Unlocking the Potential of Prompt-Tuning in Bridging Generalized and Personalized Federated Learning

Wenlong Deng, Christos Thrampoulidis*, Xiaoxiao Li*
Department of Electrical and Computer Engineering,
The University of British Columbia, Vancouver, BC, Canada
{dwenlong,cthrampo,xiaoxiao.li}@ece.ubc.ca
https://github.com/ubc-tea/SGPT

Abstract

Vision Transformers (ViT) and Visual Prompt Tuning (VPT) achieve state-of-the-art performance with improved efficiency in various computer vision tasks. This suggests a promising paradigm shift of adapting pre-trained ViT models to Federated Learning (FL) settings. However, the challenge of data heterogeneity among FL clients presents a significant hurdle in effectively deploying ViT models. Existing Generalized FL (GFL) and Personalized FL (PFL) methods have limitations in balancing performance across both global and local data distributions. In this paper, we present a novel algorithm, SGPT, that integrates GFL and PFL approaches by employing a unique combination of both shared and group-specific prompts. This design enables SGPT to capture both common and group-specific features. A key feature of SGPT is its prompt selection module, which facilitates the training of a single global model capable of automatically adapting to diverse local client data distributions without the need for local fine-tuning. To effectively train the prompts, we utilize block coordinate descent (BCD), learning from common feature information (shared prompts), and then more specialized knowledge (group prompts) iteratively. Theoretically, we justify that learning the proposed prompts can reduce the gap between global and local performance. Empirically, we conduct experiments on both label and feature heterogeneity settings in comparison with state-of-the-art baselines, along with extensive ablation studies, to substantiate the superior performance of SGPT.

1. Introduction

FL is a framework that allows machine learning models to be learned from multiple clients without sharing their data [37]. In the landscape of computer vision, the integration of ViT [9] with FL emerges as a pivotal research domain

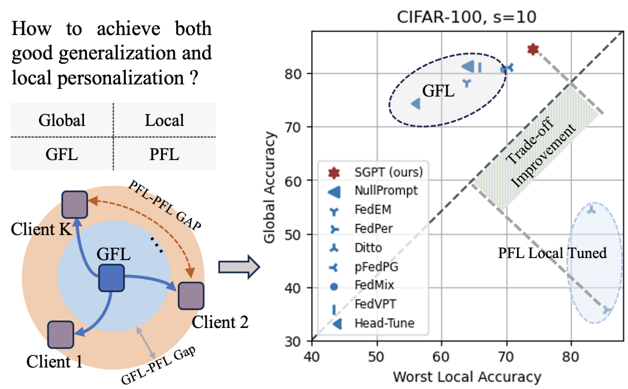


Figure 1. Global accuracy and worst local accuracy on CIFAR-100 with $s = 10$ (s is the number of classes per client). Points located in the top-right corner correspond to great performance on both the global data and local clients' data distributions. PFL models perform well on local data, however, lack the ability to predict out-of-client data. Global models have a better generalization but cannot well adapt to each local data distribution. Our proposed *SGPT* ($\star$) achieves the best trade-off.

as it promises a significant improvement in image recognition tasks. Firstly, ViT's attention mechanism has demonstrated exceptional ability in deriving robust and discriminative representations [2, 3, 9, 51] in terms of scalability and adaptability to a variety data scenarios; a feature crucial in FL environments. Secondly, ViTs demonstrate a remarkable ability to generalize from limited data by leveraging powerful publicly available pre-trained ViT models [9, 17, 19, 22] as initialization, making them inherently suitable for FL's decentralized nature. While ViTs are often seen as computationally demanding, recent advancements have significantly enhanced their efficiency casting them suitable for FL[1]. These improvements include 1) updating part of model parameters [47] or 2) optimizing additional parameters with

*Equal Corresponding Author.

[1]A detailed related work review on parameter efficient tuning for transformer is provided in the appendix.

a frozen model [28, 57], during local training of FL followed by federated averaging [37]. In this paper, we focus on the latter approach by employing Visual Prompt Tuning (VPT) [19] given its efficiency and effectiveness in vision tasks [19, 47, 56].

Although prompt tuning techniques allow efficient FL, applying them to certain FL scenarios still remains an open research challenge [56], particularly when data across clients exhibits heavy heterogeneity in terms of domain discrepancy [31] or imbalanced class distribution [29]. To navigate *data heterogeneity*, two primary strategies are used: Generalized FL (GFL) and Personalized FL (PFL). On the one hand, GFL approaches focus on learning a single global FL model that achieves high generalization, with methods like FedAvg variants [21, 30, 31]. On the other hand, PFL tailors models to individual clients and clustered client groups. For instance, some PFL methods [12, 55] involve local data fine-tuning to customize a global model for personalized models. Others [4, 14, 35] take into account client similarity and integrate clients with similar data distributions into clusters. In the regime of FL with ViT, FedPR [13], a recent prompt-tuning-based GFL method, learns client prompts and aggregates them into global prompts. Recently, FedPG [56], another PFL method, uses a Hyper-Network [44] to generate client-specific prompts. Both GFL and PFL methods have their own limitations: (1) GFL methods are insufficient when dealing with significant data heterogeneity [30] with one global model; (2) PFL customizes client models, which can lead to overfitting on local data [55]. Overall, this limits their ability to generalize to other distributions and may not allow them to adapt to out-of-federation clients.[2]

To overcome the limitations of GFL and PFL methods under data heterogeneity, it is essential to leverage their respective strengths through a combination of the two. We show that this is possible by appropriately leveraging prompt-tuning. Concretely, we achieve this by developing a new FL algorithm Shared and Group Prompt Tuning (dubbed *SGPT*). Our algorithm focuses on learning a shared global model during training, allowing it to acquire global information, while also enabling local adaptation with prompt selection. This approach leads to high accuracy in generalizing to global distribution as well as efficient alignment to various local client data distributions (see Fig. 1). To elaborate, firstly, *SGPT* globally learns **shared** and **group** prompts, which facilitate the learning of both universal and group-specific knowledge. Secondly, the prompt selection module (see Fig.2 (b)) effectively finds data groups and assigns group prompts to each input, thus *automatically aligning the global model with local distributions* (see Fig.2 (c)), without needing local fine-tuning. Thirdly, we use block-coordinate-descent (BCD) for effective parameter training, starting with learning common features (shared prompts) before optimizing group prompts for specialized knowledge iteratively. Finally, we present a theoretical error bound and identify two factors *generalization* and *distribution discrepancy* that affect *the gap between the global and local performance*. Our *SGPT* effectively considered these two terms. In summary, our contributions are as follows.

- We introduce *SGPT*, a novel approach that employs shared prompts to capture common information and utilizes group prompts to effectively align the global model with local distributions via a group selection module without local fine-tuning.
- We introduce a BCD optimization routine that iterates between learning the easy and common knowledge (through shared prompts) and then the more complex and specific knowledge (through group prompts). This way, we tackle optimization-specific challenges in the algorithm implementation.
- We theoretically bound the gap between global and local performance and identify two key factors: generalization and distribution discrepancy. *SGPT* can affect both these two factors, thus tightening the bound.
- We empirically test our algorithm on a wide range of datasets and types of heterogeneities. The results are compelling, demonstrating that *SGPT* consistently surpasses all baseline models.

2. Problem Setting and Preliminary

2.1. Problem Setting

In this paper, we examine a scenario involving M clients and a central server. The clients' data distributions are heterogeneous, characterized by either domain discrepancies or imbalanced class distributions. We denote the distribution for client i as $\mathcal{D}_i$ with $i \in [M]$ [3]. Each client i contains N_i data samples $\{(x_j^i, y_j^i)\}_{j=1}^{N_i}$. Further, let the parameter of a pre-trained ViT model be θ that are frozen during training. Denote trainable prompts as P, and classifier weights as W_C. We introduce the objective function of prompt-tuning the task model in FL:

$$\arg\min_{P, W_C} \sum_{i=1}^{M} \frac{N_i}{N} \sum_{j=1}^{N_i} l(\theta, P, W_C; x_j^i, y_j^i). \tag{1}$$

where $l : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ is the cross-entropy loss and N is the total number of data on all clients. In this way, we can leverage the representation power of ViT while enabling efficient tuning by only learning the prompts and classifier.

2.2. Visual Prompt Tuning (VPT)

Prompt Tuning is an efficient alternative to full fine-tuning for large-scale Transformer models. VPT [19] introduces only a small amount (less than 1% of model parameters)

[2]A detailed related work review on FL is provided in the appendix.

[3][m]={0, 1, . . . , m}

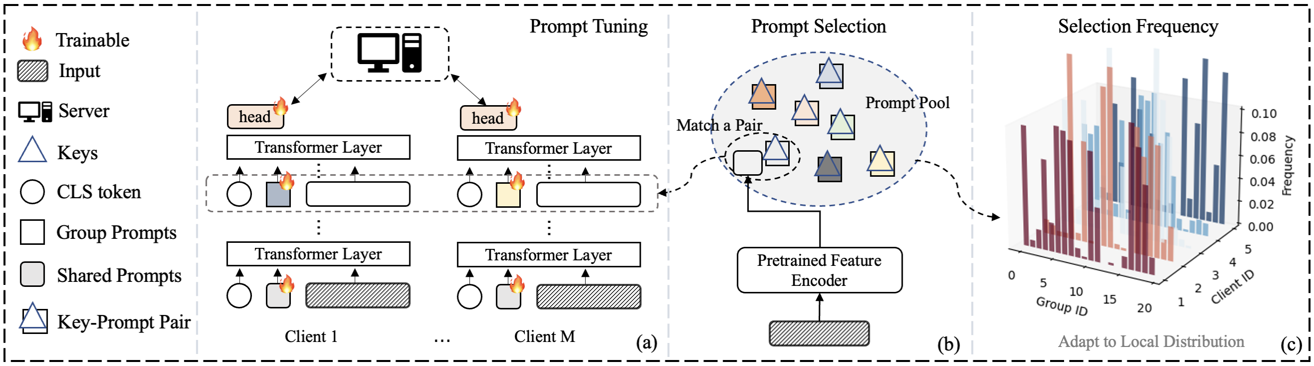


Figure 2. Pipleline of our method. (a) Provides an overview of the federated group-aware prompt-tuning *SGPT* procedure. Each model comprises shared prompts and group prompts, facilitating the acquisition of both common and group-specific knowledge. The shared prompts and classification head are globally trained, while the group prompt is inserted into intermediate layers trained within its respective data group and shared globally. (b) Depicts the prompt selection module. Here, each input undergoes processing by a pre-trained ViT model encoder. Similarities between keys and last layer CLS token features are calculated, and the prompt corresponding to the most similar key is selected for training, enabling group-aware training at the sample level. (c) Given that data distributions vary across clients, the frequencies of selected group prompts differ, ensuring our model aligns with various local data distributions.

of trainable parameters in the input space while keeping the backbone model frozen. Depending on the number of Transformer layers involved, [19] proposed VPT and VPT-D for efficient fine-tuning. To be specific, for VPT and VPT-D, prompts are inserted into the first Transformer layer and all layers respectively. Take VPT as an example, the prompt token is a learnable d-dimensional vector. The learnable prompt P is trained as follows:

$$\begin{aligned}
[cls_1, Z_1, E_1] &= L_1\left([cls_0, P, E_0]\right) \\
[cls_u, Z_u, E_u] &= L_u\left([cls_{u-1}, Z_{u-1}, E_{u-1}]\right) \;\; u = 2, \ldots, U \\
y &= \mathrm{Head}\left(x_U\right),
\end{aligned}$$

where U is the number of layers; Z_u represents the prompt features computed by the u-th Transformer layer L_u; E_u is a collection of image patch embeddings as the inputs to the $(u+1)$-th Transformer layer L_{u+1}; and $cls_u \in \mathbb{R}^d$ denotes the classification embeddings at L_{u+1}'s input space.

3. Method

3.1. Architecture

In this section, we follow VPT [19] to utilize a pre-trained ViT and adapt it to new tasks by prompt-tuning. We further advanced the architecture so that it can align to different data distributions automatically. Specifically, our approach involves learning shared prompts for common features while employing specialized group prompts through a prompt selection module to align with various local client data distributions effectively. Following VPT (shown in Section 2.2), we initialize with a pre-trained ViT model comprising U layers but learn shared and group prompts (see Fig. 2 (a)). First, we define the selection function $\texttt{Select} : \mathcal{X} \rightarrow [G]$ (detailed in Section. 3.2) that efficiently groups data into G groups. Then we introduce the shared and group prompts:

Shared Prompts: The shared prompts P_S are designed to capture common representations. Recent studies [15, 39] have shown low-level representations can be shared across groups, and distilling commonly used information into shared prompts can enhance the model's generalization. Motivated by the observation that features from different classes, processed by early layers of a pre-trained ViT, are uniformly distributed on the manifold (shown in Fig. 5 in Appendix), indicating shared information across classes. Therefore, we attach shared prompts P_S to the embedding features of the early layers, *i.e.* the first layer:

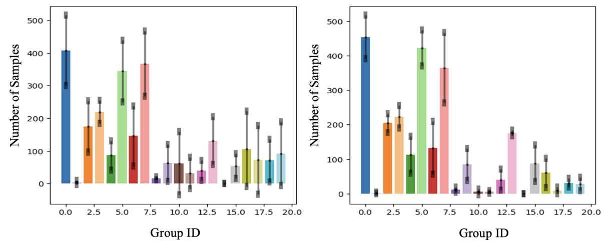
$$[cls_1, Z_1^S, E_1] = L_1\left([cls_0, P_S, E_0]\right). \tag{2}$$

Group Prompts: The group prompts set $P_G = \{p_1, \ldots, p_G\}$ containing G group prompt $p_g, g \in [G]$, which is designed to extract specialized information. In contrast to the early layers, the diverse and specialized features have shown to be preserved in higher layers [42]. Thus, we use the $\texttt{Select}$ function to assign group membership $g = \texttt{Select}(x)$ to a sample x (detailed in Eq. 5) and insert corresponding group prompts $p_g \in P_G$ to higher layers (*i.e.* the u-th layer) to extract task-specific features [42]:

$$\left[cls_u, z_u^g, Z_u^S, E_u\right] = L_u\left(\left[cls_{u-1}, p_g, Z_{u-1}^S, E_{u-1}\right]\right), \tag{3}$$

where Z_u^S represents the shared prompt features, z_u^g represents the group prompt feature computed by the u-th Transformer layer L_u. At last, we rewrite the objective function Eq. (1) into:

$$\underset{P_G, P_S, W_C}{\arg\min} \sum_{i=1}^{M} \frac{N_i}{N} \sum_{j=1}^{N_i} l(\theta, p_g, P_S, W_C; x_j^i, y_j^i), \tag{4}$$



(a) (b)

Figure 3. Stability analysis of one example client on CIFAR-100 dataset with $s = 10$. We plot the mean and standard deviation of the prompt selection number overall communication rounds. (a) Without stability regularization, the variance is larger and is unstable. (b) With our proposed momentum updating, the variance is reduced and is more stable.

where $p_g \in P_G$ are the selected group prompts that aligned with the predicted group membership of input x from the function `Select`.

3.2. Learning Prompt Selection Function

In this section, we introduce the prompt selection module `Select` that learns data-specific keys for picking group prompts. These keys are trained to capture the similarity of feature representations within their respective groups. Here, we propose a simple yet effective similarity-based clustering approach: we learn a key k_g for each group $g \in [G]$ as a centroid [54] and cluster data to its nearest centroid. Specifically, we first process the input sample x using the pre-trained model h_θ to obtain its generalized feature representation [49]. Subsequently, the keys $K = \{k_1, ..., k_G\}$ then cluster feature representations into groups based on cosine similarity:

$$\texttt{Select}(x) = \arg\max_{g \in [G]} cos(h_\theta(x), k_g). \tag{5}$$

However, training keys K in the FL setting faces several challenges: 1) collapse, leading to data clustering in few groups [34] and 2) instability due to heterogeneous client data causing inconsistent clustering (as shown in Fig 3 (a)).

To avoid collapse, we calibrate the `Select` function in Eq. 5 by weighting with the accumulated selection probability q_g during training. Specifically, let $v_g^t = \sum_{t'=1}^{t} \sum_{i=1}^{M} v_{g,i}^{t'}$ as the total number of times group g is selected up to communication round t across all clients i. The selection probability at communication round t is then calculated as $q_g^t = \frac{v_g^t}{\sum_{g \in [G]} v_g^t}$, we drop the communication round t to lighten notation in the later part. Then, we calculate the following loss function:

$$L_{key} = -cos(h_\theta(x), k_g) \tag{6}$$
$$\text{where } g \in \arg\max_{g \in [G]} \big(\cos\big(h_\theta(x), k_g\big) - 1\big) \cdot q_g,$$

To enforce the stability of clustering, we perform momentum parameter aggregation [16, 48] on the server side for both keys and group prompts to ensure selection consistency and knowledge consistency, respectively. Denoting the aggregated parameters for a group g's key and prompts at round t as k_g^t and p_g^t, and the momentum parameters as $\hat{k}_g^t$ and $\hat{p}_g^t$ (see Algorithm 2 in appendix), the parameters are updated as follows:

$$\begin{aligned} \hat{k}_g^t &= \alpha_k \hat{k}_g^{t-1} + (1 - \alpha_k) k_g^t, \\ \hat{p}_g^t &= \alpha_g \hat{p}_g^{t-1} + (1 - \alpha_g) p_g^t, \quad g \in [G], \end{aligned} \tag{7}$$

where α_k and α_g are the momentum rates (0.5 in our case).

3.3. Block Coordinate Descent for Optimization

Because we use `Select` function, a non-continuous function, for selecting group prompts, this renders the objective function in Eq. (4) to be non-smooth, further introducing optimization challenges. To effectively navigate the challenges, we employ a Block Coordinate Descent (BCD) method for the local training of Eq. (4). The BCD optimization involves dividing the parameters into sub-groups and optimizing them in an iterative manner. In our case, this means optimizing the shared prompts and group prompts separately. Furthermore, the order of updating each parameter sub-group is critical for achieving good performance [45]. Inspired by cognitive development theories on human learning progression [11], our method first learns the easy, common feature information (shared prompts) and then optimize the group prompts to extract more specialized knowledge.[4] In this way, we formulate the local objective functions for client i as:

$$\arg\min_{P_G, W_C} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[l(p_{\texttt{Select}(x)}, P_S^\star, W_C; x, y)], \tag{8}$$
$$st. P_S^\star, W_C \in \arg\min_{P_S, W_C} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[l(P_S, W_C; x, y)]. \tag{9}$$

We first find the optimal share prompts P_S independently to learn common information. Then, the group prompts P_G will residually learn group-specific knowledge upon the optimal shared prompts.

Block I: Learning Shared Prompts. The shared prompts are learned independently (lines 1-5 in Algorithm 1) to capture common information with the following loss function:

$$L_{share} = l(P_S, W_C; x, y) \tag{10}$$

where l is the cross-entropy loss.

Block II: Learning Group Prompts and Keys: Having obtained the shared prompts, the group prompts are learned (lines 6-11 in Algorithm 1) to extract group-specific knowledge. Initially, the shared prompts P_S are inserted but frozen. Then, we employ `Select` function to determine x's group

[4]To demonstrate the effectiveness of this ordered approach, we present an ablation study on the update sequences in Table 3.

Algorithm 1 Block Coordinate Descent on Local Client

Input: Weights $W = \{W_C, P_G, P_S\}$, pre-trained ViT h_θ, prompt selection module `Select` with learnable keys $K = \{k_g\}_{g=1}^G$, training data $(x, y) \sim \mathcal{D}$, learning rate η, local training steps E in one communication round.

```
Block I: Learn Shared Prompts Only:
for e = 1 → E do
    Optimize the Eq. (8)
    {P*_S, W_C} ← {P_S, W_C} − η · ∇l(P_S, W_C; x, y)
end for
Block II: Learn Group Prompts with frozen P*_S :
for e = 1 → E do
    p_g, k_g ← PromptSelection((x))
    {p_g, W_C} ← {p_g, W_C} − η · ∇l(p_g, P*_S, W_C; x, y)    ▷ Eq. (9)
    K ← K − η · ∇l_key(k_g; x, y)                              ▷ Eq. (6)
end for
procedure PromptSelection(Select, x)
    p_g ← P_Select(x)                                         ▷ Select group prompt
    k_g ← K_Select(x)                                         ▷ Select group key
end procedure
```

$g = \texttt{Select}(x)$ (detailed in Eq. (5)) and insert corresponding group prompts p_g to extract group-specific features [42]. The feature representation derived from these prompts is combined with the cls token via average pooling for the final classification. Consequently, the group prompts effectively learn group-specific knowledge. Additionally, The keys K in `Select` function are learned simultaneously, and the total loss function is:

$$L_{total} = L_{key} + l(P_G, W_C; x, y), \tag{11}$$

where l is cross-entropy and L_{key} is the loss for learning `Select` function. We perform Algorithm 1 to optimize the parameters over the global communication rounds.

3.4. Efficient Inference

In this section, we explain the inference procedure of *SGPT*. Given a sample x, we first use the `Select` function (see Eq. (5)) to determine its group membership $g = \texttt{Select}(x)$. Then, the shared prompt P_S and corresponding group prompt p_g are inserted into the model to achieve sample-level adaptation for inference. When performing tests on new clients, the frequencies of selected group prompts can be automatically adjusted by `Select` function (shown in Fig. 2 (c)), ensuring our model aligns with their local data distributions. We provide more training and inference details of our proposed *SGPT* in the Appendix.

4. Theoretical Analysis

In this section, we provide analytical justification for narrowing the empirical risk of the global model found by empirical loss minimization and the population risk of the optimal model of a client. Following the heterogeneity setting in [36], we assume each local client i's data distribution $\mathcal{D}_i$ is a mixture of G underlying distributions (groups).

Assumption 1. *On a client i, there exist G underlying (independent) distributions $\mathcal{D}_g^i$, $g \in [G]$, such that for $i \in [M]$, $\mathcal{D}_i$ is mixture of the distributions $\mathcal{D}_g^i$ with mixing probability vector $\pi_i = [\pi_1^i, \dots, \pi_G^i]$:*

$$\mathcal{D}_i = \sum_{g \in [G]} \pi_g^i \mathcal{D}_g^i, \quad \sum_g \pi_g^i = 1 \quad and \quad \pi_g^i \geq 0, \tag{12}$$

where $\pi_g^i = \frac{N_g^i}{N_i}$ is the probability of a data sample on client i belong to group g and N_g^i is a fixed number of samples from $\mathcal{D}_g^i$ for all $g \in [G], i \in [M]$.

Based on this, we also introduce the probability distribution $\mathcal{C}_g$ of data belonging to group g as follows. For a group g, its global data distribution $\mathcal{C}_g$ is a mixture of the distributions $\mathcal{D}_g^i, i \in [M]$ with mixing probability vector $\pi_g = [\pi_g^1, \dots, \pi_g^M]$:

$$\mathcal{C}_g = \sum_{i \in [M]} \pi_g^i \mathcal{D}_g^i. \tag{13}$$

Following [18], we refer to the distribution $\mathcal{C}_g$ as the "participated clients' data distribution" for the g-th group.

Finally, for $g \in [G]$ and hypothesis $h_g \in \mathcal{H}$, we use $h_{\texttt{Select}} = \{h_1, ..., h_g, ..., h_G\}_{\texttt{Select}(x)}$ to denote the group-aware hypothesis determined by function `Select` when datapoint x is given as input. Let $\widehat{h}_g = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{\widehat{\mathcal{C}}_g}(h)$ denote the empirical model for data group g and $\hat{h}_{\texttt{Select}} = \{\hat{h}_1, ..., \hat{h}_g, ..., \hat{h}_G\}_{\texttt{Select}(x)}$ denote the corresponding global model.

Theorem 4.1 (Gap between the global and local performance)**.** *Assume the loss function ℓ is bounded in $[0, 1]$ and the function `Select` is a data grouping method. Let the VC-dimension of hypothesis class $\mathcal{H}$ be d. Then, with a probability of at least $1 - \delta$ over the training set,*

$$\begin{aligned} \mathcal{L}_{\widehat{\mathcal{D}}_i}(\widehat{h}_{\texttt{Select}}) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) \leq & \\ \sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + & \underbrace{2 \sum_{i=1}^{G} \frac{N_g^i}{N_i} \sqrt{\frac{2d}{N_g} \left(1 + \log(\frac{N_g}{d})\right)}}_{Generalization} \\ + & \underbrace{\sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + \mathrm{disc}(\widehat{\mathcal{D}}_g^i, \widehat{\mathcal{C}}_g)\right)}_{Distribution\ Discrepancy}, \end{aligned} \tag{14}$$

where $\mathrm{disc}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{D}_1}(h) - \mathcal{L}_{\mathcal{D}_2}(h)|$ and N_g is the tocal number of data in group g from all the clients.

The detailed proof is provided in the Appendix. The left-hand side of Eq. (14) represents the gap between the minimum empirical risk[5] of the global model found by empirical loss minimization using the `Select` grouping function and the population loss of the optimal model of client i. The right-hand side of Eq. (14) bounds this gap with respect to weighted averages of two factors: 1) **Generalization** (GE) that is related to N_g, and 2) **Distribution Discrepancy** (DD) between the global group distribution $\mathcal{C}_g$ and the local group distribution $\mathcal{D}_g^i$ of client i. *SGPT* accounts for these two terms and reduces the gap with shared and group prompts. Specifically, recent studies [15, 39] have shown that low-level representations exhibit considerable similarity across different groups, suggesting a relatively small DD term. In response to these findings, our approach involves inserting shared prompts P_S in early (low-level) layers of ViT and training using all data, thereby maximizing the value of N_g ($g = 1, N_g = N$) to effectively reduce the dominant GE term. At higher layers, where the DD dominates the bound due to diverse feature representations [59], a selection module groups similar data to learn the same group prompt, ensuring $\mathcal{D}_g^i \approx \mathcal{C}_g^i$ thus reducing DD. For a detailed discussion, see Appendix C.4.

5. Experiments

5.1. Experiment Setup

Datasets. In this section, we introduce datasets with label and feature heterogeneity. *Label heterogeneity*: we demonstrate the effectiveness of our proposed approach for *label heterogeneity* using two datasets. (1) *CIFAR100 dataset* comprises 50,000 training images and 10,000 testing images distributed across 100 classes. (2) *Fivedataset* consists of a sequence of 5 datasets (SVHN, CIFAR10, not-MNIST, Fashion-MNIST, and MNIST) as outlined in the work by [10]. *Feature heterogeneity*: we also consider feature heterogeneity and follow [56] to demonstrate the effectiveness of our proposed approach using Office-Caltech10 and DomainNet for *feature heterogeneity*: (1) Office-Caltech10 [43] is composed of four data domains, including Amazon, DSLR, Webcam, and Caltech. Each domain contains ten classes, with 2,533 images in total. (2) DomainNet [40] consists of 0.6 million images of 345 classes distributed across six domains: clipart, infograph, painting, quickdraw, real, and sketch. Following [31, 56], we use the top ten most frequent classes to form a sub-dataset for our experiments.

Non-IID Settings. In this section, we introduce the FL environments and the data partition strategies for various datasets. For clients with *label heterogeneity*, in CIFAR-100, we introduce 100 clients and set a low (hard) client participating ratio (γ) to 0.05. To introduce data heterogeneity among clients, we apply the "Pathological Partition" [28, 38]. We first sort the data by labels and then allocate data from a specific number of classes (s) to each client. Since s is the number of classes each user can have, as s decreases, the degree of data heterogeneity increases. As to the Five Datasets, we distribute the data among 20 clients, with every 4 clients originating from the same dataset. We set the participating rate to $\gamma = 0.1$ and perform training for 50 communication rounds. For conducting clients with *feature heterogeneity*, we follow the newest benchmark [56] and assign a data domain to a client, indicating the number of clients (M) is set as 4 and 6 for Office-Caltech10 and DomainNet, respectively.

Implementation Details. Following [13, 56], we use ImageNet-21K pre-trained ViT-B-16 [9] as our model because it achieves a good trade-off between performance and efficiency. Since ViT-B-16 is originally trained on images with size 224 and patch size 16, we resize our images to 224 to align with the model's specifications. During the prompt-tuning process, we focus on the shared prompts, group prompts, and the classifier. We use local training epochs (E = 5) for all experiments and set the prompt length to 1 for efficiency. For *label heterogeneity* datasets, we use the last layer's cls token as the input feature for `Select` and set group number G as 20 and 5 for CIFAR-100 and Five-dataset respectively. For *feature heterogeneity*, we use the intermediate (5-th) layer's cls token as the input feature for `Select` because intermediate layers capture the texture-related information [15] and set G as 4 and 6 for Office and DomainNet datasets respectively. We follow other settings in [56] for consistency.

Baseline Methods. For *label heterogeneity*, we conduct a comparative analysis of our method against various global-model approaches: VPT [19] that optimized using FedAvg [37] (FedVPT), Head-Tuning [47], FedMix [58], as well as recent FedPR [13]. Additionally, for personalized Federated Learning, we considered pFedPG [44, 56] and FedEM [36]. For *feature heterogeneity*, we implement the same baseline methods as [56] as it is the newest benchmark for FL feature heterogeneity.

5.2. Results

5.2.1 Label Heterogeneity Results

We calculate three metrics for evaluation: (1) *Global Accuracy* represents the mean accuracy made by each client overall all testing images regarded as global distribution performance (2) *Local Accuracy* is calculated by averaging the accuracy of each local client on its local test data. (3) *Worst Local Accuracy* demonstrates the worst-performing client result, showcasing the ability to adapt to local data distribution. We follow [28] to report the averaged testing accuracy for the last 10 consecutive global communication rounds.

Overall Performance The results of CIFAR-100 and Five-datasets are presented in Table 1. Due to the difficulty of

[5]We also give the gap on population distribution in the Appendix.

Table 1. Performance comparison of different methods on the CIFAR-100 dataset and the Five dataset. The **bold** and underline highlights represent the best and second-best results, respectively.

Datasets	CIFAR-100 (%) ↑						Five-Dataset (%) ↑		
Method	Global		Local		Worst Local		Global	Local	Worst Local
	$s=50$	$s=10$	$s=50$	$s=10$	$s=50$	$s=10$			
Head-Tune	76.69	75.35	76.68	75.36	65.96	55.53	75.09	75.09	28.60
FedVPT [19]	82.35	80.79	85.67	80.79	72.28	66.43	80.88	80.88	53.04
FedVPT-D [19]	85.85	79.49	85.85	79.49	74.83	63.38	82.09	82.09	54.96
FedMix [58]	85.65	80.83	85.67	80.82	74.33	69.59	81.07	81.07	42.62
pFedPG [56]	84.67	81.19	85.22	81.82	72.33	70.53	72.12	82.48	56.24
FedEM [36]	81.52	78.41	81.53	78.40	72.06	63.87	79.54	79.54	46.50
FedPR [13]	85.92	81.42	85.92	81.35	75.26	63.93	81.29	81.29	42.12
SGPT (Ours)	**86.72**	**84.64**	**86.71**	**84.64**	**77.56**	**73.85**	**83.40**	**83.40**	**61.04**

Table 2. Performance comparison of different methods on the Office-Caltech10 and DomainNet datasets. The **bold** and underline highlights represent the best and second-best results respectively.

Datasets	Office-Caltech10 (%) ↑					DomainNet (%) ↑						
Method	A	C	D	W	Avg.	C	I	P	Q	R	S	Avg.
Per-FedAvg [12]	91.67	90.22	100.0	100.0	95.47	69.39	48.71	82.07	35.30	90.63	72.56	66.44
FedRep [6]	91.15	88.44	100.0	100.0	94.90	64.26	38.20	72.86	62.10	82.66	60.11	63.37
FedVPT [19]	92.71	84.44	100.0	100.0	94.29	65.59	44.14	76.58	47.30	91.04	60.29	64.16
FedVPT-D [19]	91.67	89.33	100.0	100.0	95.25	63.31	43.07	74.80	54.80	87.26	67.15	65.07
pFedPG [56]	94.79	92.44	100.0	100.0	96.81	73.00	50.08	84.33	60.00	94.00	68.41	71.64
FedPR [13]	95.31	95.11	100.0	96.61	96.76	88.02	49.16	86.11	70.00	96.06	83.94	78.88
SGPT (Ours)	**95.31**	**95.56**	**100.0**	**100.0**	**97.72**	**89.54**	**52.82**	**87.56**	**70.00**	**96.14**	**86.82**	**80.48**

datasets, directly leveraging the pre-trained ViT and applying head-tuning cannot achieve good performance. With the help of prompt-tuning, FedVPT serves as a strong baseline compared to recent personalized FL algorithms. This can be attributed to the effectiveness of transformers in mitigating catastrophic forgetting and accelerating convergence in dealing with heterogeneous data [41], which validates our motivation to apply ViT in FL. With the help of our proposed *SGPT*, we outperform all baselines on both global and worst-local test accuracy across various degrees of data heterogeneity and datasets by a significant margin.

Different Label Heterogeneity Level. As s decreases, this represents an increase in label heterogeneity. The CIFAR-100 results in Table 1 show a performance drop across all methods as heterogeneity increases with the decrease of s from 50 to 10. Fortunately, our *SGPT* demonstrates robustness to data heterogeneity, with a smaller performance drop compared with other methods.

5.2.2 Feature Heterogeneity Results.

In addition to label heterogeneity, our method can also be applied to datasets with feature shifts. We follow [56] and report both the performance for each client and the average performance overall for clients. The results of Office-Caltech10 and DomainNet datasets with the presence of domain shifts across clients are presented in Table 2. Among all baselines, our *SGPT* achieves the highest average accuracies on Office-Caltech10 and DomainNet at 97.72% and 80.48% respectively. In addition, *SGPT* can also align the global model with different local client data with the worst local performance of each dataset at 95.31% and 52.82% respectively. The results validate our goal to reduce the performance gap between global and local data distribution.

5.3. Global and Local Performance Trade-off

We demonstrate our method's capacity to achieve high test accuracy on both global data distribution and local clients' data distributions. As shown in Figure 1, we visualize both the global accuracy and the worst local accuracy (client with the worst test accuracy) for the CIFAR-100 dataset with $s = 10$. Points situated in the top-right corner indicate better performance in both global data distribution and local clients' data distribution. PFL methods that are finetuned locally tend to overfit on local data distribution. GFL methods, though achieving good generalization, often underperform in local data. Notably, our method shows the greatest capacity for aligning with both global and local data distributions.

5.3.1 Analysis and Ablation Study

In this section, we provide a detailed analysis of each module in our methods. For more analysis, please see the Appendix.

Effect of Shared Prompt and Group Prompt. To demonstrate the benefits of reducing the terms related to GE and DD as outlined in Theorem 4.1, we conduct ablation studies on shared and group prompts. As indicated in Table 3, replacing

Table 3. Effect of different prompts and block coordinate descent. We report the results on the CIFAR-100 dataset with $s = 10$.

Share	Group	BCD-Inv	BCD	Global ↑	Worst Local ↑
√				79.49	63.38
	√			82.51	70.79
√	√			77.82	62.62
√	√	√		76.76	62.15
√	√		√	**84.64**	**73.85**

shared prompts with group prompts, to reduce distribution discrepancy, resulting in a 7% gain in the worst local accuracy. This implies successfully aligning the global model with local data distributions, thereby validating our theoretical motivation. Additionally, the group prompts improve global accuracy by around 3% because similar data lead to lower gradient dissimilarity [26] and benefit the optimization process [21]. By additionally adding shared prompts and optimizing prompts with BCD, the global and worst local performance improve by around 2% and 3% respectively. As a result, learning common information with shared prompts benefits the generalization.

Effect of Block Coordinate Descent. In this section, we analyze the effect of our proposed BCD optimization (Section 3.3) on the CIFAR-100 dataset, specifically with $s = 10$. As shown in Table 3, the direct addition of group prompts leads to a performance decrease by nearly 3%. With the aid of our BCD optimization that iteratively learns shared prompts first and then groups prompts, a significant improvement by 7% is observed. When inverting the BCD order (denoted as BCD-Inv), the result drops significantly. These results validate our BCD optimization approach.

Ablation on Clustering Performance. We conduct ablation studies on various improvements in learning `Select` function. We evaluate the clustering accuracy using the CIFAR-100 test dataset, with coarse labels [24] serving as the ground truth. As shown in Table 4, the direct application of FedAvg to learn keys results in clustering all data into the same group, resulting in only 5% accuracy. By adding selection probability q_g, `Select` function successfully learns meaningful keys, as shown in Eq. (6). Introducing momentum update, as shown in Eq. (7), further enhances the clustering performance, achieving approximately 10% and 5% improvements, respectively. Additionally, in Fig 3, we plot the mean and standard deviation of the prompt selection numbers over all communication rounds. This demonstrates that our proposed momentum updating improved stability.

Where to attach prompts? Here we take CIFAR-100 with $s = 10$ as an example and use a heuristic search strategy: (1) We start with examining the position for shared prompts by adding them to the first U_S layers. Our findings, shown in Fig 4, inserting shared prompts beyond the 3-rd layer decreases performance. This is because higher layers suffer more from heterogeneity [59]. As a result, $U_S = 3$. (2)

Table 4. Ablation studies with different improvements on `Select`. We report the results on the CIFAR-100 dataset with $s = 10$.

Dataset	CIFAR-100 (%)	
Method	$s = 50$	$s = 10$
FedAvg	5.00	5.00
w/ q_g	46.09	51.26
w/ momentum and q_g	56.39	56.62

Table 5. The size of learnable parameters compared with that of vanilla FedAvg [37], which trains the whole models.

Method	FedAvg			Ours
Architecture	ResNet-18	ResNet-50	ViT-16/B	ViT-16/B
# Parameters ↓	$11M$	$24M$	$86M$	**0.1M**

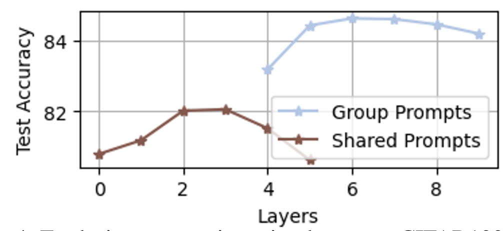


Figure 4. Exploring prompt insertion layers on CIFAR100 (s=10). The brown curve represents performance using only shared prompts, while the blue curve illustrates performance with group prompts inserted at varying layers, alongside shared prompts in layer 3.

Based on the best position for shared prompts, we study the position of group prompts. As depicted in Fig 4, adding prompts from 4-th to 6-th yields the optimal performance of 84.64%. In summary, without specific designs (*i.e.*, our algorithm), training prompts on higher layers prove challenging due to increased heterogeneity. We suggest that prompt tuning in FL should take this aspect into consideration. **Efficiency.** In this section, we conduct a comparison between the number of parameters that need training and communication between *SGPT* and training a whole network in classical FL (*e.g.*, FedAvg [37]). As shown in Table 5, prompt-tuning requires significantly less trainable parameters compared with traditional FL thus improving the efficiency and saving the communication cost.

6. Conclusion

This work demonstrates the significant advancements in FL through the integration of ViT to bridge GFL and PFL. This is achieved through our proposed *SGPT*, which introduces a shared and group prompt tuning strategy, enabling the model to adeptly capture both common and group-specific features. The prompt selection module of *SGPT* facilitates the training of a global model that can automatically adapt to varied local client data distributions without necessitating local fine-tuning. We employ BCD for effective optimization of the learnable parameters. Theoretically, our approach minimizes the error bound between global and local performances. Empirical tests and ablation studies highlight *SGPT*'s superior performance and efficiency.

References

[1] Wenxuan Bao, Haohan Wang, Jun Wu, and Jingrui He. Optimizing the collaboration structure in cross-silo federated learning. *arXiv preprint arXiv:2306.06508*, 2023. 18

[2] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10231–10241, 2021. 1

[3] 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. 1

[4] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2749–2758, 2021. 2, 12

[5] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020. 12

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021. 7

[7] Wenlong Deng, Yuan Zhong, Qi Dou, and Xiaoxiao Li. On fairness of medical image classification with multiple sensitive attributes via learning orthogonal representations. In *International Conference on Information Processing in Medical Imaging*, pages 158–169. Springer, 2023. 12

[8] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 12

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 6

[10] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 386–402. Springer, 2020. 6

[11] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1): 71–99, 1993. 4

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020. 2, 7

[13] Chun-Mei Feng, Bangjun Li, Xinxing Xu, Yong Liu, Huazhu Fu, and Wangmeng Zuo. Learning federated visual prompt in null space for mri reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8064–8073, 2023. 2, 6, 7, 12

[14] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. 2, 12

[15] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020. 3, 6

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 4

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[18] Xiaolin Hu, Shaojie Li, and Yong Liu. Generalization bounds for federated learning: Fast rates, unparticipating clients and unbounded losses. In *The Eleventh International Conference on Learning Representations*. 5, 14

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 1, 2, 3, 6, 7, 12

[20] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020. 12

[21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2, 8, 12

[22] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 1

[23] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multimodal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 12

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[25] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems,*

Security and Sustainability (WorldS4), pages 794–797. IEEE, 2020. 12

[26] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 8

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 12

[28] Hongxia Li, Zhongyi Cai, Jingya Wang, Jiangnan Tang, Weiping Ding, Chin-Teng Lin, and Ye Shi. Fedtp: Federated learning by transformer personalization. *arXiv preprint arXiv:2211.01572*, 2022. 2, 6, 12, 19

[29] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 2

[30] 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. 2, 12

[31] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021. 2, 6, 20

[32] Yaowei Li, Ruijie Quan, Linchao Zhu, and Yi Yang. Efficient multimodal fusion via interactive prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2023. 12

[33] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 12

[34] Louis Mahon and Thomas Lukasiewicz. Hard regularization to prevent collapse in online deep clustering without data augmentation. *arXiv preprint arXiv:2303.16521*, 2023. 4

[35] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. 2, 12

[36] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34:15434–15447, 2021. 5, 6, 7, 18

[37] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 6, 8, 12

[38] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021. 6, 12, 19

[39] Oleksiy Ostapenko, Timothee Lesort, Pau Rodriguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay. In *Conference on Lifelong Learning Agents*, pages 60–91. PMLR, 2022. 3, 6

[40] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 6

[41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10071, 2022. 7

[42] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 3, 5

[43] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010. 6

[44] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021. 2, 6, 12

[45] Hao-Jun Michael Shi, Shenyinying Tu, Yangyang Xu, and Wotao Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016. 4

[46] Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34:11220–11232, 2021. 12

[47] Guangyu Sun, Matias Mendieta, Taojiannan Yang, and Chen Chen. Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning. *arXiv preprint arXiv:2210.01708*, 2022. 1, 2, 6

[48] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 4

[49] Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33:7852–7862, 2020. 4

[50] Shusaku Tsumoto. Contingency matrix theory: Statistical dependence in a contingency table. *Information Sciences*, 179 (11):1615–1627, 2009. 19

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[52] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with

matched averaging. *arXiv preprint arXiv:2002.06440*, 2020. 12

[53] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022. 12

[54] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 4, 12

[55] Shanshan Wu, Tian Li, Zachary Charles, Yu Xiao, Ziyu Liu, Zheng Xu, and Virginia Smith. Motley: Benchmarking heterogeneity and personalization in federated learning. *arXiv preprint arXiv:2206.09262*, 2022. 2

[56] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19159–19168, 2023. 2, 6, 7, 12, 20

[57] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. Fedlora: Model-heterogeneous personalized federated learning with lora tuning. *arXiv preprint arXiv:2310.13283*, 2023. 2, 12

[58] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. *arXiv preprint arXiv:2107.00233*, 2021. 6, 7

[59] Yaodong Yu, Alexander Wei, Sai Praneeth Karimireddy, Yi Ma, and Michael Jordan. Tct: Convexifying federated learning using bootstrapped neural tangent kernels. *Advances in Neural Information Processing Systems*, 35:30882–30897, 2022. 6, 8

[60] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019. 12