FedCLIP: Fast Generalization and Personalization for CLIP in Federated Learning

Wang Lu[1] Xixu Hu[2] Jindong Wang[3*] Xing Xie[3]
[1] National Engineering Research Center, Beijing, China
[2] City University of Hong Kong, Hong Kong
[3] Microsoft Research Asia, Beijing, China
newlw230630@gmail.com, xixuhu2-c@my.cityu.edu.hk, {jindong.wang, xingx}@microsoft.com

Abstract

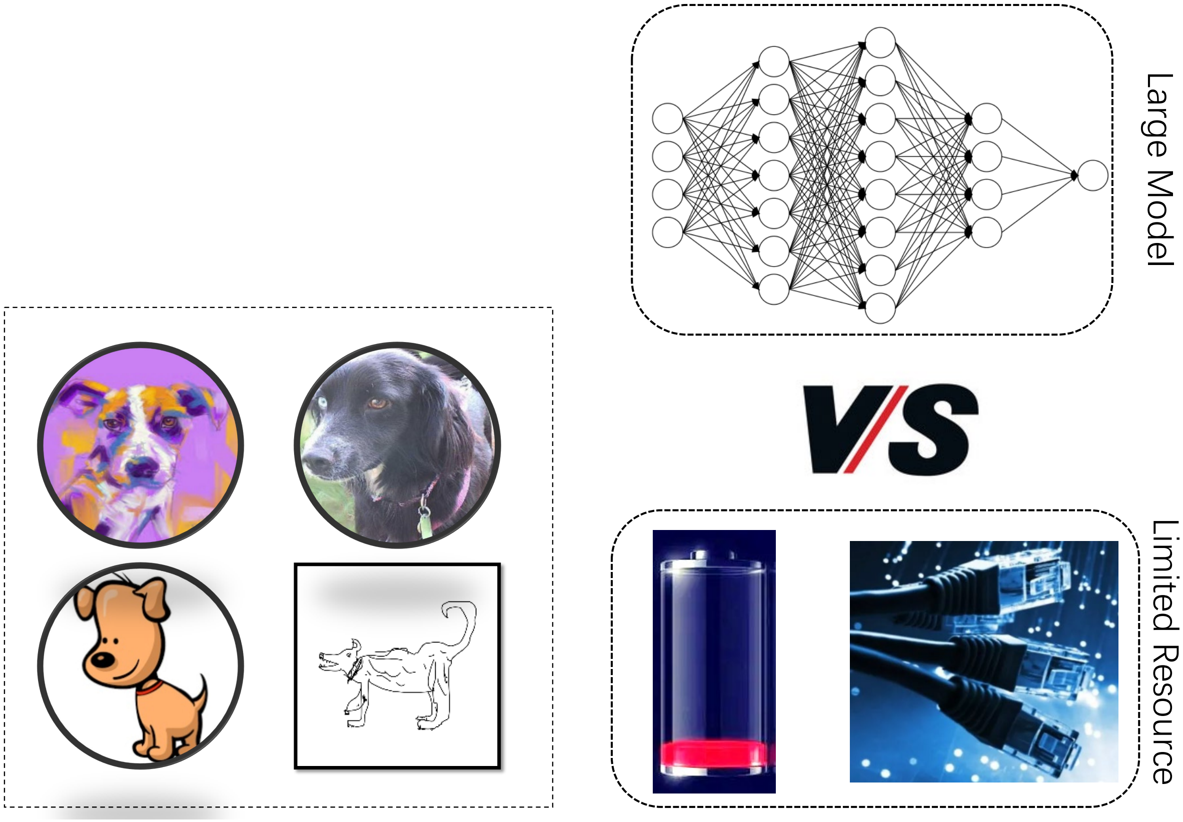
*Federated learning (FL) has emerged as a new paradigm for privacy-preserving computation in recent years. Unfortunately, FL faces two critical challenges that hinder its actual performance: data distribution heterogeneity and high resource costs brought by large foundation models. Specifically, the non-IID data in different clients make existing FL algorithms hard to converge while the high resource costs, including computational and communication costs that increase the deployment difficulty in real-world scenarios. In this paper, we propose an effective yet simple method, named FedCLIP, to achieve fast generalization and personalization for CLIP in federated learning. Concretely, we design an attention-based adapter for the large model, CLIP, and the rest operations merely depend on adapters. Lightweight adapters can make the most use of pretrained model information and ensure models be adaptive for clients in specific tasks. Simultaneously, small-scale operations can mitigate the computational burden and communication burden caused by large models. Extensive experiments are conducted on three datasets with distribution shifts. Qualitative and quantitative results demonstrate that FedCLIP significantly outperforms other baselines (**9**% overall improvements on PACS) and effectively reduces computational and communication costs (**283x** faster than FedAVG). Our code will be available at: `https://github.com/microsoft/PersonalizedFL`.*

1 Introduction

The success of machine learning, especially deep learning, is inseparable from a large amount of data. However, data, as an important resource, usually scatter across different individuals or organizations. In recent years, people pay more attention to data privacy and security and some organizations even enact relevant regulations and laws, e.g. The EU general data protection regulation (GDPR) [49] and China's cyber power [20]. Under this circumstance, direct raw data communication can be impossible in reality, making traditional data-centric machine learning paradigms unlikely to work. To cope with this challenge, federated learning (FL) [53] emerges as a new distributed machine learning paradigm and has been widely adopted in various applications.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[*] Corresponding author: Jindong Wang: jindong.wang@microsoft.com.



(a) Data distribution shifts

(b) Huge resource demands

Figure 1: Existing issues in federated learning. In Figure 1(a), circles denote participated clients while squares denote unseen targets.

Federated learning makes it possible to perform model aggregation without directly accessing the raw user data from different clients. One of the earliest works in FL is called FedAVG [33] which aggregates distributed information using a simple and powerful averaging algorithm. FedAVG mainly contains four steps, including training local models with local data, uploading local models to the server, aggregating models in the server, and distributing the aggregated model to each individual or organization. These four steps are executed for multiple rounds for better information aggregation. FedAVG can ensure that raw data does not leave the local client and thus protect data privacy and security. Due to its simplicity and great performance, FedAVG quickly became popular in many areas [24, 40, 3].

In this paper, we are specially interested in federated learning under the large foundation models era [4]. Foundation models, as suggested by the name, have become increasingly popular in different machine learning tasks, such as Vision Transformer in computer vision [56] and the GPT series in natural language processing [37]. Since these models are extremely large, e.g., GPT-3 [5] has 175 billion parameters, our key question is: how to perform effective and efficient federated learning using these large models?

Specifically, two critical research challenges arise in this situation: data distribution shifts and huge resource demands. On the one hand, data distribution shifts widely exist in the real world, e.g. figures shown in Figure 1(a). When meeting heterogeneous data, common federated learning methods can suffer from slow convergence and low accuracy due to inconsistent optimization directions, local optima, or some other factors [12]. A qualified FL model can cope with both various clients and unseen targets, i.e. personalization and generalization. On the other hand, huge resource demands of increasingly popular large models lead to conflicts with realistically constrained resources, as shown in Figure 1(b). In addition to high computational costs, communication cost is also a critical metric in federated learning. For instance, the CLIP [36] model based on VIT-B/32 contains more than 10^8 trainable parameters and most existing networks cannot afford to transmit it quickly. Achieving fast generalization and personalization with minimal resource costs is an urgent issue to be addressed.

Some existing work tried to address the issues mentioned above [32, 55, 14]. FedAP [32] attempted to learn the similarity among clients and then leveraged the learned similarity matrix to guide aggregation. FedAP could achieve acceptable personalization results but it ignored generalization. Another paper [55] discussed two gaps, including the out-of-sample gap and the participation gap. These two gaps correspond to goals of generalization and personalization respectively. This paper performed extensive empirical studies to analyze these issues but it did not offer a possible solution for large models. PromptFL [14] only updated the prompts instead of the whole model to accelerate the whole process. However, clients still require large amounts of computation and PromptFL is not designed for personalization and generalization.

In this paper, we propose FedCLIP to achieve fast generalization and personalization for CLIP in federated learning. Since larger pretrained models, e.g. CLIP, have contained enough prior information, our goal is to find where we should focus in specific tasks. The core part of FedCLIP is AttAI, an attention-based adapter for the image encoder in CLIP. Instead of finetuning whole networks, AttAI directly utilizes fixed features extracted by pretrained models and explores where FedCLIP should pay attention to for specific tasks. Simply training AttAI can ensure FedCLIP preserving prior information as much as possible while it allows models adapted for specific tasks. Through AttAI, FedCLIP does not rely on pretrained models anymore once obtaining diversified and robust features and thus FedCLIP can save large amounts of computational costs and communication costs. Therefore, FedCLIP is extensible and can be deployed to many applications.

Our contributions are as follows.

1. We propose FedCLIP, a fast generalization and personalization learning method for CLIP in federated learning. It can achieve personalization for participating clients and its remarkable generalization ability can attract new clients.

2. Extensive experiments on three public image benchmarks demonstrate that FedCLIP can have achieved personalization and generalization performance at the same time (**9**% overall improvements on PACS). More importantly, FedCLIP reduces the number of trainable parameters thus saving communication costs and computational costs (**283x** faster than FedAVG).

3. FedCLIP is extensible and can be applied in many real applications, which means it can work well in many circumstances. We can even embed it in some other architectures, e.g. BERT [46] and ViT [16]. Our code will be available at: `https://github.com/microsoft/PersonalizedFL`.

The remainder of this paper is organized as follows. In Sec. 2, we introduce related work. And then we elaborate on the proposed method in Sec. 4. Extensive experiments are reported and analyzed in Sec. 4. Finally, we conclude the paper and provide possible future work in Sec. 5.

2 Related Work

2.1 Challenges in Machine Learning

Machine learning has achieved great success and gradually entered people's daily lives [42, 34, 50]. It has been applied to many fields, e.g. human activity recognition [29], face recognition [28], and healthcare [8]. Successful machine learning applications, especially deep learning based applications, often require a large amount of data and lots of computational resources. In most cases, a deluge of data and computing resources can lead to easy success, such as ChatGPT [47]. However, data and computation also mean money and resources. In reality, it is impossible to aggregate all data together in some situations. There seems to be a contradiction between the massive resource requirements of traditional methods and the limited real environment.

Generalization is another challenging problem caused by data distribution shifts. Its goal is to learn a generalized model with limited data and it expects that the learned model can work well on unseen targets with unknown distributions. [51] gives a survey on domain generalization and first groups existing methods into three categories, including data manipulation [31], representation learning [30], and learning strategy [19].

2.2 Federated Learning

Data is often scatted everywhere and cannot be aggregated together due to some factors, such as laws and regulations [49] and the awakening of people's awareness of data security and privacy protection. In such an environment, federated learning came into being [53, 41]. According to [53], federated learning can be grouped into three categories, including horizontal federated learning, vertical federated learning, and federated transfer learning. Most deep learning based methods belong to horizontal federated learning and so is this paper. For a more detailed introduction, please refer to the survey [27].

FedAVG is a traditional horizontal federated learning method [33]. Although it is simple, it was applied in many applications. When meeting data distribution heterogeneity, FedAVG appeared powerless [43]. And many researchers proposed various methods to solve the above problems. FedProx [25] added a proximal regularized term to FedAVG and it allowed slight model gaps between clients and the server. In FedBN [26], the authors thought that parameters in batch normalization layers can represent data distribution, and keeping specific batch normalization layers for each client could make local models personalized. Another latest method, FedAP [32], learned the similarity between clients based on the statistics of the batch normalization layers while preserving the specificity of each client with different local batch normalization. The above methods all achieve satisfactory results in their corresponding scenarios. However, most of them focused on personalization and ignored generalization issues [7].

Generalization in federated learning is a novel problem. In recent two years, some papers tried to solve this problem. [55] first discussed the generalization in federated learning and it proposed a framework to disentangle performance gaps, including out-of-sample gaps and participation gaps. FED-DRO [7] proposed a novel federated learning framework to explicitly decouple a model's dual duties with two prediction tasks and it mainly focused on label shifts. Some other work tried to adapt existing domain generalization methods to generalization in federated learning [15, 45, 35, 6]. FL Games [15] utilized Nash equilibrium to learn causal features that were invariant across clients which is similar to Invariant Risk Minimization (IRM) [2]. FedSAM [35] proposed a general effective algorithm based on Sharpness Aware Minimization (SAM) local optimizer [11]. Although these methods can bring generalization, they were not designed for large models and could not make full use of knowledge brought by pretrained models.

2.3 CLIP and Large Models

From perceptron [13] to AlexNet [1] to ResNet [17] to Vision transformer [56] to CLIP [36], pretrained models have become larger and larger. The importance of pretrained models has been increasing and pretrained models contain a growing amount of knowledge. For specific applications, researchers usually choose suitable backbone models and then adopt some techniques, e.g. finetune [44], to slightly adapt pretrained models. Since pretrained models are trained via a large amount of data, features extracted from them are often generalized and insightful. Few works pay attention to large models in federated learning and high demands of computational costs and communication costs hinder the development of this field. In this paper, we focus on CLIP in federated learning.

CLIP [36] learned SOTA image representations from scratch on a dataset of 400 million(image, text) pairs collected from the internet. The natural language was used to reference learned visual concepts. It has been applied in many fields and demonstrated its superiority [38, 22]. However, in federated learning, CLIP is still in its infancy. PromptFL [14] replaced the federated model training with the federated prompt training to simultaneously achieve efficient global aggregation and local training by exploiting the power of foundation models in a distributed way. However, it still requires certain computational costs and it is not designed for data distribution heterogeneity problems. Moreover, it is hard to tune the hyperparameters for the prompt techniques in Transformer.In this paper, we focus on fast personalization and generalization for CLIP.

3 Method

3.1 Problem Formulation

In a generalization and personalization federated learning setting, N different clients, denote as $\{C_1, C_2, \cdots, C_N\}$, participate in exchanging information and they have data, denoted as $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$ with different distributions, which means $P(\mathcal{D}_i) \neq P(\mathcal{D}_j)$. In this paper, we only focus on homogeneous data with the same input space and output space, i.e. $\mathcal{X}_i = \mathcal{X}_j, \mathcal{Y}_i = \mathcal{Y}_j, \forall i \neq j$. Each dataset, $\mathcal{D}_i = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{n_i}$, consists of three parts, a training dataset $\mathcal{D}_i^{train} = \{(\mathbf{x}_{i,j}^{train}, y_{i,j}^{train})\}_{j=1}^{n_i^{train}}$, a validation dataset $\mathcal{D}_i^{valid} = \{(\mathbf{x}_{i,j}^{valid}, y_{i,j}^{valid})\}_{j=1}^{n_i^{valid}}$ and a test dataset $\mathcal{D}_i^{test} = \{(\mathbf{x}_{i,j}^{test}, y_{i,j}^{test})\}_{j=1}^{n_i^{test}}$. Three sub-datasets in each client have no overlap and $n_i = n_i^{train} + n_i^{valid} + n_i^{test}, \mathcal{D}_i = \mathcal{D}_i^{train} \cup \mathcal{D}_i^{valid} \cup \mathcal{D}_i^{test}$. Our goal is to aggregate all clients' information with preserving data privacy and security and learn a good model f for each client $\mathcal{D}_i$:

$$\min_f \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i^{test}} \sum_{j=1}^{n_i^{test}} \ell(f(\mathbf{x}_{i,j}^{test}), y_{i,j}^{test}), \tag{25}$$

where ℓ is a loss function. Moreover, for generalization, we assume that there exist M different clients, denote as $\{F_1, F_2, \cdots, F_M\}$, with data $\{\mathcal{D}_1^F = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{m_1}, \mathcal{D}_2^F = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{m_2}, \cdots, \mathcal{D}_N^F = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{m_M}\}$. These M clients do not participate in training, and we hope f can also be able to perform well on these clients.

$$\min_f \frac{1}{M} \sum_{i=1}^{M} \frac{1}{m_i} \sum_{j=1}^{m_i} \ell(f(\mathbf{x}_{i,j}), y_{i,j}), \tag{26}$$

3.2 Preliminaries

CLIP CLIP, Contrastive Language Image Pre-training, is an efficient and scalable method of learning [36]. To compensate for the problems caused by the amount of data and model parameters, it trained a large model with over 4×10^8 pairs of data. With help of natural language supervision, CLIP can better understand concepts of visual images and better learn the semantic connections behind images. Usually, CLIP models contain more information and they might be more robust.

A simple CLIP model regularly contains two parts, an image encoder f^I and a text encoder f^T. In common models, labels are frequently represented as numbers or one-hot vectors. For CLIP, to better utilize semantic information, these labels are often transformed into sentences, e.g. 'A photo of dogs'. And then text feature vectors, $\mathbf{T}$ are extracted from these sentences via f^T. Concurrently, images are encoded into visual feature vectors, $\mathbf{I}$, via f^I. Cosine similarities between $\mathbf{T}$ and $\mathbf{I}$ are used to training and predicting.

FedAVG In FedAVG [33], each client trains f with local clients' data, and then parameters of updated models, w_i, are transmitted to the server. The server typically aggregates the parameters according to Eq. 27,

$$w^* = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} w_i \tag{27}$$

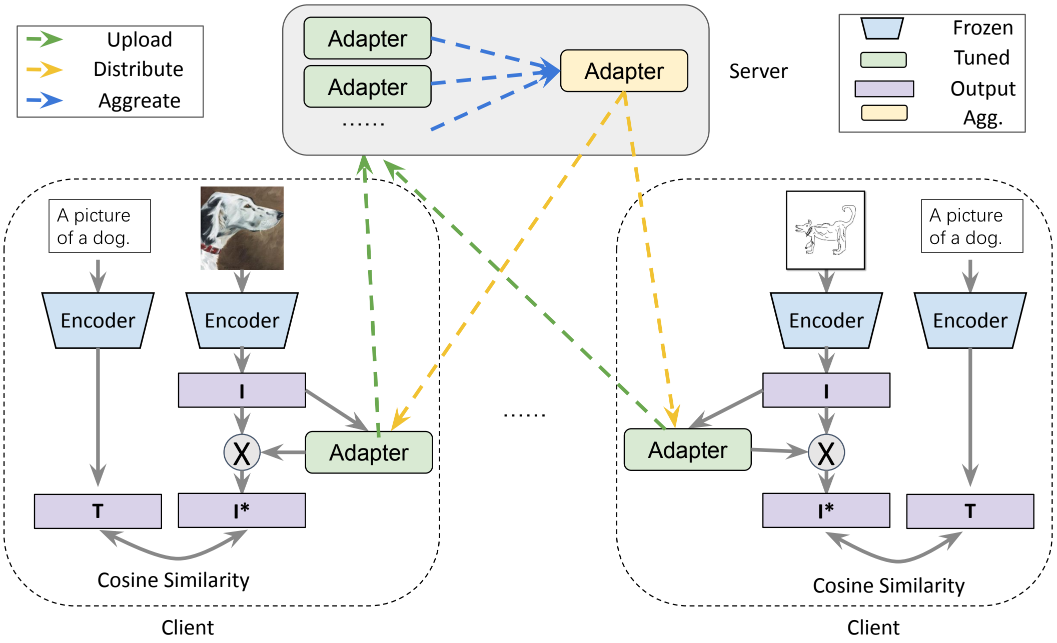


Figure 2: The framework of FedCLIP.

After aggregation, w^* is distributed. When $|w|$ is larger, the server cannot afford communication costs.

3.3 FedCLIP

To reduce computational costs and communications and make the most use of existing pretrained model information, we propose FedCLIP. Pretrained models already have abilities to extract robust and diversified features. Tuning whole networks with limited data can compromise the original ability of pretrained models. What we need to do is to try our best to preserve useful prior knowledge and let it be used to a suitable extent for our task. Besides, tuning large networks is impractical in federated learning due to limited resources in reality. Therefore, instead of operating on the whole model, FedCLIP concentrates on a simple attention-based adapter for the image encoder, AttAI.

Figure 2 gives the framework of FedCLIP. As shown in Figure 2, our method mainly contains four steps.

1. For Client i, we utilize a pretrained CLIP model to extract features of data, denoted as T_i and I_i.
2. In each client, we utilize $\mathcal{D}_i^{train}$ to train the corresponding adapter, g_i. And then we upload $\{g_i\}_{i=1}^N$ to the server.
3. In the server, the parameters of all g_i are weighted averaged and we can obtain g^*. The server then distributes g^* to each client and updates the parameters of each g_i.
4. Repeat Step 2 and Step 3 until convergence or reaching maximum rounds.

In step 1, we utilize the pretrained CLIP model to extract features. We consider the pretrained model is so powerful that we do not need to explore some other features. For $(\mathbf{x}, y)$, we can obtain corresponding features,

$$\mathbf{I} = f^I(\mathbf{x}), \mathbf{T} = f^T(y) \tag{28}$$

What we need to do next is to identify which parts of features are suitable for our specific tasks. Therefore, we introduce an attention-based adapter, g, to locate where we should concentrate on. Particularly, we utilize one linear layer, Tahn activation function, one linear layer, and Softmax activation function to construct g. The Softmax function is used to ensure our final outputs ranging from 0 to 1. Once we obtain the attention vector $att = g(\mathbf{I})$, we utilize it to update the visual feature via a dot multiply operation,

$$\mathbf{I}^* = g(\mathbf{I}) \cdot \mathbf{I}. \tag{29}$$

Then, similar to [36], we normalize $\mathbf{I}^*$ and $\mathbf{T}$ to compute the final logits.

$$\mathbf{I} = \frac{\mathbf{I}^*}{|I^*|}, \mathbf{T} = \frac{\mathbf{T}}{|\mathbf{I}|}, \tag{30}$$

$$\hat{\mathbf{I}} = s \times \mathbf{I} \times \mathbf{T}^T, \hat{\mathbf{T}} = \hat{\mathbf{I}}^T. \tag{31}$$

where s is a scale parameter.

Now, we can utilize the ground truth, a vector $\tilde{\mathbf{y}} = [0, 1, 2, 3, \cdots, B]$

$$\begin{aligned} \ell_{cls}^I &= \ell(\hat{\mathbf{I}}, \tilde{\mathbf{y}}), \\ \ell_{cls}^T &= \ell(\hat{\mathbf{T}}, \tilde{\mathbf{y}}), \end{aligned} \tag{32}$$

where ℓ is CrossEntropy loss [57] while B is the number of images in a batch.

We only exchange parameters of adapters, w^g, and therefore in the server, we replace Eq. 27 with Eq. 33.

$$w^{g,*} = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} w_i^g. \tag{33}$$

Since w^g contains substantially less amount of trainable parameters than w, FedCLIP saves computational costs and communication costs.

3.4 Summary

For clarity, we give a detailed description of FedCLIP in Algorithm 1. In Line 1, directly obtaining generalized and diversified features with fixed CLIP make it possible to utilize more prior knowledge of pretrained models. In Line 2, with adapters, we can concentrate on valuable information and eliminate the influence of redundant information in specific tasks. Rich prior knowledge and targeted attention make the ultimately extracted features more robust, effective, and adaptable, resulting in our method having good generalization and personalization capabilities. From Line 2 to Line 5, performing computation and transmission merely with adapters can save a lot of resources and ensure the efficiency of our method.

3.5 Discussion

Adapter is a common technique in transfer learning [18]. It is at a small scale and has a plug-and-play implementation. In this paper, we mainly focus on adaptations to image encoders. Actually, we also can add adapters to text encoders. We can even change the inputs of text encoders to incorporate more semantic information.

4 Experiments

In this section, we extensively evaluate FedCLIP in three common visual image classification benchmarks.

Algorithm 1 FedCLIP

```
Input: $N$ clients' datasets $\{\mathcal{D}_i\}_{i=1}^N$, a pretained CLIP model consist of an image encoder, $f^I$, and a text encoder, $f^T$
Output: An adapter $g$
For client $i$, computer the corresponding features $I_i = f^I(\mathbf{X}_i), T_i = f^T(\mathbf{Y}_i)$
For client $i$, train the local adapter, $g_i$, according to Eq.5 to Eq.9
Send the current adapter $g_i$ to the server
Aggregate adapters' parameters via Eq. 33 and obtain $w^{g*}$
Transmit $w^{g*}$ to each client
Repeat steps $2 \sim 5$ until convergence
```

4.1 Datasets

PACS PACS [23] is a popular object classification benchmark. It is composed of four sub-datasets, including photo, art-painting, cartoon, and sketch. There exist $9,991$ images in total and the dataset contains 7 classes, including dog, elephant, giraffe, guitar, horse, house, and person. Large discrepancies in image styles widely exist among different sub-datasets. In this paper, we view each sub-dataset as a client. We choose three sub-datasets as participated clients while the rest served as the target client to evaluate generalization ability. For each participated client, we split the corresponding sub-dataset into three parts, 60% for training, 20% for validation, and the rest 20% for testing. Validation parts of data are used for model selection.

VLCS VLCS [10] is another widely accepted public image classification benchmark. It also consists of four sub-datasets (VOC2007, LabelMe, Caltech10, and SUN09). It contains $10,729$ instances with 5 classes. Feature shifts exist generally among different sub-datasets. Similar to PACS, four sub-datasets correspond to four clients. Three sub-datasets play the roles of participants while the rest one act as an upcoming client.

Office-Home Office-Home [48] is a larger image classification benchmark, which contains 65 classes. Office-Home comprises four sub-datasets (Art, Clipart, Product, and Real_World) with about $15,500$ images. The feature shifts from Office-Home mainly come from image styles and viewpoints, but they are much smaller than PACS. We assess methods on Office-Home in a similar manner to PACS.

4.2 Implementation Details and Comparison Methods

For these three common image classification benchmarks, we use the CLIP pre-trained model with ViT-B/32 [9] as the image encoder. For model training, we utilize cross-entropy loss and Adam optimizer. The learning rate is tuned from 5×10^{-5} to 5×10^{-3}. We set local update epochs as $E = 1$ where E means the number of training epochs in one round while we set the total communication round number as $R = 200$. Since, at each time, we set one sub-dataset as the target, i.e. upcoming client, there exist four tasks for each benchmark. We run three trials to record the average results. To better illustrate the function and necessity of using larger pretrained models, we also utilize a related small architecture, AlexNet [21], to perform some base federated learning methods.

We compare our method with two methods including a common federated learning method, FedAVG, and a method designed for non-iid data, FedProx.

1. FedAVG [33]. The server aggregates all client models' parameters. FedAVG will aggregate networks with several layers for AlexNet while FedAVG will aggregate both image encoders and text encoders for CLIP.

2. FedProx [25]. It adds a proximal term to FedAVG and allows the existence of slight differences between clients and the server.

Table 10: Generalization accuracy. **Bold** means the best.

Dataset			PACS					Office-Home					
Backbone	Method	A	C	P	S	AVG	Backbone	Method	A	C	P	R	AVG
AlexNet	FedAVG	31.54	43.69	44.55	36.29	39.02	AlexNet	FedAVG	15.70	17.00	31.56	28.99	23.31
	FedProx	29.79	46.80	44.67	35.12	39.09		FedProx	16.48	17.66	29.83	27.98	22.99
CLIP	FedAVG	53.08	80.08	90.00	76.99	75.04	CLIP	FedAVG	65.60	57.64	71.64	75.42	67.57
	FedProx	66.06	87.33	91.68	78.42	80.87		FedProx	65.60	57.64	71.64	75.42	67.57
	Ours	**96.34**	**97.91**	**99.76**	**85.59**	**94.90**		Ours	**78.00**	**63.69**	**87.52**	**87.79**	**79.25**

Table 11: Personalization accuracy. **Bold** means the best.

Dataset			PACS				Office-Home						
Target	BackBone	Method	C	P	S	AVG	Target	BackBone	Method	C	P	R	AVG
A	AlexNet	FedAVG	72.86	61.08	78.22	70.72	A	AlexNet	FedAVG	50.74	63.47	38.81	51.01
		FedProx	71.37	56.89	81.53	69.93			FedProx	51.78	66.74	40.07	52.86
	CLIP	FedAVG	76.28	86.83	42.42	68.51		CLIP	FedAVG	64.38	79.14	78.76	74.09
		FedProx	90.81	90.42	63.95	81.73			FedProx	64.38	79.14	78.76	74.09
		Ours	**97.65**	**99.40**	**86.75**	**94.60**			Ours	**68.61**	**87.37**	**88.06**	**81.35**
			A	P	S	AVG				A	P	R	AVG
C	AlexNet	FedAVG	46.45	66.17	75.67	62.76	C	AlexNet	FedAVG	23.51	61.78	41.56	42.28
		FedProx	47.19	64.07	77.45	62.90			FedProx	24.54	64.04	40.18	42.92
	CLIP	FedAVG	84.11	92.81	81.02	85.98		CLIP	FedAVG	73.81	80.38	80.48	78.23
		FedProx	86.06	92.81	85.61	88.16			FedProx	73.81	80.38	80.48	78.23
		Ours	**96.33**	**99.10**	**86.88**	**94.10**			Ours	**78.97**	**87.60**	**87.60**	**84.72**
			A	C	S	AVG				A	C	R	AVG
P	AlexNet	FedAVG	37.65	75.00	81.53	64.73	R	AlexNet	FedAVG	23.30	49.94	40.87	38.04
		FedProx	35.45	73.93	83.57	64.32			FedProx	21.03	48.91	39.84	36.59
	CLIP	FedAVG	83.13	93.38	84.97	87.16		CLIP	FedAVG	70.93	**68.73**	77.73	72.46
		FedProx	83.86	93.59	88.54	88.66			FedProx	70.93	**68.73**	77.73	72.46
		Ours	**97.56**	**97.65**	**86.75**	**93.99**			Ours	**78.35**	68.38	**87.94**	**78.23**
			A	C	P	AVG				A	C	P	AVG
S	AlexNet	FedAVG	53.30	68.80	66.17	62.76	P	AlexNet	FedAVG	22.27	49.14	58.51	43.31
		FedProx	52.32	69.66	66.47	62.82			FedProx	20.21	50.06	58.29	42.85
	CLIP	FedAVG	90.71	94.02	94.91	93.21		CLIP	FedAVG	69.07	66.21	77.79	71.02
		FedProx	91.44	94.66	95.81	93.97			FedProx	69.07	66.21	77.79	71.02
		Ours	**97.31**	**97.65**	**99.40**	**98.12**			Ours	**78.56**	**68.50**	**87.37**	**78.14**

4.3 Results

Generalization Ability We first evaluate the generalization ability of each method via accuracy on clients that do not participate in training. Table 10 shows the generalization results for each task on PACS and Office-Home. We have the following observations from these results. 1) Our method achieves the best generalization ability on average with remarkable improvements (about 14% for PACS and about 12% for Office-Home). Moreover, our method achieves the best generalization ability in each task, which demonstrates the excellent generalization ability of our method. 2) Compared to methods with AlexNet as the backbone, methods with CLIP as the backbone can obtain better performance. It demonstrates that large well-trained models can be able to bring better generalization. 3) Compared to methods with CLIP as the backbone, our method has a further improvement, which demonstrates that our method leverages prior knowledge better.

Table 12: Comprehensive average accuracy. **Bold** means the best

Datasets			PACS						Office-Home		
Backbone	AlexNet		CLIP			Backbone	AlexNet		CLIP		
Methods	FedAVG	FedProx	FedAVG	FedProx	Ours	Methods	FedAVG	FedProx	FedAVG	FedProx	Ours
A	60.93	59.89	64.65	77.81	**95.04**	A	42.18	43.77	71.97	71.97	**80.51**
C	57.99	58.88	84.50	87.95	**95.06**	C	35.96	36.60	73.08	73.08	**79.46**
P	59.68	59.41	87.87	89.42	**95.43**	P	36.42	34.90	72.26	72.26	**80.55**
S	56.14	55.89	89.16	90.08	**94.99**	R	39.73	39.13	72.12	72.12	**80.55**
AVG	58.69	58.52	81.55	86.32	**95.13**	AVG	38.57	38.60	72.36	72.36	**80.27**

Table 13: Comprehensive average accuracy on VLCS. **Bold** means the best

Backbone	AlexNet		CLIP		
Methods	FedAVG	FedProx	FedAVG	FedProx	Ours
C	62.13	61.37	72.48	68.57	**83.68**
L	63.01	63.77	75.04	76.50	**82.62**
S	63.15	63.59	68.13	75.50	**82.82**
V	62.32	62.04	69.55	70.09	**83.30**
AVG	62.65	62.69	71.30	72.67	**83.11**

Personalization Ability Then, we evaluate the personalization ability of each method via the accuracy on test data of each participating client. Table 11 shows the personalization results for each task on PACS and Office-Home. We also have some insightful observations. 1)Although all clients share the same adapter in our method, our method still achieves the best average accuracy. Moreover, FedCLIP almost achieves the best performance on each client for every task. 2) Compared to methods with AlexNet, corresponding methods with CLIP perform better overall. For CLIP-based methods, results are quite sensitive to hyperparameters, e.g. learning rate. And FedAVG has disappointing results on some specific clients. 3) Our method has the most use of prior knowledge since it achieves the stablest results.

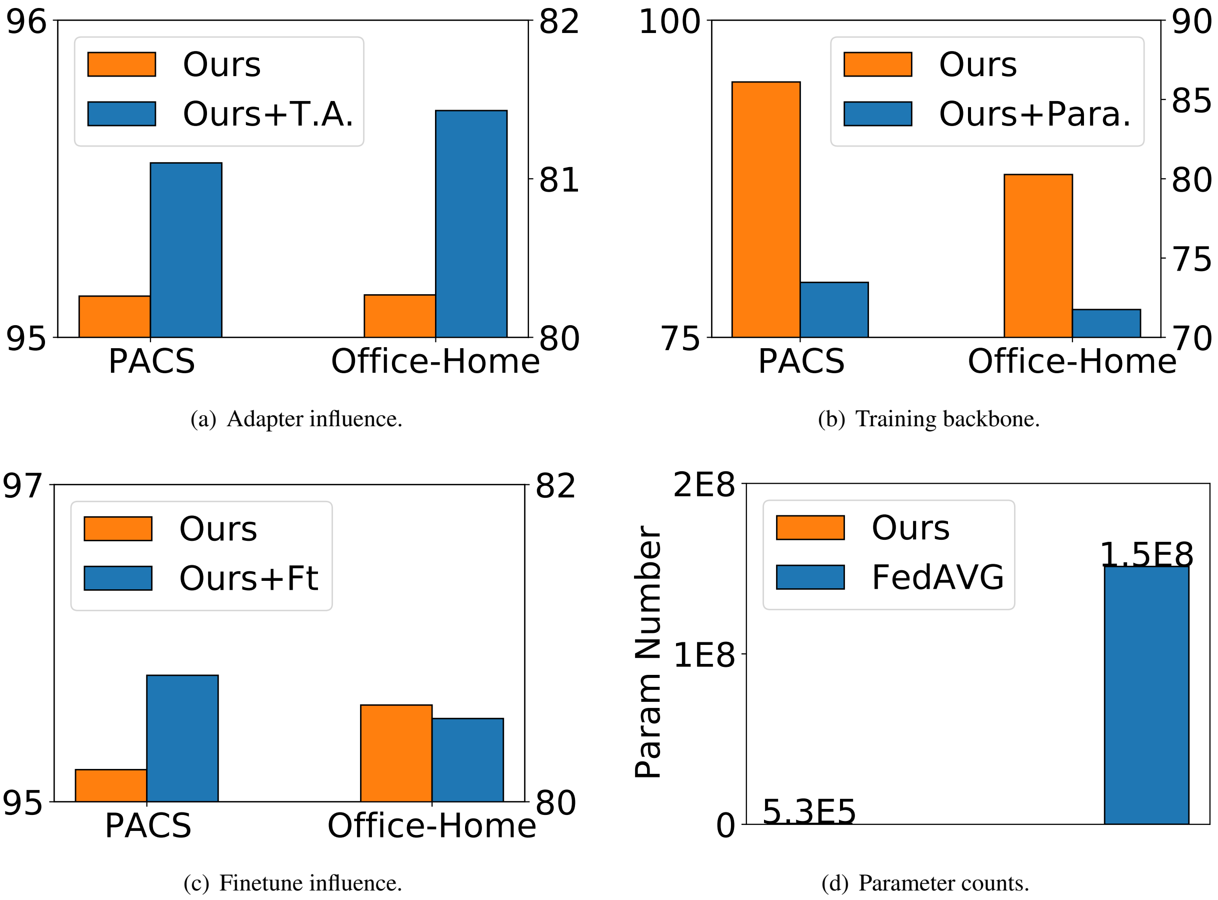
Comprehensive Ability Finally, taking into account the performance of both personalization and generalization, we provide an overall performance in Table 12. Without a doubt, our method achieves the best overall performance with significant improvements (about 9% for PACS and 8% for Office-Home). Compared to methods based on AlexNet, corresponding methods based on CLIP perform better.

More results on VLCS Due to space limitations, we only report comprehensive ability on VLCS. As shown in Table 13, our method still achieves the best performance with improvements of over 10%. Moreover, our method achieves the best in each task. The results prove the superiority of our method again.

4.4 Analysis

Can more adapters bring better performance? In our method, we only add one adapter to the image encoder. We can add another adapter to the text encoder. As shown in Figure 3(a), adding more adapters brings slight improvements. However, the improvements are so small that we need to assess whether it is necessary to do so since more adapters regularly mean more computational costs and more communication costs.

Can more trainable parameters bring better performance? If we train both adapters and the backbones, the results could be worse. Since CLIP models have a wealth of good information, it is not suitable to change



(a) Adapter influence.

(b) Training backbone.

(c) Finetune influence.

(d) Parameter counts.

Figure 3: Analysis on PACS.

parameters with only a few data for a specific task. Changes in CLIP with few data can destroy the feature extraction capabilities. As shown in Figure 3(b), we train more parameters but achieve worse performance.

Will finetuning bring better personalization? According to [54], finetuning can be a useful technique for better personalization. We also add experiments with finetune. As shown in Figure 3(c), finetune has no advance in personalization, which demonstrates that our method can be remarkable and robust when meeting non-iid.

Resource Cost Comparison The number of trainable parameters represents how many resources we need to cost in federated learning. As shown in Figure 3(d), our method merely has $5.3E5$ parameters while FedAVG with CLIP requires $1.5E8$ trainable parameters. Common methods via training whole networks have 283 times as many parameters as ours, which illustrates that our method is fast and resource-efficient.

5 Conclusion and Future Work

In this article, we propose FedCLIP, a fast generalization and personalization learning method for CLIP in federated learning. FedCLIP designs an attention based adapter to replace updating the whole model. Therefore, FedCLIP makes the most use of prior knowledge and saves computational costs and communication costs. Comprehensive experiments have demonstrated the superiority of FedCLIP. In the future, we plan to embed

FedCLIP into more architectures and design more flexible adapters for different tasks. We also plan to apply FedCLIP for heterogeneous architectures and more realistic applications.

References

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164, 2018.

[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. stat, 1050:27, 2020.

[3] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh. Federated learning review: Fundamentals, enabling technologies, and future applications. Information processing & management, 59(6):103061, 2022.

[4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, 2021.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[6] D. Caldarola, B. Caputo, and M. Ciccone. Improving generalization in federated learning by seeking flat minima. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII, pages 654–672. Springer, 2022.

[7] H.-Y. Chen and W.-L. Chao. On bridging generic and personalized federated learning for image classification. In International Conference on Learning Representations, 2022.

[8] Y. Chen, W. Lu, X. Qin, J. Wang, and X. Xie. Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. arXiv preprint arXiv:2206.08516, 2022.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[10] 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.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In International Conference on Learning Representations, 2021.

[12] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10112–10121, 2022.

[13] M. W. Gardner and S. Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15):2627–2636, 1998.

[14] T. Guo, S. Guo, J. Wang, and W. Xu. Promptfl: Let federated participants cooperatively learn prompts instead of models–federated learning in age of foundation model. arXiv preprint arXiv:2208.11625, 2022.

[15] S. Gupta, K. Ahuja, M. Havaei, N. Chatterjee, and Y. Bengio. Fl games: A federated learning framework for distribution shifts. In Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022), 2022.

[16] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al. A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence, 45(1):87–110, 2022.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[18] W. Hou, H. Zhu, Y. Wang, J. Wang, T. Qin, R. Xu, and T. Shinozaki. Exploiting adapters for cross-lingual low-resource speech recognition. IEEE ACM Trans. Audio Speech Lang. Process., 30:317–329, 2022.

[19] Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pages 124–140. Springer, 2020.

[20] N. Inkster. China's cyber power. Routledge, 2018.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NeurIPS, volume 25, pages 1097–1105, 2012.

[22] S. Lee, H. Park, D. U. Kim, J. Kim, M. Boboev, and S. Baek. Image-free domain generalization via clip for 3d hand pose estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2934–2944, 2023.

[23] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In Proceedings of the IEEE international conference on computer vision, pages 5542–5550, 2017.

[24] L. Li, Y. Fan, M. Tse, and K.-Y. Lin. A review of applications in federated learning. Computers & Industrial Engineering, 149:106854, 2020.

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

[26] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In International Conference on Learning Representations, 2021.

[27] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou. From distributed machine learning to federated learning: A survey. Knowledge and Information Systems, 64(4):885–917, 2022.

[28] W. Liu, Y. Wen, B. Raj, R. Singh, and A. Weller. Sphereface revived: Unifying hyperspherical face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2):2458–2474, 2022.

[29] W. Lu, Y. Chen, J. Wang, and X. Qin. Cross-domain activity recognition via substructural optimal transport. Neurocomputing, 454:65–75, 2021.

[30] W. Lu, J. Wang, and Y. Chen. Local and global alignments for generalizable sensor-based human activity recognition. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

[31] W. Lu, J. Wang, Y. Chen, S. Pan, C. Hu, and X. Qin. Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition. IMWUT, 2022.

[32] W. Lu, J. Wang, Y. Chen, X. Qin, R. Xu, D. Dimitriadis, and T. Qin. Personalized federated learning with adaptive batchnorm for healthcare. IEEE Transactions on Big Data, 2022.

[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, pages 1273–1282. PMLR, 2017.

[34] N. Paluru, A. Dayal, H. B. Jenssen, T. Sakinis, L. R. Cenkeramaddi, J. Prakash, and P. K. Yalavarthy. Anam-net: Anamorphic depth embedding-based lightweight cnn for segmentation of anomalies in covid-19 chest ct images. IEEE Transactions on Neural Networks and Learning Systems, 32(3):932–946, 2021.

[35] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu. Generalized federated learning via sharpness aware minimization. In International Conference on Machine Learning, pages 18250–18280. PMLR, 2022.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[37] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[38] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In International Conference on Machine Learning, pages 8821–8831. PMLR, 2021.

[39] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al. The future of digital health with federated learning. NPJ digital medicine, 3(1):1–7, 2020.

[40] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. Information Fusion, 90:148–173, 2023.

[41] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. arXiv, 2019.

[42] I. H. Sarker. Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3):160, 2021.

[43] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-iid data. IEEE transactions on neural networks and learning systems, 31(9):3400–3413, 2019.

[44] C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune bert for text classification? In Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18, pages 194–206. Springer, 2019.

[45] I. Tenison, S. A. Sreeramadas, V. Mugunthan, E. Oyallon, E. Belilovsky, and I. Rish. Gradient masked averaging for federated learning. arXiv preprint arXiv:2201.11986, 2022.

[46] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4593–4601, 2019.

[47] E. A. van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting. Chatgpt: five priorities for research. Nature, 614(7947):224–226, 2023.

[48] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5018–5027, 2017.

[49] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10:3152676, 2017.

[50] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, 119:3–11, 2019.

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

[52] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. Nature, 594(7862):265–270, 2021.

[53] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2):1–19, 2019.

[54] T. Yu, E. Bagdasaryan, and V. Shmatikov. Salvaging federated learning by local adaptation. arXiv preprint arXiv:2002.04758, 2020.

[55] H. Yuan, W. R. Morningstar, L. Ning, and K. Singhal. What do we mean by generalization in federated learning? In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022.

[56] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF international conference on computer vision, pages 558–567, 2021.

[57] Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31, 2018.