

# Efficient Model Personalization in Federated Learning via Client-Specific Prompt Generation

Fu-En Yang<sup>1,2</sup>    Chien-Yi Wang<sup>2</sup>    Yu-Chiang Frank Wang<sup>1,2</sup>  
<sup>1</sup>National Taiwan University    <sup>2</sup>NVIDIA  
 {f07942077, ycwang}@ntu.edu.tw, chienyiw@nvidia.com

## Abstract

Federated learning (FL) emerges as a decentralized learning framework which trains models from multiple distributed clients without sharing their data to preserve privacy. Recently, large-scale pre-trained models (e.g., Vision Transformer) have shown a strong capability of deriving robust representations. However, the data heterogeneity among clients, the limited computation resources, and the communication bandwidth restrict the deployment of large-scale models in FL frameworks. To leverage robust representations from large-scale models while enabling efficient model personalization for heterogeneous clients, we propose a novel personalized FL framework of client-specific Prompt Generation (pFedPG), which learns to deploy a personalized prompt generator at the server for producing client-specific visual prompts that efficiently adapts frozen backbones to local data distributions. Our proposed framework jointly optimizes the stages of personalized prompt adaptation locally and personalized prompt generation globally. The former aims to train visual prompts that adapt foundation models to each client, while the latter observes local optimization directions to generate personalized prompts for all clients. Through extensive experiments on benchmark datasets, we show that our pFedPG is favorable against state-of-the-art personalized FL methods under various types of data heterogeneity, allowing computation and communication efficient model personalization.

## 1. Introduction

With access to web-scale training data (e.g., LAION-5B [43]), deep learning has demonstrated remarkable achievements across computer vision [19, 18, 41] and natural language understanding [12, 54, 2]. However, in real-world scenarios, user data is typically scattered across various domains, such as hospital sites or edge devices. Due to increasing risks of privacy breaches and stricter privacy protection regulations [9], centralized learning schemes are not

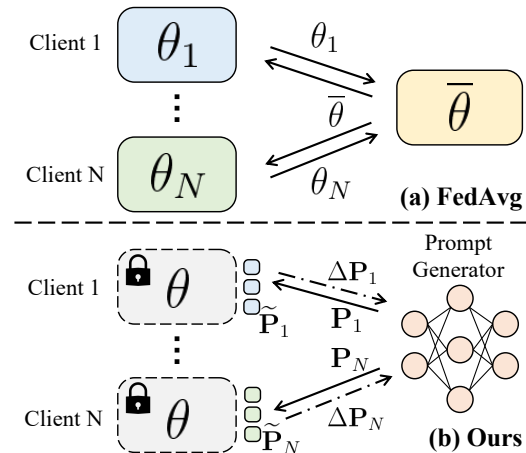


Figure 1. Comparison between (a) FedAvg and (b) our approach. Instead of updating and transporting entire models  $\theta$ , our FL method learns to generate personalized prompts  $\mathbf{P}$  by implicitly observing local optimization directions  $\Delta\mathbf{P} = \tilde{\mathbf{P}} - \mathbf{P}$  for efficient model personalization on top of frozen foundation models.

preferable. With the aim of collaboratively training models without exposing users’ private data, Federated learning (FL) has emerged as a prominent distributed learning framework and has garnered growing research interest. This privacy-preserving learning paradigm has been widely adopted in applications like medical image diagnosis [6], face recognition [31], and person re-identification [57].

Without the need of data sharing among clients, the mainstream FL approach of FedAvg [34] learns a global model by averaging model parameters trained on clients’ private data. However, data distributed in each client might be *heterogeneous* in terms of *domain discrepancy* [29] or *imbalanced class distribution* [26]. Sharing a global model across heterogeneous data clients is prone to highly deviate from their local distribution, leading to severe performance degradation [44, 33]. Previous FL works [28, 26] propose types of constraints (e.g.,  $L_2$  [28] or contrastive regularization [26]) to prevent the local training to be divergent from each other. To better handle the inevitable data het-

erogeneity across clients, personalized federated learning (pFL) methods [44, 33, 4, 52, 45] are instead proposed to allow each client to train a personalized model that adapts to their own data distribution. For example, pFedHN [44] introduces a hypernetwork at the server to directly generate model parameters for each client, whereas pFedLA [33] learns a layer-wise model aggregation policy to assign different weights for personalized model aggregation. While the above pFL approaches are desirable for handling heterogeneous data, they are typically restricted to small backbone architectures (e.g., LeNet [24]) due to the high complexity of outputting model parameters [44] or aggregation weights [33] for large-scale models. Consequently, the capability of derived features is limited, leading to a lack of performance improvement and training instability.

Recently, training from large foundation models [1] for downstream tasks has become a prominent paradigm in centralized learning. To leverage the strong representations derived by foundation models for alleviating data heterogeneity, ViT-FL [40] incorporates pre-trained Vision Transformer (ViT) [13] into standard FL algorithms (e.g., FedAvg [34]) and shows improved robustness and stability on heterogeneously distributed data. However, the use of large pre-trained models for all clients in existing FL algorithms can cause extensive computational and communication burdens, as these methods require transporting entire model parameters between clients and the server. Additionally, overfitting issues might occur when large-scale models are trained with relatively limited client data.

For efficiently tuning large-scale models, prompt learning [21, 55, 56] provides a flexible way to adapt pre-trained models to downstream tasks by solely training the additional inserted trainable parameters (i.e., prompts). For instance, VPT [21] treats prompts as task-specific parameters and prepends them to the input tokens of a pre-trained ViT. In this way, prompts could be optimized to capture task-specific information while instructing a frozen model to perform tasks of interest. However, a straightforward way to adopt prompt learning into FL, i.e., simply averaging prompts learned from all clients, cannot address data heterogeneity among clients effectively and often leads to unsatisfactory performance (as evident in Tables 1-3). Therefore, there is a crucial challenge to develop new FL methods that can leverage prompt learning effectively while handling data heterogeneity among clients.

In this paper, we aim at achieving efficient model personalization among clients with data heterogeneity. As depicted in Fig. 1, different from conventional FL methods (e.g., FedAvg [34]) that updates and transports entire model parameters, we propose a novel personalized FL scheme of *client-specific Prompt Generation (pFedPG)* that exploits underlying client-specific characteristics to produce personalized prompts for each client, which enables efficient adap-

tation to local data distribution. To be more precise, each client trains the client-specific prompts to instruct a model to perform recognition tasks on the target client using its private data. As the local training is not required to update entire large models, the computation overload could be minimized while the possible overfitting issues are mitigated accordingly. On the other hand, we employ a personalized prompt generation module on the server side, which is learned to obtain the underlying optimization directions among clients. With such client characteristics implicitly observed, we are capable of producing personalized prompts to facilitate efficient adaptation for each client with heterogeneous data distribution. By iteratively training the above two stages in a mutually beneficial manner, we are capable of achieving effective yet efficient model personalization on top of the robust representations derived from large-scale foundation models.

We now summarize the contributions of this work below:

- We propose a personalized FL framework of client-specific Prompt Generation (pFedPG), which alternates between *personalized prompt generation* and *personalized prompt adaptation* to enable efficient model personalization under heterogeneous data.
- We design a client-specific prompt generator at the server, which effectively exploits personalized optimization directions and produces client-specific prompts for updating each client model.
- Evaluations on several benchmark datasets in domain discrepancy and imbalanced class distribution verify that our method performs favorably against existing personalized FL approaches and exhibits sufficient training efficiency.

## 2. Related Works

**Federated Learning (FL)** Federated Learning is a learning framework in machine learning with the goal of training models from distributed data sources while protecting data privacy. The most widely recognized approach for federated learning is FedAvg [34], which partitions the learning process into local training and global averaging. However, data distributed in real-world scenarios are typically non-IID, indicating the presence of domain discrepancy or imbalanced class distribution among clients. Directly averaging models trained on heterogeneous data can lead to severe performance degradation and training instability. To address this challenge, several methods [28, 22, 26, 48, 49, 53, 35] have been proposed to regularize local training in FedAvg [34]. For instance, FedProx [28] and SCAFFOLD [22] restrict the local update to be consistent by  $L_2$  distance over model weights and variance reduction technique over gradients, respectively. MOON [26] applies a contrastive objective to

regularize the optimization of local models, ensuring that they do not deviate significantly from the global model.

**Personalized Federated Learning (pFL)** Instead of constructing a global model shared among all clients, personalized FL algorithms [29, 14, 8, 27, 44, 33, 4, 37, 52, 45, 10, 46, 3] are proposed to address data heterogeneity issues by learning customized models at each client. Several works [8, 37, 4] achieve model personalization by only aggregating parts of a model (*e.g.*, feature extractor) at the server while keeping or learning additional modules (*e.g.*, classifier) locally. Per-FedAvg [14] analogizes the local training and server aggregation processes as inner and outer loops optimization in model-agnostic meta-learning [15], facilitating local model adaptation from the global model initialization. PartialFed [48] and FedALA [52] derive customized models by adaptively aggregating the global and local models. Similarly, pFedLA [33] learns a layer-wise aggregation policy to construct a personalized model by assigning larger weights to clients with higher similarities. Some recent works [10, 46, 3] achieve model personalization by either learning sparse models or applying adapter layers. Instead of employing average-based aggregation at the server, pFedHN [44] directly generates model parameters for all clients. However, its applicability is limited to small and shallow models (*e.g.*, LeNet [24]) due to the high complexity of the model parameter space.

**Foundation Models and Prompt Learning** Leveraging publicly available pre-trained foundation models [1, 13, 19, 18, 41] to downstream tasks has emerged as a prominent scheme in centralized learning. In particular, Transformer [51, 13] architectures have demonstrated exceptional ability in deriving robust and discriminative representations. In the FL community, some works [40, 36, 5] start to investigate the effectiveness of leveraging foundation models into the FL framework. For instance, ViT-FL [40] first incorporates the pre-trained Vision Transformer (ViT) [13] architecture into FL and shows improved model performance and training stability. However, most FL algorithms typically require updating *entire* model, making the adoption of foundation models challenging in real-world FL scenarios (*e.g.*, edge devices or medical sites) due to limited computation/communication resources.

Prompt learning techniques [30, 32, 25] have been widely used in the NLP community for adapting language models to downstream tasks effectively via only optimizing a small amount of continuous task-specific prompt vectors. Recently, Visual Prompt Tuning (VPT) [21] has also been proposed as an efficient and effective alternative to fully fine-tuning the large-scale ViT model. It introduces additional learnable prompts into the input image embedding space. These prompts act as task-specific parameters, adapt-

ing the frozen backbone model to perform downstream tasks. Very recently, several concurrent works [17, 47] choose to insert prompts to a frozen CLIP [41] text encoder at local clients. While allowing efficient FL, these methods follow FedAvg and adopt *average-based* prompt aggregation, which is not optimal for clients with significant data heterogeneity. Thus, applying prompt learning techniques to data heterogeneous FL scenarios remains an open research challenge. In this work, we propose a unique personalized prompt generation to enable efficient model personalization upon clients with heterogeneous data.

### 3. Proposed Method

#### 3.1. Problem Formulation

For the sake of completeness, we first define the problem setting in this paper. Following previous personalized federated learning works [14, 8, 27, 44, 33, 4, 37, 52], we assume that training data are distributed in  $N$  separated clients with heterogeneous datasets  $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$ , each contains a set of image-label pairs  $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}_n|}$ . These datasets follow *non-IID* (independent and identically distributed) data distribution in terms of either domain discrepancy or imbalanced label space. With the interest of training efficiency and local data privacy preserved, we aim at learning a client-specific prompt generation mechanism that produces  $K$  personalized visual prompts  $\mathbf{P}_n = [p_n^1, p_n^2, \dots, p_n^K]$  that adapt a pre-trained foundation model  $F^*$  to perform classification tasks on each local client. Through our learned client-specific prompts, we enable efficient model personalization for each heterogeneous client while preserving the robust representation from a frozen foundation model without the risks of overfitting.

#### 3.2. Efficient Model Personalization in FL via Client-Specific Prompt Generation

As illustrated in Fig. 2, we propose a personalized federated learning framework of *client-specific Prompt Generation* (pFedPG). To leverage underlying client characteristics and enable efficient model personalization for all clients, pFedPG alternates between the stages of *personalized prompt adaptation* and *personalized prompt generation* at local clients and the global server, respectively.

In the stage of *personalized prompt adaptation*, pFedPG advances the visual prompt learning technique [21] in FL frameworks. A small number of trainable parameters, denoted as *prompts*  $\mathbf{P}_n = [p_n^1, p_n^2, \dots, p_n^K]$ , are inserted into a frozen foundation model  $F^*$  to encode client-specific information at client  $n$ . In the stage of *personalized prompt generation*, a personalized prompt generator  $G$  is learned to produce personalized prompts for each client by exploiting the underlying characteristics among clients. Once the learning process is complete, we are able to efficiently

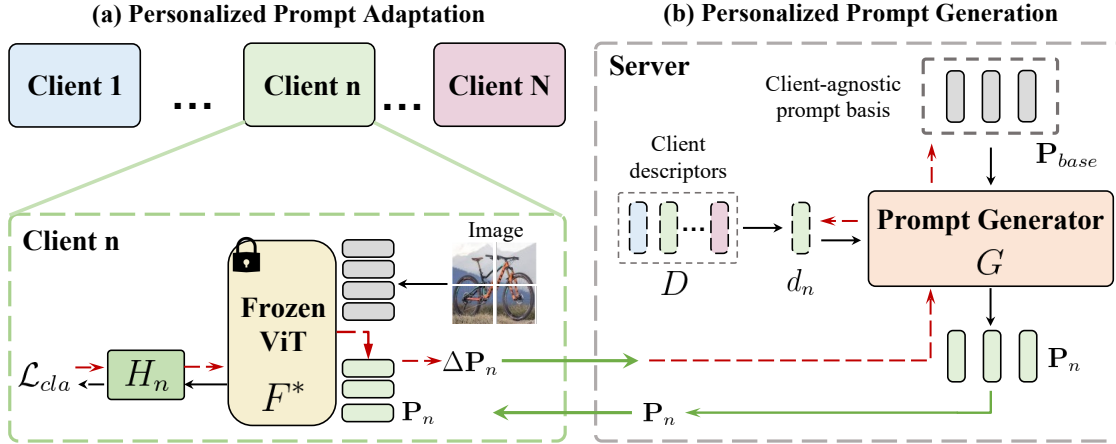


Figure 2. Overview of our client-specific Prompt Generation (pFedPG) framework. pFedPG learns a prompt generator  $G$  together with client-agnostic prompt basis  $\mathbf{P}_{base}$  and a bank of client descriptors  $D = \{d_n\}_{n=1}^N$  at the server. With local classification loss observed, both client-specific prompts  $\mathbf{P}_n$  and local classification head  $H_n$  are updated at each client  $n$ . We alternate between the stages of (a) *personalized prompt adaptation* and (b) *personalized prompt generation* to enable efficient personalization of foundation models like ViT.

adapt the frozen foundation model  $F^*$  by the client-specific prompts  $\mathbf{P}_n$  to perform recognition tasks at each client  $n$ . We now detail each learning stage, including the training/inference processes below.

### 3.2.1 Personalized prompt adaptation at local clients

To enable efficient model adaptation on top of large-scale foundation models and prevent possible overfitting problems caused by updating on relatively limited private data, we advance *Personalized Prompt Adaptation* based on the prompt learning [21] scheme. Note that, the prompts could be treated as client-specific learnable parameters and directly optimized through gradients during training. With the prompts learned, we can efficiently adapt the foundation model  $F^*$  to the data distribution of interest.

As depicted in Fig. 2(a), this training stage aims to learn client-specific prompts  $\mathbf{P}_n = [p_n^1, p_n^2, \dots, p_n^K]$  by leveraging the Transformer-based frozen foundation model  $F^*$  with locally updated classification head  $H_n$ . To be more specific, we follow [13] and divide an input image  $\mathbf{x}$  to  $m$  image patches  $\{a^i\}_{i=1}^m$  and then derive the latent embedding  $\mathbf{z}$  by a frozen feature embedding module Embed as follows:

$$\begin{aligned} \mathbf{x} &= [a^1, a^2, \dots, a^m], \quad a \in \mathbb{R}^{3 \times h \times w}, \\ \mathbf{z} &= [z^1, z^2, \dots, z^m], \quad z = \text{Embed}(a), \end{aligned} \quad (1)$$

where  $h$  and  $w$  denote the height and width of an image patch, and the patch embedding  $z^m$  is projected to  $l$ -dimension. Once the latent embedding  $\mathbf{z}$  is obtained, we form the input embedding of the Transformer encoder  $F^*$  by concatenating  $\mathbf{z}$  with a classification token  $c \in \mathbb{R}^l$  (pre-trained with the ViT backbone), and the client-specific prompts  $\mathbf{P}_n = [p_n^1, p_n^2, \dots, p_n^K]$  as  $[c, \mathbf{P}_n, \mathbf{z}]$ . To encourage

the client-specific prompts to adapt upon this client's data, we employ the standard cross-entropy loss  $\mathcal{L}_{cla}$  over  $|\mathcal{D}_n|$  samples, and is calculated as:

$$\mathcal{L}_n = \frac{1}{|\mathcal{D}_n|} \sum_{j=1}^{|\mathcal{D}_n|} \mathcal{L}_{cla}(H_n(F^*([c, \mathbf{P}_n, \mathbf{z}_j])), y_j). \quad (2)$$

As a result, the client-specific prompts  $\mathbf{P}_n$  can be optimized end-to-end by gradient descent (the same as  $H_n$ ) with learning rate  $\gamma$  as  $\tilde{\mathbf{P}}_n \leftarrow \mathbf{P}_n - \gamma \cdot \partial(\mathcal{L}_n)/\partial \mathbf{P}_n$ .

With personalized prompt adaptation, pFedPG is able to realize parameter-efficient model adaptation without requiring updating entire model parameters yet mitigating possible overfitting concerns and huge computation workloads.

### 3.2.2 Personalized prompt generation at the server

Conventional FL methods (e.g., [34]) typically adopt average-based model aggregation at the server. However, this aggregation manner poses a significant risk of deviating from local data distributions and introduces massive communication overheads, especially when deploying large-scale models among heterogeneous clients. Recall that the prompts trained locally could be treated as client-specific parameters to adapt the frozen model to the client of interest. Instead of averaging model parameters or prompts from clients, we aim at learning a unique personalized prompt generation mechanism at the server to exploit cross-client knowledge and then produce personalized prompts that serve as a good initialization to facilitate efficient local adaptation. Since the server cannot access local private data, it is challenging to obtain the client-specific characteristics for encouraging the produced personalized prompts to boost local adaptation. In the following, we will elaborate



on how our personalized prompt generation be learned in the FL scheme.

**Design and architecture** As illustrated in Fig. 2(b), with the goal of generating personalized prompts  $\{\mathbf{P}_1, \dots, \mathbf{P}_N\}$  for all  $N$  clients, our pFedPG learns to transform a set of client-agnostic prompt basis  $\mathbf{P}_{base}$  through a conditional prompt generator  $G(\cdot; \varphi)$  parameterized by  $\varphi$  with the guidance of client descriptor  $d_n$  selected from  $D = \{d_1, d_2, \dots, d_N\}$ . To be more specific, we realize the conditional prompt generator  $G$  based on cross-attention [51] while the client-agnostic prompts  $\mathbf{P}_{base}$  and the client descriptor  $d_n$  are expected to capture client-agnostic information and encode the client-specific characteristics, respectively. As a result, generating personalized prompts could be achieved by retrieving client-relevant knowledge from  $\mathbf{P}_{base}$  through the query of the client descriptor  $d_n$ , as formulated below,

$$\begin{aligned} \mathbf{P}_n &= G(\mathbf{P}_{base}, d_n) = \mathbf{P}_{base} + \text{Atten}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) W^\mathcal{O} \\ &= \mathbf{P}_{base} + \text{Softmax}\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{l_k}}\right)\mathcal{V}W^\mathcal{O}, \end{aligned} \quad (3)$$

where  $\mathcal{Q} = [d_n]W^\mathcal{Q}$ ,  $\mathcal{K} = \mathbf{P}_{base}W^\mathcal{K}$ ,  $\mathcal{V} = \mathbf{P}_{base}W^\mathcal{V}$ ,

where  $\sqrt{l_k}$  is a scaling factor and  $l$  is the embedding dimension.  $W^\mathcal{Q} \in \mathbb{R}^{l \times l_k}$ ,  $W^\mathcal{K} \in \mathbb{R}^{l \times l_k}$ ,  $W^\mathcal{V} \in \mathbb{R}^{l \times l_v}$ , and  $W^\mathcal{O} \in \mathbb{R}^{l_v \times l}$  are learnable projection matrixes, where  $l_k$  and  $l_v$  are internal dimensions, as in [51].

**Learning of personalized prompt generation** As the goal of personalized prompts is to serve as a good initialization for each client that facilitates the local adaptation, we learn our personalized prompt generation module (*i.e.*,  $G$ ,  $\mathbf{P}_{base}$  and  $d_n$ ) through the training rewards observed from the local optimization process. Inspired by [44, 33], the change of prompts after local training  $\Delta\mathbf{P}_n = \widetilde{\mathbf{P}}_n - \mathbf{P}_n$  indicates the direction of local optimization at client  $n$  that could be treated as training feedback, assessing the quality of the server-generated prompt initialization for each client. With  $\Delta\mathbf{P}_n$  observed, we are capable of training our pFedPG end-to-end via gradient descent.

To be more specific, the update of the conditional prompt generator  $G(\cdot; \varphi)$  can be derived by the gradients computed locally and expressed by the chain rule as

$$\begin{aligned} \Delta\varphi &= \nabla_\varphi \mathcal{L}_n = (\nabla_\varphi \mathbf{P}_n)^T \nabla_{\mathbf{P}_n} \mathcal{L}_n \\ &\cong (\nabla_\varphi \mathbf{P}_n)^T \Delta\mathbf{P}_n, \end{aligned} \quad (4)$$

where  $\nabla_{\mathbf{P}_n} \mathcal{L}_n$  is approximated by  $\Delta\mathbf{P}_n$  that indicates the optimization direction of local training. We apply the same optimization rule to learn the client-agnostic prompts  $\mathbf{P}_{base}$  and client descriptor  $d_n$  end-to-end with  $G$ , and summarize

---

### Algorithm 1 pFedPG for Efficient and Personalized FL

---

**Input:** Number of communication rounds  $T$ ,  $F^*$ ,  $G$ ,  $\mathbf{P}_{base}$ ,  $D$ , and  $N$  sets of  $\mathbf{P}_n$  and  $H_n$ ,  $n \in [1, N]$

**Data:**  $N$  labeled datasets  $\mathcal{D}_n$ ,  $n \in [1, N]$

**Output:**  $F^*$ ,  $H_n$ ,  $\mathbf{P}_n$

```

1: Let  $t = 0$ ;
2: while  $t < T$  do
3:   # Personalized prompt adaptation at clients
4:   for  $n$  in  $1 : N$  do
5:     Keep  $F^*$  freeze;
6:     Set  $\mathbf{P}_n = G(\mathbf{P}_{base}, d_n)$ ,  $d_n \in D$  (Eq. (3));
7:     Randomly sample a minibatch from  $\mathcal{D}_n$ ;
8:     Update  $H_n$  with  $\mathcal{L}_n$  (Eq. (2));
9:     Update  $\mathbf{P}_n$  by  $\widetilde{\mathbf{P}}_n \leftarrow \mathbf{P}_n - \gamma \frac{\partial(\mathcal{L}_n)}{\partial \mathbf{P}_n}$ ;
10:     $\Delta\mathbf{P}_n = \widetilde{\mathbf{P}}_n - \mathbf{P}_n$ ;
11:   end for
12:   # Personalized prompt generation at the server
13:   Receive  $\Delta\mathbf{P}_n$  from all  $N$  clients;
14:   Update  $G$ ,  $\mathbf{P}_{base}$ , and  $D$  by Eq. (5);
15:    $t = t + 1$ ;
16: end while

```

---

the gradient update as follows,

$$\begin{aligned} \varphi &\leftarrow \varphi - \alpha \nabla_\varphi \mathbf{P}_n^T \Delta\mathbf{P}_n, \\ \mathbf{P}_{base} &\leftarrow \mathbf{P}_{base} - \alpha \nabla_{\mathbf{P}_{base}} \varphi^T \nabla_\varphi \mathbf{P}_n^T \Delta\mathbf{P}_n, \\ d_n &\leftarrow d_n - \alpha \nabla_{d_n} \varphi^T \nabla_\varphi \mathbf{P}_n^T \Delta\mathbf{P}_n. \end{aligned} \quad (5)$$

We note that, the client-agnostic prompt basis  $\mathbf{P}_{base}$  and conditional prompt generator  $G$  are optimized by all clients, enforcing them to exploit cross-client knowledge, while client descriptor  $d_n$  is solely regarding client  $n$ , to encourage the derivation of client-specific characteristics. With our proposed personalized prompt generation module, pFedPG is able to generate personalized prompts to facilitate local adaptation while leveraging learned knowledge across clients without explicitly accessing private data.

### 3.3. pFedPG Training and Inference

In Algorithm 1, we summarize the training details of our proposed pFedPG. We alternate between the learning processes of personalized prompt generation and personalized prompt adaptation until converging.

Once the learning of the proposed framework is complete, we deploy the learned client-specific prompts  $\mathbf{P}_n$  to instruct the pre-trained feature extractor  $F^*$  to extract discriminative representations together with locally trained classification head  $H_n$  for performing the recognition task at each client. Formally, the categorical predictions  $y^*$  over  $Y$  classes at each client  $n$  can be computed as:

$$y^* = \arg \min_{k \in K} H_n(F^*([c, \mathbf{P}_n, \mathbf{x}])). \quad (6)$$

Table 1. Quantitative comparisons on Office-Caltech10 and DomainNet datasets using ViT-B/16. **Bold** denotes the best result.

Datasets	Office-Caltech10 (%)					DomainNet (%)							Comm.
Method	<i>A</i>	<i>C</i>	<i>D</i>	<i>W</i>	Avg.	<i>C</i>	<i>I</i>	<i>P</i>	<i>Q</i>	<i>R</i>	<i>S</i>	Avg.	Cost
<b>Baselines</b>													
SingleSet-Full	80.73	73.33	90.62	94.92	84.90	47.34	37.14	67.21	55.30	84.88	45.13	56.17	-
SingleSet-VPPT [21]	83.33	74.67	96.88	96.61	87.87	57.98	41.55	74.64	59.60	89.56	60.47	63.97	-
FedAvg [40]	89.58	80.44	100.0	100.0	92.51	63.50	38.05	71.89	60.80	78.55	60.47	62.21	$8.58 \times 10^7$
<b>Personalized Federated Learning</b>													
Per-FedAvg [14]	91.67	90.22	100.0	100.0	95.47	69.39	48.71	82.07	35.30	90.63	72.56	66.44	$8.58 \times 10^7$
FedRep [8]	91.15	88.44	100.0	100.0	94.90	64.26	38.20	72.86	<b>62.10</b>	82.66	60.11	63.37	$8.58 \times 10^7$
FedRoD [4]	92.19	90.67	100.0	100.0	95.72	66.54	42.92	74.15	57.20	84.63	66.43	65.31	$8.58 \times 10^7$
FedBABU [37]	89.06	85.78	100.0	100.0	93.71	63.31	43.07	74.80	43.80	87.26	67.15	63.23	$8.58 \times 10^7$
<b>Efficient Federated Learning</b>													
FedVPT [21]	92.71	84.44	100.0	100.0	94.29	65.59	44.14	76.58	47.30	91.04	60.29	64.16	$7.68 \times 10^3$
FedVPT-D [21]	91.67	89.33	100.0	100.0	95.25	63.31	43.07	74.80	54.80	87.26	67.15	65.07	$9.22 \times 10^3$
pFedPG (Ours)	<b>94.79</b>	<b>92.44</b>	<b>100.0</b>	<b>100.0</b>	<b>96.81</b>	<b>73.00</b>	<b>50.08</b>	<b>84.33</b>	60.00	<b>94.00</b>	<b>68.41</b>	<b>71.64</b>	$7.68 \times 10^3$

## 4. Experiments

### 4.1. Datasets and Experimental Setup

#### 4.1.1 Datasets

We evaluate our method on five public benchmark datasets covering types of data heterogeneity, including domain discrepancy and imbalanced class distribution. For *domain discrepancy*, **Office-Caltech10** [42, 16] is composed of four data domains including *Amazon*, *DSLR*, *Webcam*, and *Caltech*. Each domain contains ten classes, with 2,533 images in total. **DomainNet** [39] consists of 0.6 million images of 345 classes distributed across six domains, *Clipart*, *Infograph*, *Painting*, *Quickdraw*, *Real* and *Sketch*. Following [29], we use the top ten most frequent classes to form a sub-dataset for our experiments. As for medical image diagnosis tasks, **Dermoscopic-FL** [6] is comprised of four data sites collected from HAM10K [50] and MSK [7]. Each data site contains three types of skin lesions, with 10,490 images in total. More detailed statistics and sampled images are provided in the supplementary material. For *imbalanced class distribution*, **CIFAR-10** [23] contains 5,000 training images and 1,000 testing images per class, totaling ten classes. **CIFAR-100** [23] consists of 60,000 images of 100 categories with 500 training images and 100 testing images per class.

#### 4.1.2 Experimental settings

To properly evaluate our proposed approach and fairly compare it with existing FL methods, we conduct experiments on two types of heterogeneous FL settings: domain discrepancy and imbalanced class distribution. For conducting clients with *domain discrepancy*, we assign a data domain to a client, indicating the number of clients ( $N$ ) is set as 4, 6, and 4 for Office-Caltech10, DomainNet, and Dermoscopic-

FL datasets, respectively. As for simulating *imbalanced class distribution*, we consider two non-IID settings using CIFAR-10 and CIFAR-100. Following [40], the first non-IID setting we considered is randomly selecting disjoint  $c$  classes for each client and denoted as *disjoint label space*. In our experiments,  $c = 2$  and  $c = 10$  for CIFAR-10 and CIFAR-100, respectively. As for the other non-IID setting, data in each class would be partitioned into all clients following a Dirichlet distribution  $Dir(\alpha)$ . We follow [4] and set  $\alpha$  to 0.1 over 10 clients.

#### 4.1.3 Implementation details

We use ViT-B/16 [13] pre-trained on ImageNet21k [11] as the backbone of  $F^*$  and a single linear layer to realize the classification head  $H_n$ . The input images of all datasets are resized to  $224 \times 224$  pixels. For each client, we train  $\mathbf{P}_n$  and  $H_n$  using the SGD optimizer with a learning rate  $\gamma$  of 0.25 with a weight decay rate of 0.001 and a batch size of 64 for 5 epochs. The number of communication round  $T$  is set to 100. We set the learning rate  $\alpha$  for updating  $G$ ,  $\mathbf{P}_{base}$ , and  $D$  to 0.001. The number of prompts  $K$  of  $\mathbf{P}_n$  and  $\mathbf{P}_{base}$  is set as 10 for datasets except for Dermoscopic-FL with  $K = 3$ . The hyperparameters above are tuned by cross-validation. In all our experiments, we implement our model using PyTorch [38] and conduct training on NVIDIA TESLA V100 GPUs with 32 GB memory.

### 4.2. Quantitative Evaluation

We compare our proposed pFedPG with existing FL methods on benchmark datasets representing various types of data heterogeneity (*i.e.*, domain discrepancy and imbalanced class distribution). In our experiments, *SingleSet-Full* and FedAvg [34] are viewed as baselines, where the former trains a model at each client without information sharing, while the latter aggregates client models to con-

Table 2. Quantitative comparisons on CIFAR-10/100 datasets using ViT-B/16. **Bold** denotes the best result.

Datasets	CIFAR-10 (%)		CIFAR-100 (%)	
	Disjoint	$Dir(0.1)$	Disjoint	$Dir(0.1)$
<b>Baselines</b>				
SingleSet-Full	89.51	83.85	67.74	49.64
SingleSet-VPT [21]	88.91	84.32	63.42	46.46
FedAvg [40]	88.04	79.79	63.33	51.37
<b>Personalized Federated Learning</b>				
Per-FedAvg [14]	88.13	85.14	69.31	52.68
FedRep [8]	87.07	82.40	65.71	50.36
FedRoD [4]	87.61	80.36	63.90	51.42
FedBABU [37]	83.15	76.33	55.91	50.19
<b>Efficient Federated Learning</b>				
FedVPT [21]	89.39	85.11	55.49	45.26
FedVPT-D [21]	89.56	85.43	66.91	50.25
pFedPG (Ours)	<b>90.08</b>	<b>87.57</b>	<b>70.96</b>	<b>55.91</b>

struct a shared global model. In addition, *SingleSet-VPT* indicates each client independently applies visual prompt tuning [21] to learn prompts at the input embedding space.

In Tables 1-3, we summarized the results compared with the state-of-the-art pFL works. To be more specific, Per-FedAvg [14] applies meta-learning [15] to derive customized models for each client from a global initialization. FedRep [8] aggregates feature extractors but keeps classifiers trained locally; FedBABU [37] only updates and shares feature extractors during FL training. FedRoD [4] additionally learns a personalized classification head without model aggregation. Instead of updating entire model parameters, two *efficient* FL baselines, *FedVPT* and *FedVPT-D*, are conducted, which keep the backbone frozen, and aggregate prompts globally. Following [21], FedVPT inserts prompts to the input, and FedVPT-D prepends prompts to the input and hidden layers. Note that, we use ViT-B/16 [13] as the backbone of the above methods for fair comparisons.

In Table 1, we provide the quantitative comparisons on Office-Caltech10 and DomainNet datasets with the presence of **domain shifts** across clients. Our approach achieved the highest 96.81% and 71.64% average accuracies on Office-Caltech10 and DomainNet, respectively, as shown from Table 1. Furthermore, our method demonstrated the best communication efficiency, using only approximately 0.01% of parameters in comparison to other existing pFL methods. Note that the costs of FedRep [8] and FedBABU [37] are the numbers of model parameters of the ViT backbone (*i.e.*, 85.8M), while the communication costs of [34, 14, 4] can be approximated to 85.8M, as they transmit the ViT backbone along with a single-layer classifier, which adds relatively few parameters.

In addition to domain discrepancy, we conducted comparisons on the **imbalanced class distribution** scenario using CIFAR-10 and CIFAR-100 datasets, as shown in Table 2. As mentioned in Sec. 4.1.2, two types of imbalanced

Table 3. Quantitative comparisons on Dermoscopic-FL dataset using ViT-B/16. **Bold** denotes the best result.

Method	A	B	C	D	Avg.
<b>Baselines</b>					
SingleSet-Full	76.09	97.29	71.65	73.57	79.65
SingleSet-VPT [21]	70.90	96.25	70.12	68.33	76.40
FedAvg [40]	62.54	96.12	51.52	68.08	69.57
<b>Personalized Federated Learning</b>					
Per-FedAvg [14]	76.09	91.99	70.12	74.56	78.19
FedRep [8]	69.06	96.12	60.37	68.58	73.53
FedRoD [4]	63.55	96.67	58.84	69.33	72.10
FedBABU [37]	58.19	97.16	49.09	68.58	68.26
<b>Efficient Federated Learning</b>					
FedVPT [21]	74.92	96.77	67.07	75.06	78.46
FedVPT-D [21]	73.91	96.12	74.09	77.81	80.48
pFedPG (Ours)	<b>79.26</b>	<b>97.29</b>	<b>76.22</b>	<b>78.80</b>	<b>82.89</b>

data are simulated, including disjoint label space and imbalanced label distribution drawn from  $Dir(0.1)$ . Table 2 demonstrates that our method performed favorably against existing FL works over the two datasets on both types of label imbalance. To further exhibit the ability of our method to more practical scenarios, we compare with state-of-the-art works for the cross-site medical image diagnosis task using Dermoscopic-FL. As we can observe in Table 3, our pFedPG consistently performed superiorly against other FL methods on all hospital sites.

We observed that, with the presence of significant data heterogeneity (*e.g.*, large style difference in DomainNet) across clients, existing FL works which obtain a shared feature encoder [8, 4, 37] by aggregation might still deviate from local data domains, while Per-FedAvg [14] focuses on deriving a global initialization would not be preferable under severe discrepancy across clients. As shown in Tables 1-3, FedVPT and FedVPT-D achieve comparable or even superior performance over existing FL works, exhibiting the ability of efficient FL methods to mitigate possible overfitting issues. However, sharing a set of global prompts is still not desirable for heterogeneous clients. To explicitly enable efficient model personalization to tackle heterogeneous data, our approach learns to generate personalized prompts to facilitate local adaptation for each client. With the above results, we successfully confirm the effectiveness and robustness of our proposed pFedPG to address data heterogeneity with training efficiency.

### 4.3. Analysis of Our pFedPG

In this section, we first conduct experiments to confirm the effectiveness of our designed personalized prompt generation. Then, we provide a detailed analysis of the impact of different number prompts. Due to the page limitations, we provide the analysis of model backbones and the size of client data in the supplementary material.

Table 4. Analysis of our personalized prompt generation and the architecture of prompt generator  $G$  on benchmark datasets.

Module	Method	Office-Caltech10	DomainNet	CIFAR-10	CIFAR-100
Prompt generation	FedVPT	94.29	64.16	89.39	55.49
	$\mathbf{P}_{base}$	93.16	64.87	88.23	66.89
Architecture of $G$	MLP [44]	94.96	63.33	87.47	66.73
	AdaIN [20]	95.72	70.08	89.77	69.44
	<b>pFedPG</b>	<b>96.81</b>	<b>71.64</b>	<b>90.08</b>	<b>70.96</b>

**Effectiveness of personalized prompt generation** In the upper part of Table 4, we intend to verify the effectiveness of our personalized prompt generation for facilitating adaptation at each client on benchmark datasets, where CIFAR-10/100 are under the setting of disjoint label space. In Table 4, we first ablate  $\mathbf{P}_n$  with the global prompts obtained by global averaging (as in *FedVPT*). As reported in Table 4, the globally averaged prompts cannot achieve satisfactory performance since sharing a single set of prompts would not be favorable to heterogeneous clients. In addition, we examine the performance of applying the trained *client-agnostic prompt basis*  $\mathbf{P}_{base}$  to clients instead of applying personalized prompts  $\mathbf{P}_n$ . We observed that the performance of  $\mathbf{P}_{base}$  is still inferior to ours (which applies  $\mathbf{P}_n$ ). As evident from the above experiments, the effectiveness of our proposed personalized prompt generation for allowing personalized FL under various types of data heterogeneity would be successfully verified.

#### Effectiveness of our designed prompt generator $G$

From the results shown in the lower half of Table 4, we see that the performance dropped when we replaced our cross-attention-based prompt generator  $G$  and  $\mathbf{P}_{base}$  with an MLP-based network as [44], which acts on client descriptors and then output prompts for each client. The inferior performance of the MLP-based prompt generator is due to its high training complexity and instability, resulting from the requirement of deploying a fully-connected layer for each prompt embedding. Another alternative prompt generator is to compute adaptive instance normalization (AdaIN) [20] for  $\mathbf{P}_{base}$  and the client descriptor  $d_n$ . This method allows for the transfer of client-agnostic prompts  $\mathbf{P}_{base}$  to personalized prompts  $\mathbf{P}_n$  by replacing the mean and variance calculated from the client descriptor  $d_n$ , similar to the style transfer approach [20]. However, as seen in Table 4, directly computing AdaIN did not explicitly model the prompt generation process, resulting in inferior performance compared to ours. The results summarized in Table 4 confirm the effectiveness of our designed architecture of prompt generator  $G$ .

**Impact of the number of prompts  $K$**  We also analyze the impact of the number of prompts  $K$  on benchmark

Table 5. Impact of the number of prompts  $K$  on benchmark datasets, where CIFAR-10/100 are drawn from  $Dir(0.1)$ .

$K$	Office-Caltech10	DomainNet	CIFAR-10	CIFAR-100
1	96.09	70.27	86.14	55.77
5	96.77	70.53	87.41	55.79
10	<b>96.81</b>	<b>71.64</b>	<b>87.57</b>	<b>55.91</b>
50	95.10	69.55	85.63	54.52
100	94.53	68.79	85.02	53.61
200	94.46	66.83	83.53	52.34

datasets, and show the results in Table 5. We found that when the number of prompts is set too low (*e.g.*,  $K = 1$ ), the model’s accuracy drops slightly due to insufficient capacity. In contrast, if the number of prompts is set too high, such as 100 or 200, the model’s performance significantly degrades. This is because a large number of prompts may encode noisy and task-irrelevant information, which can adversely affect the quality of the features derived from foundation models. With the above observation, we thus set  $K$  as 10 for these datasets which achieves the best trade-off between communication cost and performance.

## 5. Conclusion

In this paper, we proposed a novel client-specific Prompt Generation framework (pFedPG) for enabling efficient model personalization among heterogeneous clients. By alternative optimization of the proposed personalized prompt generation and client-specific prompt adaptation, our pFedPG is capable of producing personalized prompts for each client by observing underlying directions of local training among clients, while clients optimize such client-specific prompts to adapt a pre-trained model to local data distribution. We conducted extensive quantitative experiments, verifying that our framework performed favorably against SOTA pFL approaches at heterogeneous data clients while achieving training and communication efficiency.

**Acknowledgment** This work is supported in part by the National Science and Technology Council under grant NSTC111-2634-F-002-020 and National Taiwan University under grant NTU-112L900901. We also thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.



## References

- [1] 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. 2, 3
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1
- [3] Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. In *ICML*, 2023. 3
- [4] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*, 2022. 2, 3, 6, 7
- [5] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han-Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. In *ICLR*, 2023. 3
- [6] Zhen Chen, Meilu Zhu, Chen Yang, and Yixuan Yuan. Personalized retrogress-resilient framework for real-world medical federated learning. In *MICCAI*, 2021. 1, 6
- [7] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *ISBI*, 2018. 6
- [8] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, 2021. 3, 6, 7
- [9] Bart Custers, Alan M Sears, Francien Dechesne, Ilina Georgieva, Tommaso Tani, and Simone Van der Hof. *EU personal data protection in policy and practice*. Springer, 2019. 1
- [10] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Displf: Towards communication-efficient personalized federated learning via decentralized sparse training. In *ICML*, 2022. 3
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 4, 6, 7
- [14] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, 2020. 3, 6, 7
- [15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3, 7
- [16] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, 2007. 6
- [17] 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. 3
- [18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 3
- [19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 3
- [20] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 8
- [21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2, 3, 4, 6, 7
- [22] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 2020. 2
- [23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2, 3
- [25] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3
- [26] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021. 1, 2
- [27] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, 2021. 3
- [28] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020. 1, 2
- [29] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fed{bn}: Federated learning on non-{iid} features via local batch normalization. In *ICLR*, 2021. 1, 3, 6
- [30] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3
- [31] Chih-Ting Liu, Chien-Yi Wang, Shao-Yi Chien, and Shang-Hong Lai. Fedfr: Joint optimization federated framework for generic and personalized face recognition. In *AAAI*, 2022. 1
- [32] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 3
- [33] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, 2022. 1, 2, 3, 5

- [34] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. [1](#), [2](#), [4](#), [6](#), [7](#)
- [35] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *CVPR*, 2022. [2](#)
- [36] John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2210.08090*, 2022. [3](#)
- [37] Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *ICLR*, 2022. [3](#), [6](#), [7](#)
- [38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. [6](#)
- [39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. [6](#)
- [40] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *CVPR*, 2022. [2](#), [3](#), [6](#), [7](#)
- [41] 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 *ICML*, 2021. [1](#), [3](#)
- [42] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. [6](#)
- [43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. [1](#)
- [44] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *ICML*, 2021. [1](#), [2](#), [3](#), [5](#), [8](#)
- [45] Yiqing Shen, Yuyin Zhou, and Lequan Yu. Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning. In *CVPR*, 2022. [2](#), [3](#)
- [46] Aliaksandra Shysheya, John Bronskill, Massimiliano Patacchiola, Sebastian Nowozin, and Richard E Turner. Fit: Parameter efficient few-shot transfer learning for personalized and federated image classification. In *ICLR*, 2023. [3](#)
- [47] Shangchao Su, Mingzhao Yang, Bin Li, and Xiangyang Xue. Cross-domain federated adaptive prompt tuning for clip. *arXiv preprint arXiv:2211.07864*, 2022. [3](#)
- [48] Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *NeurIPS*, 2021. [2](#), [3](#)
- [49] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022. [2](#)
- [50] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 2018. [6](#)
- [51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. [3](#), [5](#)
- [52] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. *arXiv preprint arXiv:2212.01197*, 2022. [2](#), [3](#)
- [53] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*, 2022. [2](#)
- [54] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. [1](#)
- [55] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [2](#)
- [56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. [2](#)
- [57] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *ACM MM*, 2020. [1](#)