FEDCLIP: FAST GENERALIZATION AND PERSONALIZATION FOR CLIP IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

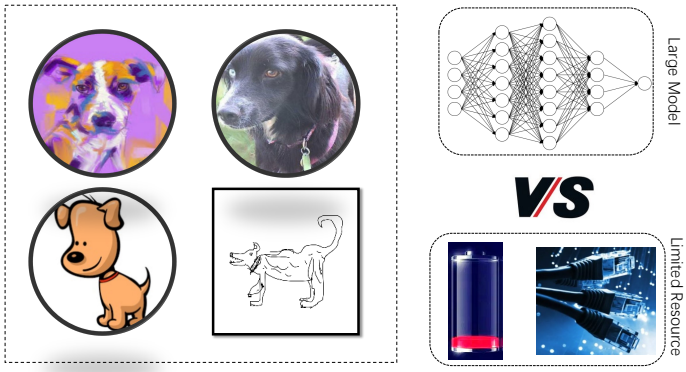
ABSTRACT

When federated learning (FL) meets trustworthy and reliable large-scale models, two critical challenges come: data distribution heterogeneity and high resource costs. 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, 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).

1 INTRODUCTION

Federated learning (FL) makes it possible to perform model aggregation without directly accessing the raw user data from different clients (Yang et al., 2019). This paper studies FL in the scenario of *large models*. For trustworthy and reliable large-scale model learning, traditional FL methods, e.g., FedAVG (McMahan et al., 2017), encounter two problems, including *data distribution shifts* and *huge resource demands*. On the one hand, data distribution shifts widely exist in the real world, e.g., Figure 1(a). When facing heterogeneous data, FL methods suffer from slow convergence and low accuracy due to inconsistent optimization directions, local optima, or some other factors (Gao et al., 2022). 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 (Radford et al., 2021) 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. Therefore, in this paper, we focus on: *how to perform effective and efficient federated learning using these large models?* Specifically, we use CLIP (Radford et al., 2021) as an example.

Some existing work tried to address the issues mentioned above (Lu et al., 2022; Yuan et al., 2022; Guo et al., 2022). FedAP (Lu et al., 2022) 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 (Yuan et al., 2022) 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 (Guo et al., 2022) 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.



(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.

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. Comprehensive experimental results prove that FedCLIP significantly outperforms other baselines (**9%** overall improvements on PACS) and effectively reduces computational and communication costs (**283x** faster).

2 METHOD

2.1 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 $\mathcal{D}_i$:

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

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.

$$\min_f \frac{1}{M} \sum_{i=1}^{M} \frac{1}{m_i} \sum_{j=1}^{m_i} \ell(f(\mathbf{x}_{i,j}), y_{i,j}), \tag{2}$$

2.2 FEDCLIP

A simple CLIP model regularly contains two parts, an image encoder f^I and a text encoder f^T. 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.

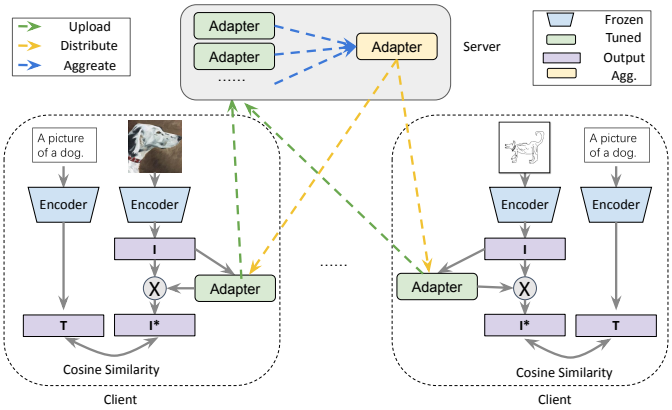


Figure 2: The framework of FedCLIP.

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 while Algorithm 1 gives the detailed steps.

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

1: For client i, computer the corresponding features $I_i = f^I(\mathbf{X}_i), T_i = f^T(\mathbf{Y}_i)$
2: For client i, train the local adapter, g_i, according to Eq.5 to Eq.9
3: Send the current adapter g_i to the server
4: Aggregate adapters' parameters via Eq. 3 and obtain w^{g*}
5: Transmit w^{g*} to each client
6: Repeat steps $2 \sim 5$ until convergence

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. 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. Once we obtain the attention vector $att = g(\mathbf{I})$, we utilize it to update the visual feature via a dot multiply operation, $\mathbf{I}^* = g(\mathbf{I}) \cdot \mathbf{I}$. Then, similar to (Radford et al., 2021), we normalize $\mathbf{I}^*$ and $\mathbf{T}$ to compute the final logits. 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. We only exchange parameters of adapters, w^g,

$$w^{g,*} = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} w_i^g. \tag{3}$$

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

Adapter is a common technique in transfer learning (Hou et al., 2022). 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.

3 EXPERIMENTS

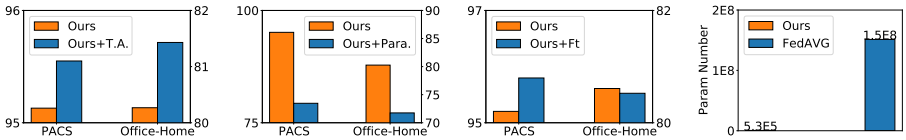
We extensively evaluate FedCLIP in three common visual image classification benchmarks, PACS (Li et al., 2017), VLCS (Fang et al., 2013), and Office-Home (Venkateswara et al., 2017). We use the CLIP pre-trained model with ViT-B/32 (Dosovitskiy et al., 2021) as the image encoder and compare our method with FedAVG (McMahan et al., 2017) and FedProx (Li et al., 2020) [1].

3.1 RESULTS

Table 1: 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**

Taking into account the performance of both personalization and generalization, we provide an overall performance in Table 1[2]. 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.



(a) Adapter influence. (b) Training backbone. (c) Finetune influence. (d) Parameter counts.

Figure 3: Analysis on PACS.

3.2 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 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 (Yu et al., 2020), 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.

4 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.

[1] For more details on datasets and implementations, please refer to Sec. A.1 and Sec.A.2 respectively.

[2] For more results, please refer to Sec. A.3

REFERENCES

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Chen Fang, Ye Xu, and Daniel 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*, pp. 1657–1664, 2013.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong 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*, pp. 10112–10121, 2022.

Tao Guo, Song Guo, Junxiao Wang, and Wenchao Xu. Promptfl: Let federated participants cooperatively learn prompts instead of models–federated learning in age of foundation model. *arXiv preprint arXiv:2208.11625*, 2022.

Wenxin Hou, Han Zhu, Yidong Wang, Jindong Wang, Tao Qin, Renjun Xu, and Takahiro Shinozaki. Exploiting adapters for cross-lingual low-resource speech recognition. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:317–329, 2022. doi: 10.1109/TASLP.2021.3138674. URL `https://doi.org/10.1109/TASLP.2021.3138674`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, volume 25, pp. 1097–1105, 2012.

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

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.

Wang Lu, Jindong Wang, Yiqiang Chen, Xin Qin, Renjun Xu, Dimitrios Dimitriadis, and Tao Qin. Personalized federated learning with adaptive batchnorm for healthcare. *IEEE Transactions on Big Data*, 2022.

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

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*, pp. 8748–8763. PMLR, 2021.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Honglin Yuan, Warren Richard Morningstar, Lin Ning, and Karan 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. URL `https://openreview.net/forum?id=VimqQq-i_Q`.

Table 2: 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**

A APPENDIX

A.1 DATASETS

PACS PACS (Li et al., 2017) 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 (Fang et al., 2013) 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 (Venkateswara et al., 2017) 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.

A.2 IMPLEMENTATION DETAILS AND COMPARISON METHODS

For these three common image classification benchmarks, 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 (Krizhevsky et al., 2012), 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 (McMahan et al., 2017). 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 (Li et al., 2020). It adds a proximal term to FedAVG and allows the existence of slight differences between clients and the server.

A.3 RESULTS

Generalization Ability We first evaluate the generalization ability of each method via accuracy on clients that do not participate in training. Table 2 shows the generalization results for each task

Table 3: 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**

Table 4: 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**

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.

Personalization Ability Then, we evaluate the personalization ability of each method via the accuracy on test data of each participating client. Table 3 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.

More results on VLCS We also report comprehensive ability on VLCS. As shown in Table 4, 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.