79**	**70.36**	**66.29**	**84.92**

For the classical training, we directly train TinyViT with all data, denoted as TinyViT_{cen}. We also choose two commonly used parameter-efficient fine-tuning methods, adapter fine-tuning and LoRA [Hu *et al.*, 2021] as baselines, which are denoted as $\text{CLIP}_{adapter}$ and CLIP_{LoRA}. $\text{CLIP}_{adapter}$ will learn a shared adapter, but the number of parameters in the adaptor is the same as that of `FedAG`, although `FedAG` is equipped with several domain-specific adapters. We set the rank for CLIP_{LoRA} as 32.

Federated Learning We use two classical federated learning approaches, FedAvg [McMahan *et al.*, 2017] and FedProx [Li *et al.*, 2020], as baselines. These approaches are trained only with client data without interacting with CLIP. Since our model `FedAG` uses synthetic data for fine-tuning the client models, in the experiments, we also fine-tuned FedAvg and FedProx on the server. The fine-tuned models are denoted as FedAvg_{ft} and FedProx_{ft}.

The most relevant baselines are FedCLIP [Lu *et al.*, 2023] and FedOT [Xiao *et al.*, 2023]. FedCLIP deploys a CLIP model for each client and fine-tunes the adapter on the local side. The adapters are uploaded to the server for aggregation, similar to FedAvg. FedOT [Xiao *et al.*, 2023] is a federated version of Offsite-Tuning, where the CLIP model generates a compressed model and an emulator, which are shared with clients for their training.

4.3 Implementation Details

For each dataset, we assign each in-domain data to one client. We utilize ViT_Tiny_patch16_224[5] for the client model and ViT_B_32[6] for the image encoder for the server side. Our experimental setup involves 10 communication rounds. For the local update, we set the local trainin epoch as 10, the local learning rate as 0.0001, the batch size is 32, and the optimizer used in the optimization is Adam. For the server update, we set $\lambda = 0.1$, $\gamma = 0.1$, and $\delta = 0.001$, the epoch of quality-aware in-domain mutual learning as 3, and the epoch of adapter initilization as 5. All experiments are conducted on an NVIDIA A6000 with CUDA version 12.0, running on a Ubuntu 20.04.6 LTS server. All baselines and the proposed `FedAG` are implemented using PyTorch 2.0.1.

[5]https://huggingface.co/WinKawaks/vit-tiny-patch16-224
[6]https://huggingface.co/openai/clip-vit-base-patch32

5 Results

5.1 In-domain Evaluation

Table 1 presents the results of the in-domain evaluation, where we train the models using the domains shown in the table and conduct the testing with the head-out domain data. We can observe that the proposed `FedAG` performs best on all domains in all datasets. CLIP_{zero} is a zero-shot learning model with CLIP, which does not use any training data. We can observe that it performs better than the classical centralized learning approach TinyViT_{cen} and federated learning models FedAvg, FedAvg_{ft}, FedProx, and FedProx_{ft}. These comparisons prove the predictive power of foundation models for downstream tasks.

The centralized PEFT approaches CLIP_{LoRA} and $\text{CLIP}_{adapter}$ achieve comparable performance but outperform the zero-shot model CLIP_{zero}, which confirms the necessity of fine-tuning foundation models for boosting performance. Although they are trained in a centralized manner and perform the best among all baselines, their performance is worse than that of `FedAG`. The reason is that these two approaches only use one adapter or two low-rank matrices to store mixed domain knowledge. However, our model uses domain-specific adapters to capture the characteristics of domains, thus leading to the best performance in the in-domain evaluation. These results also validate the design of multiple domain adapters.

For the classical federated learning approaches, we can observe that using synthetic data to fine-tune the aggregated model on the server hurts the model training. These results also confirm the necessity of employing the momentum update in `FedAG` (i.e., Eq. (1)) for the client model before training again. When comparing with the federated fine-tuning approaches, we can find they also perform better than CLIP_{zero} but have performance gaps with centralized PEFT approaches CLIP_{LoRA} and $\text{CLIP}_{adapter}$. These results demonstrate the efficacy of injecting domain knowledge into foundation models in a federated way.

5.2 Out-of-domain Evaluation

In the previous section, our main focus was on in-domain evaluation. However, the ultimate goal of training a foundation model is to make it applicable to various downstream

Table 2: Out-of-domain results. “Centra.” means the centralized learning, “FLFM” means federated learning with foundation models.

Setting		Method	ImageCLEF-DA	Office-Home	DomainNet		
			Pascal	**Clipart**	**Infograph**	**Quickdraw**	**Sketch**
Zero-shot		CLIP$_{zero}$	82.13	61.07	39.34	13.06	58.11
Centra.	Classical	TinyViT$_{cen}$	71.66	42.66	20.15	10.67	40.75
	PEFT	CLIP$_{LoRA}$	81.22	67.15	42.10	14.38	59.48
		CLIP$_{adapter}$	81.08	67.31	42.22	13.85	59.01
Federated	Classical	FedAvg	78.33	43.58	26.75	10.78	40.56
		FedAvg$_{syn}$	73.02	41.12	24.27	10.33	37.91
		FedProx	78.69	45.88	27.50	12.04	40.97
		FedProx$_{syn}$	72.68	40.75	24.63	11.89	38.54
	FLFM	FedClip	82.45	64.44	41.65	12.89	59.23
		FedOT	82.10	65.27	40.70	15.51	60.30
		`FedAG`	**83.78**	**68.15**	**45.56**	**21.04**	**63.29**

Table 3: Ablation study results on the DomainNet dataset.

Method	In-domain			Cross-domain		
	Clipart	**Painting**	**Real**	**Infograph**	**Quickdraw**	**Sketch**
`FedAG`$_{mome}$	68.54	65.60	83.00	44.38	20.14	62.85
`FedAG`$_{quality}$	68.12	65.13	83.11	44.79	20.58	63.15
`FedAG`$_{cross}$	70.04	66.11	84.13	40.63	15.70	59.04
`FedAG`$_{reg}$	68.26	64.05	81.08	42.01	17.55	61.69
`FedAG`	**70.36**	**66.29**	**84.92**	**45.56**	**21.04**	**63.29**

tasks, including inference on unseen data. To assess this capability, we conduct an out-of-domain evaluation using the trained models used in Table 1 to validate the unseen domains, the results of which are presented in Table 2.

For the out-of-domain evaluation, we observe similar trends as in the in-domain evaluation, as shown in Table 1. Specifically, `FedAG` outperforms all baselines, and CLIP$_{zero}$ performs better than classical models. However, compared to the in-domain evaluation results, the performance gaps between the centralized PEFT models (i.e., CLIP$_{LoRA}$ and CLIP$_{adapter}$) and CLIP$zero$ are not as significant. In fact, their performance is even worse than that of FedOT in several domains. These results highlight the limitations of existing models in generalizing out-of-domain knowledge.

In contrast to existing approaches, our proposed `FedAG` consistently achieves superior performance, leading to significant improvements in accuracy. For instance, in the Quickdraw domain of the DomainNet dataset, our approach demonstrates a 36% performance increase compared to the best baseline FedOT. These results strongly indicate that our model effectively handles out-of-domain knowledge.

5.3 Abaltion Study

We use the following baselines to validate the effectiveness of our model design. `FedAG`$_{mome}$ does not use momentum update (i.e., Eq. (1)) for the local model after receiving the learned global model. `FedAG`$_{quality}$ denotes removing data quality estimation in Eq. (6). `FedAG`$_{cross}$ denotes removing the module of attention-regularized cross-domain learning. `FedAG`$_{reg}$ means that we remove the designed attention-based regularization term $\mathcal{R}$ in Eq. (10).

The results of the ablation studies on the DomainNet dataset are presented in Table 3. It is evident that removing each designed module results in a performance drop, underscoring the necessity of each module. Interestingly, the in-domain results suggest that cross-domain learning may not be as crucial compared to momentum updates and data quality estimation. However, in the out-of-domain evaluation, `FedAG`$_{cross}$ plays a significant role, as its removal leads to a dramatic performance drop. These findings align with the motivations behind our model design, emphasizing the importance of the cross-domain learning module in addressing the out-of-domain issue.

6 Conclusion

In this study, we introduced Federated Adapter Generalization (`FedAG`), an innovative federated fine-tuning approach designed to address the challenges of domain-specific characteristics and out-of-domain generalization in vision foundation models. Using multiple fine-grained adapters and novel learning modules, `FedAG` effectively integrates domain-specific knowledge and enhances generalization across diverse domains. Our extensive experiments on various datasets validate the efficacy of `FedAG`, showing performance improvements over traditional fine-tuning methods. This work underscores the importance of developing federated learning strategies that respect data privacy while maintaining high model performance across different domains, paving the way for more robust and adaptable vision foundation models.

References

[Bommasani *et al.*, 2021] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[Chen *et al.*, 2019] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3514–3522, 2019.

[Chen *et al.*, 2024] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11285–11293, 2024.

[Chua *et al.*, 2023] Terence Jie Chua, Wenhan Yu, Jun Zhao, and Kwok-Yan Lam. Fedpeat: Convergence of federated learning, parameter-efficient fine tuning, and emulator assisted tuning for artificial intelligence foundation models with mobile edge computing. *arXiv preprint arXiv:2310.17491*, 2023.

[Deng *et al.*, 2023] Yongheng Deng, Ziqing Qiao, Ju Ren, Yang Liu, and Yaoxue Zhang. Mutual enhancement of large and small language models with cross-silo knowledge transfer. *arXiv preprint arXiv:2312.05842*, 2023.

[Fan *et al.*, 2023] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fatellm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*, 2023.

[Gal *et al.*, 2022] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[Guo *et al.*, 2023] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 2023.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Hu *et al.*, 2021] 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.

[Khalid *et al.*, 2023] Umar Khalid, Hasan Iqbal, Saeed Vahidian, Jing Hua, and Chen Chen. Cefhri: A communication efficient federated learning framework for recognizing industrial human-robot interaction. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10141–10148. IEEE, 2023.

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

[Li *et al.*, 2023] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[Lu *et al.*, 2023] Wang Lu, Xixu Hu, Jindong Wang, and Xing Xie. Fedclip: Fast generalization and personalization for clip in federated learning. *arXiv preprint arXiv:2302.13485*, 2023.

[Marchisio *et al.*, 2022] Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. *arXiv preprint arXiv:2212.10503*, 2022.

[Marchisio *et al.*, 2023] Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5474–5490, 2023.

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

[Peng *et al.*, 2024] Zhaopeng Peng, Xiaoliang Fan, Yufan Chen, Zheng Wang, Shirui Pan, Chenglu Wen, Ruisheng Zhang, and Cheng Wang. Fedpft: Federated proxy fine-tuning of foundation models. *arXiv preprint arXiv:2404.11536*, 2024.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Raikwar and Mishra, 2022] Piyush Raikwar and Deepak Mishra. Discovering and overcoming limitations of noise-engineered data-free knowledge distillation. *Advances in Neural Information Processing Systems*, 35:4902–4912, 2022.

[Ren *et al.*, 2024] Chao Ren, Han Yu, Hongyi Peng, Xiaoli Tang, Anran Li, Yulan Gao, Alysa Ziying Tan, Bo Zhao, Xiaoxiao Li, Zengxiang Li, et al. Advances and open challenges in federated learning with foundation models. *arXiv preprint arXiv:2404.15381*, 2024.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 10684–10695, June 2022.

[Sajjad *et al.*, 2023] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.

[Su *et al.*, 2024] Shangchao Su, Mingzhao Yang, Bin Li, and Xiangyang Xue. Federated adaptive prompt tuning for multi-domain collaborative learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15117–15125, 2024.

[Woisetschläger *et al.*, 2024] Herbert Woisetschläger, Alexander Isenko, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. A survey on efficient federated learning methods for foundation model training. *arXiv preprint arXiv:2401.04472*, 2024.

[Wu *et al.*, 2022] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European Conference on Computer Vision*, pages 68–85. Springer, 2022.

[Xiao *et al.*, 2023] Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

[Zhang *et al.*, 2023] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pages 9963–9977. Association for Computational Linguistics (ACL), 2023.

[Zhuang *et al.*, 2023] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*, 2023.