Overall pipeline of Appendix. We first include related work in Section A. Then the notations and algorithm details of *SGPT* are explained in Section B. In Section C, we introduce the proof of Theorem 4.1 along with required lemmas. Finally, additional analysis are reported in Section D for a more comprehensive study.

A. Related Work

A.1. Generic Federated Learning

FedAvg [37] is a standard FL algorithm that involves multiple rounds of local training and global aggregation. Many works focus on improving FedAvg through various aspects: (1) global aggregation methods [5, 52, 60] replace the weighted average with more dedicated strategies like ensemble and distillation. (2) optimization methods [20, 21, 30] reduce the client drifts [21] by correcting local gradients [21] or employed regularization toward the global model [30] thus achieving a better convergence rate. Recently, FedPR [13], a prompt-tuning-based GFL method, learns client prompts in the null space of group prompts in the previous round and aggregates them into global prompts, but it may not perform well when the global prompts are not low-rank [7]. The disadvantage of GFL methods is they are insufficient [30] for good performance when dealing with significant data heterogeneity.

A.2. Personalized Federated Learning

PFL [25] learns a customized model for each client. To be specific, Fine-tuning methods [38, 46] adjust a global or meta-trained model to adapt to local client data by fine-tuning part of the global model. Clustered FL [4, 14, 35] clusters clients with similar data distribution, assuming that they can share the same optimal model, however, imposes a heavy computational burden. pFedHN [44] learns a hypernetwork at the server to aggregate clients' model updates and produce their entire models for the next round. The disadvantage of PFL is overcoming challenges in adapting to new clients and overfitting local data. In this paper, we learn a generalized FL model that not only achieves high accuracy on the global distribution but is also capable of aligning with different local distributions without local adaptation.

A.3. Parameter Efficient Tuning

Parameter efficient tuning (PET) [8], initially proposed for text models, enables easier access and usage of pre-trained models by reducing the memory cost needed to conduct fine-tuning due to fewer computed gradients. PETuning techniques, including methods like Adapter Tuning [28], LoRA [57], Prompt-Tuning [13], and Head-Tuning [56], freeze most parameters of pre-trained models and update only a few additional parameters or a part of the original model parameters for downstream tasks. This paper focuses on prompt tuning due to no need to modify anything inside the neural network [27, 33] and Visual Prompt Tuning (VPT) [19] has been established as an efficient and effective PETuning method for adapting large-scale ViT models to vision tasks. Recent studies on VPT have been conducted in fields like continual learning [53, 54] and multi-modality learning [23, 32]. Although these advancements have shown progress in various visual tasks, prompt tuning remains predominantly limited to centralized systems. The effectiveness of prompt tuning in a distributed framework has yet to be thoroughly investigated.

B. Algorithm details

B.1. Notation

For convenience, we summarize the notations used in this paper in Table 6.

B.2. Training Algorithm

We present more details in the training phase of *SGPT*. We provide a detailed illustration of the training process for our proposed method in Algorithm 2 based on the commonly used FedAvg [37] scheme: during each communication round, the clients engage in local training using the global model received from the server, and the server aggregates the shared parameters from the clients to update the global model. Notably, *SGPT* can be combined with other FL methods. Particularly, in *SGPT*'s training, the local model parameters (*i.e.*, P_S, W_C, K, and P_G) are sent back to the server for aggregation. For model parameters, the aggregation weight of client i is determined by $\alpha_i = N_i/N$. Regarding the aggregation of keys, we initially calculate the selection quantity of a group g on client i at round t denoted as $N_g^{i,t}$. Then, the aggregation weight for a group key is computed as $N_g^{i,t}/\sum_i N_g^{i,t}$. At last, the momentum aggregation is applied to the keys and group prompts.

Table 6. Main notations used in this paper.

		Basic Variables
M	$\triangleq$	Number of clients
$\mathcal{X}$	$\triangleq$	Input space
$\mathcal{Y}$	$\triangleq$	Label space
$\mathcal{D}_i$	$\triangleq$	Data distribution on client i and $\mathcal{D}_i$ is on $\mathcal{X} \times \mathcal{Y}$
$N = \sum_{i=1}^{M} N_i$	$\triangleq$	N_i is number of data samples at client i and N is the number of data on all clients
$\{x_i, y_i\} \sim \mathcal{D}_i$	$\triangleq$	Data sample (x_i, y_i) located on participating client i is made of i.i.d sampling from $\mathcal{D}_i$
		Function Variables
ℓ	$\triangleq$	loss function
`Select`	$\triangleq$	Prompt Selection function
$[G]$, G	$\triangleq$	Group set, number of groups G
$\mathcal{D}_g^i$	$\triangleq$	Data distribution on client i from group g
N_g^i	$\triangleq$	Number of data samples at client i from group g
$N_g = \sum_{i=1}^{M} N_g^i$	$\triangleq$	Number of data samples from group g
K, k_g	$\triangleq$	Keys set, key of g-th group $\|\mathcal{G}\|$
P_G	$\triangleq$	Weight of group prompts
P_S	$\triangleq$	Weight of shared prompts
W_C	$\triangleq$	Weight of classifier
h_θ	$\triangleq$	Pretrained foundation model
cls	$\triangleq$	Classification token
E	$\triangleq$	Image patch embeddings
Z	$\triangleq$	Prompt features embeddings

B.3. Inference Algorithm

We present more detials in the inference phase of *SGPT*. The inference procedure of *SGPT* in Algorithm 3. 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 prompts 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 automatically adjust by `Select` function (shown in Fig. 2 (c)), ensuring our model aligns with their local data distributions.

C. Technical details and Omitted proofs

C.1. Settings and Definitions

First, we formally set up some general notation: For a distribution $\mathcal{D}$ with support $(\mathcal{X}, \mathcal{Y})$ and a non-negative loss function $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ denote the population risk of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ as follows:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(h(X), Y)]\,.$$

Let $\mathcal{H}$ represent a hypothesis class and denote the hypothesis $\hat{h}$ minimizing the empirical risk as

$$\widehat{h} = \arg\min_{h\in\mathcal{H}} \mathcal{L}_{\widehat{\mathcal{D}}}(h),$$

where we denote $\hat{\mathcal{D}}$ the empirical distribution of samples drawn iid from $\mathcal{D}$. We will also denote $\mathfrak{R}_{\mathcal{D},n}(\mathcal{H})$ the Rademacher complexity of the hypothesis class $\mathcal{H}$ over the distribution $\mathcal{D}$ with n samples. Furthermore, we define the distribution mismatch between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ as

$$\operatorname{disc}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \max_{h\in\mathcal{H}} \left|\mathcal{L}_{\mathcal{D}_1}(h) - \mathcal{L}_{\mathcal{D}_2}(h)\right|. \tag{15}$$

Algorithm 2 *SGPT* Training Algorithm

Server Input: Initial weights $W = \{W_C, P_G, P_S\}$, prompt selection module `Select` with learnable keys $K = \{k_g\}_{g=1}^{G}$, number of participating clients in each round $m = \gamma \times M$, number of communication rounds T, client data ratio set $\{\alpha_i\}_{i=1}^{M}$, accumulated selection quantity $\{v_g^i\}_{i=1}^{M}, g \in [G]$.

Client i's Input: Pre-trained Transformer h_θ, training data $(x_i, y_i) \sim D_i$ for client i, learning rate η, number of local training steps E.

```
For $t = 1 \to T$ rounds, sample $m$ clients and execute **Procedure A** and **Procedure B** iteratively.
**procedure A**: ClientUpdate($i$)
    $W_i \leftarrow W$                                              ▷ Initialize with global model
    $K_i \leftarrow K$                                              ▷ Initialize with global keys
    Apply Block Coordinate Descent (Algorithm 1) to update the parameter $W_i$ and $K_i$.
    Count the group selection quantity $\{N_g^i\}_{g \in G}$
    Send updated $W_i$, $K_i$ and $\{N_g^i\}_{g \in G}$ to ServerExecute
**end procedure**
**procedure B**: ServerExecute($t$)
    Receive local models' parameters from ClientUpdate
    $[W_C^t, P_S^t, P_G^t] \leftarrow \sum_{i=1}^{m} \alpha_i [W_C^{i,t}, P_S^{i,t}, P_G^{i,t}]$      ▷ Parameter Aggregation
    **for** $g = 1 \to G$ **do**
        $k_g^t \leftarrow \sum_{i=1}^{m} \frac{N_g^{i,t}}{\sum_{i \in [m]} N_g^{i,t}} k_g^{i,t}$      ▷ Key Aggregation
        $v_g^t = v_g^{t-1} + \sum_{i=1}^{m} N_g^{i,t}$                 ▷ Accumulate Group Number
    **end for**
    Apply Momentum Aggregation Eq. 7.
    $(\hat{k}_g^0, \hat{p}_g^0 = k_g^0, k_g^0$ if $t = 0)$
    $\hat{k}_g^t = \alpha_k \hat{k}_g^{t-1} + (1 - \alpha_k) k_g^t$
    $\hat{p}_g^t = \alpha_g \hat{p}_g^{t-1} + (1 - \alpha_g) p_g^t, \quad g \in [G]$
    broadcast parameters to ClientUpdate
**end procedure**
```

Next, we formally introduce the statistical setting of our analytical investigation of the impact of group-aware hypothesis inference in a multi-client setting. For clients $i \in [M]$, let $\mathcal{D}_i$ denote their corresponding distributions of data pairs (X, Y). We assume that each local distribution $\mathcal{D}_i$ is a mixture of group-specific distributions $\mathcal{D}_g^i, g \in [G]$ for G. Concretely,

$$\mathcal{D}_i = \sum_{g \in [G]} \pi_g^i \mathcal{D}_g^i, \tag{16}$$

with mixing probability vector $[\pi_1^i, \ldots, \pi_G^i]$. This also induces a probability distribution $\mathcal{C}_g$ of data belonging to group g as follows:

$$\mathcal{C}_g = \sum_{i \in [M]} \pi_g^i \mathcal{D}_g^i. \tag{17}$$

Following [18], we refer to the distribution $\mathcal{C}_g$ as the participated client's data distribution for the g-th group. In our setting, the group assignment formalized above corresponds to the execution of the function `Select` : $\mathcal{X} \to [G]$ assigning data to different groups.

We will also consider the empirical versions of the above distributions. Formally, let $\hat{\mathcal{D}}_i$ the induced local empirical distribution of client i by sampling N_g^i iid samples from each $\mathcal{D}_g^i$. Further, let $N_i = \sum_{g \in [G]} N_g^i$ denote the total number of samples per client and $N_g = \sum_{i=1}^{M} N_g^i$ denote the total number of data samples from group g. Onwards, we assume that the mixture weights in (16) are set as $\pi_g^i = N_g^i / N_i$. Thus, we also define the empirical distribution $\hat{\mathcal{C}}_g$ of each group as $\hat{\mathcal{C}}_g = \sum_{i \in [M]} \pi_g^i \hat{\mathcal{D}}_g^i = \sum_{i \in [M]} \frac{N_g^i}{N_i} \hat{\mathcal{D}}_g^i$. Finally, given a set $\{h_1, \ldots, h_G\}$ of G hypotheses $h_g \in \mathcal{H}, g \in [G]$ one corresponding to each group, we denote $h_{\texttt{Select}}$ the group-aware (data-point dependent) hypothesis determined by the `Select` function. In other words, $h_{\texttt{Select}} = \{h_1, \ldots, h_G\}_{\texttt{Select}(x)}$ denotes a hypothesis that when acting on datapoint x returns $h_{\texttt{Select}(x)}$, for a fixed function `Select` : $\mathcal{X} \to [G]$.

Algorithm 3 *SGPT* Inference Algorithm.

```
Input: Pre-trained Transformer $h_\theta$ with $U$ layers, layers set $\mathcal{U}_S$ to insert shared prompts and layers set $\mathcal{U}_G$ to insert group prompts, trainable weights $W = \{W_C, P_G, P_S\}$, Prompt Selection Module Select with orthogonal keys $K = \{k_g\}_{g=1}^{G}$, a single test sample $x_n$.
Extract representation $h_\theta(x_n)$ with pre-trained model $h_\theta$
$g \leftarrow \arg\max_{g \in \mathcal{G}} \cos(h_\theta(x_n), b_g)$                                   ▷ Obtain group ID
Select corresponding group prompt $p_g$
Encode $x_n$ into $E_0$                          ▷ Encode image into patch embedding (Section 2.2)
for $i \to U$ do
    if $i \in \mathcal{U}_S$ then
        $[cls_1, Z_1^S, E_1] = h_\theta^{(i)}\left(\left[cls_0, P_S^{(i)}, E_0\right]\right).$        ▷ Insert shared prompts
    else if $i < u$ then
        $[cls_i, Z_i^S, E_i] = h_\theta^{(i)}\left([cls_{i-1}, Z_{i-1}^S, E_{i-1}]\right).$
    else if $i \in \mathcal{U}_G$ then
        $[cls_u, Z_u^g, Z_u^S, E_u] = h_\theta^{(u)}\left(\left[cls_{u-1}, p_g^{(i)}, Z_{u-1}^S, E_{u-1}\right]\right).$        ▷ Insert group prompt
    else
        $[cls_i, Z_i^g, Z_i^S, E_i] = h_\theta^{(i)}\left([cls_{i-1}, Z_{i-1}^g, Z_{i-1}^S, E_{i-1}]\right).$
    end if
end for
$y^* \leftarrow f(W_C)$                                   ▷ Final Prediction
return final logits prediction $y^*$
```

C.2. Lemmas

In this section, we cover some technical lemmas which are useful for proving our main result. Lemma C.1 below splits the participated clients' distribution risk into the risks of each individual's group.

Lemma C.1 (Split the distribution risk)**.** *Let fixed function* `Select` $: \mathcal{X} \to \{1, ..., G\}$. *For the group-aware hypothesis* $h_{\texttt{Select}}$ *that selects among hypotheses* $\{h_1, \ldots, h_G\}$*, it holds for any client* $i \in [M]$ *that*

$$\mathcal{L}_{\mathcal{D}_i}(h_{\texttt{Select}}) = \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{D}_g^i}(h_g).$$

Proof. By definition of the population risk:

$$\mathcal{L}_{\mathcal{D}_i}(h_{\texttt{Select}}) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\ell(h_{\texttt{Select}}, x, y)].$$

The desired follows from this by recalling the decomposition in (16) with weights mixing weights $\pi_g^i = \frac{N_g^i}{N_i}$. □

The next lemma is useful to derive global and local performance gap.

Lemma C.2 (Bound on the generalization error)**.** *Assume the loss is bounded in* $[0, 1]$ *and fix any client* $i \in [M]$. *Then with probability at least* $1 - \delta$ *over the training set,*

$$\sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\mathcal{C}_g}(h_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\widehat{\mathcal{C}}_g}(h_g) \leq 2\sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \Re_{\mathcal{C}_g, N_g}(\mathcal{H}). \tag{18}$$

Proof. For any set of real numbers $a_1, ..., a_q$ and $b_1, ..., b_1$ observe that $\min_i a_i \leq \max_i a_i$ and $\min_i b_i = -\max_i -b_i$, we get

$$\min_i a_i - \min_i b_i \leq \max_i (a_i - b_i).$$

Using this it holds that

$$\sum_{g=1}^{G}\frac{N_g^i}{N_i}\min_{h_g\in H}\mathcal{L}_{\mathcal{C}_g}(h_g)-\sum_{g=1}^{G}\frac{N_g^i}{N_i}\min_{h_g\in H}\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\leq\sum_{g=1}^{G}\frac{N_g^i}{N_i}\max_{h_g}\left(\mathcal{L}_{\mathcal{C}_g}(h_g)-\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right)\,.$$

Since the loss is bounded in $[0,1]$, changing one sample changes the above term by at most one. Thus, by the McDiarmid's inequality, with probability at least $1-\delta$,

$$\sum_{g=1}^{G}\frac{N_g^i}{N_i}\max_{h_g}\left(\mathcal{L}_{\mathcal{C}_g}(h_g)-\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right)\leq\frac{1}{N_i}\mathbb{E}\left[\sum_{g=1}^{G}N_g^i\max_{h_g}\left(\mathcal{L}_{\mathcal{C}_g}(h_g)-\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right)\right]+\sqrt{\frac{1}{N_i}\log\frac{1}{\delta}}. \tag{19}$$

Moreover, note that:

$$\begin{aligned}\mathbb{E}\left[\sum_{g=1}^{G}\frac{N_g^i}{N_i}\max_{h_g}\left(\mathcal{L}_{\mathcal{C}_g}(h_g)-\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right)\right]&=\sum_{g=1}^{G}\frac{N_g^i}{N_i}\mathbb{E}\left[\max_{h_g}(\mathcal{L}_{\mathcal{C}_g}(h_g)-\mathcal{L}_{\hat{\mathcal{C}}_g}(h_g))\right]\\&\leq 2\sum_{g=1}^{G}\frac{N_g^i}{N_i}\Re_{\mathcal{C}_g,N_g}(\mathcal{H}).\end{aligned} \tag{20}$$

Combining Eq. (19) and Eq. (20), completes the proof. □

C.3. Proof of Theorem 1

We are now ready to prove Theorem 4.1.

Proof. Given Lemma C.1, we split the distribution risk. Thus, the global-to-local performance gap becomes

$$\sum_{g=1}^{G}\frac{N_g^i}{N_i}\mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g)-\min_{h\in H}\mathcal{L}_{\mathcal{D}_i}(h),$$

where $\widehat{h}_g=\arg\min_{h\in\mathcal{H}}\mathcal{L}_{\widehat{\mathcal{C}}_g}(h)$ denote the empirical model for data group g. Next, observe from (16) that

$$\begin{aligned}\min_{h\in H}\mathcal{L}_{\mathcal{D}_i}(h)&=\min_{h\in\mathcal{H}}\sum_{g=1}^{G}\frac{N_g^i}{N_i}\mathcal{L}_{\mathcal{D}_g^i}(h)\\&\geq\sum_{g=1}^{G}\frac{N_g^i}{N_i}\min_{h_g\in H}\mathcal{L}_{\mathcal{D}_g^i}(h_g)\,.\end{aligned}$$

Denote $h^\star_{gi} = \min_{h_g \in H} \mathcal{L}_{\mathcal{D}^i_g}(h_g)$, we then get the following:

$$
\begin{aligned}
\sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) &\le \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\mathcal{D}_g^i}(h_g) \\
&= \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left[\mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) + \mathcal{L}_{\mathcal{D}_g^i}(h^\star_{gi}) - \mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) \right] \\
&= \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left[\mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) + \mathcal{L}_{\mathcal{D}_g^i}(h^\star_{gi}) - \mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) \right] \\
&\quad - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) \\
&= \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) \\
&\quad + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) - \mathcal{L}_{\mathcal{D}_g^i}(h^\star_{gi}) \right) \\
&\quad + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(h^\star_{gi}) \qquad (21)
\end{aligned}
$$

Observing that $\sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\mathcal{C}_g}(h_g) \le \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(h^\star_{gi})$ we get

$$
\begin{aligned}
(25) \le \sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g \in H} \left| \mathcal{L}_{\mathcal{D}_g^i}(h_g) - \mathcal{L}_{\mathcal{C}_g}(h_g) \right| + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g \in H} \left| \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(h_g) - \mathcal{L}_{\widehat{\mathcal{C}}_g}(h_g) \right| \\
+ \sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\widehat{\mathcal{C}}_g}(h_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \min_{h_g \in H} \mathcal{L}_{\mathcal{C}_g}(h_g) \,.
\end{aligned}
$$

Then, combining lemma C.2, absolute homogeneity of Rademacher complexity, and the definition of the discrepancy in Eq. (15), we will have with probability $1 - \delta$,

$$
\sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) \le \sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + 2 \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathfrak{R}_{\mathcal{C}_g, N_g}(\mathcal{H}) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + \mathrm{disc}(\widehat{\mathcal{D}}_g^i, \widehat{\mathcal{C}}_g) \right) \qquad (22)
$$

When the VC dimension of Hypothesis class $\mathcal{H}$ is d, then we can obtain:

$$
\sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{D}}_g^i}(\widehat{h}_g) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) \le \sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + 2 \sum_{g=1}^{G} \frac{N_g^i}{N_i} \sqrt{\frac{2d}{N_g} log(\frac{eN_g}{d})} + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + \mathrm{disc}(\widehat{\mathcal{D}}_g^i, \widehat{\mathcal{C}}_g) \right) \qquad (23)
$$

This completes the proof of Theorem 4.1.

C.4. Discussion on the Distribution Discrepancy

We detailly discuss the distribution discrepancy term in Eq. 14. Combined with Eq. 13, our DD term can be expressed as $\sum_g \mathrm{disc}(\mathcal{D}_g^i, \sum_i \pi_g^i D_g^i)$, showing the disparity between group g's data from all clients and group g's data on client i's. The proposed selection module aims to cluster data across clients from the same distribution ($D_g^i \approx D_g^j \approx D_g$ for $i, j \in [M]$), thus reducing the DD term. Despite the challenges of distributed data clustering, our selection module showed strong performance (see Table 4) and even comparable to centralized K-means results (App. Figure 6. (a) and App. D.2). In the case of random or no selection module to group data, DD term becomes $\mathrm{disc}(\sum_j D_j, D_i) = \sum_{j \neq i} \mathrm{disc}(D_j, D_i)$ for client i, showing significant disparity between all clients' combined data and client i's data. □

C.5. Performance Gap on Real Distribution

In this section, we follow the idea in [1] to give the gap between the population loss 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. Different from [1] that focus on clustering clients, here we focus on clustering data into groups: $\mathcal{L}_{\mathcal{D}_i}(\widehat{h}_{\texttt{Select}}) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h)$. Different from theirs based on clustering clients into non-overlapping coalitions, we focus on learning parameters for each data group, allowing each client to benefit from knowledge distilled from all other clients' datasets [36]. We present the theorem below:

Theorem C.3. *Assume the loss function ℓ is bounded in $[0,1]$ and the function `Select` is a data grouping method. Let $\mathfrak{R}_{\mathcal{D},m}(\mathcal{H})$ represent the Rademacher complexity of the hypothesis class $\mathcal{H}$ over the distribution $\mathcal{D}$ with m samples. Then, with a probability of at least $1-\delta$ over the training set,*

$$\mathcal{L}_{\mathcal{D}_i}(\widehat{h}_{\texttt{Select}}) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) \leq 2\sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + 4\sum_{i=1}^{G} \frac{N_g^i}{N_i} \mathfrak{R}_{\mathcal{C}_g,N_g}(\mathcal{H}) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(2\,\mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g)\right), \tag{24}$$

where $\mathrm{disc}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \max_{h \in \mathcal{H}} |\mathcal{L}_{\mathcal{D}_1}(h) - \mathcal{L}_{\mathcal{D}_2}(h)|$.

The obtained theorem also suggests that tuning G is important in achieving optimal performance, which agrees with theorem 4.1.

Proof.

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}^i}(\widehat{h}_g) &\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) + \mathcal{L}_{\mathcal{D}^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) \\
&= \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{D}_g^i}(\widehat{h}_g) - \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) \\
&= \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathcal{L}_{\mathcal{C}_g}(\widehat{h}_g) - \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g)\right) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\widehat{\mathcal{C}}_g}(\widehat{h}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g} \left(\mathcal{L}_{\mathcal{C}_g}(h_g) - \mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(h_{gi}^\star) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) \\
&+ \left| \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathcal{L}_{\widehat{\mathcal{C}}_g}(h_{gi}^\star) - \mathcal{L}_{\mathcal{C}_g}(h_{gi}^\star)\right) \right| + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g} \left(\mathcal{L}_{\mathcal{C}_g}(h_g) - \mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{C}_g}(h_{gi}^\star) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + 2\sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g} \left(\mathcal{L}_{\mathcal{C}_g}(h_g) - \mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{D}_g^i}(h_{gi}^\star) + \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + 2\sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g} \left(\mathcal{L}_{\mathcal{C}_g}(h_g) - \mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right) \\
&+ \sum_{g=1}^{G} \frac{N_g^i}{N_i} \left(\mathcal{L}_{\mathcal{C}_g}(h_{gi}^\star) - \mathcal{L}_{\mathcal{D}_g^i}(h_{gi}^\star)\right) \\
&\leq \sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{D}_g^i}(h_{gi}^\star) + 2\sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathrm{disc}(\mathcal{D}_g^i, \mathcal{C}_g) + 2\sum_{g=1}^{G} \frac{N_g^i}{N_i} \max_{h_g} \left(\mathcal{L}_{\mathcal{C}_g}(h_g) - \mathcal{L}_{\hat{\mathcal{C}}_g}(h_g)\right) \qquad (25)
\end{aligned}$$

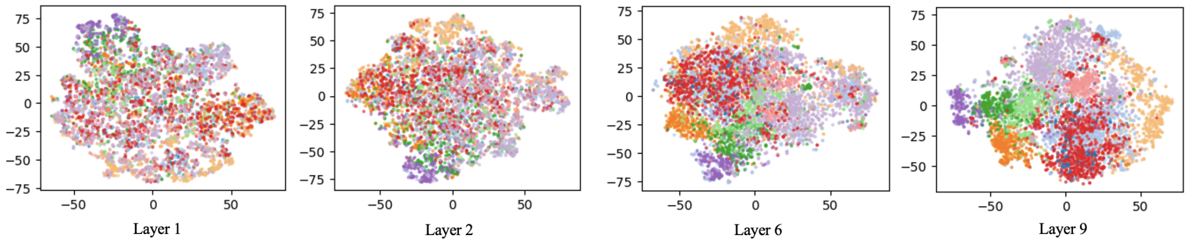


Figure 5. T-SNE maps of CIFAR-100 data features processed by the different layers of the ImageNet-21K pre-trained ViT-16/B model. Data from different coarse classes are labeled with different colors.

Then, combining lemma C.2, absolute homogeneity of Rademacher complexity, and the definition of the discrepancy in Eq. (15), we will have

$$\sum_{g=1}^{G} \frac{N_g^i}{N_i} \mathcal{L}_{\mathcal{D}_g^i}(\widehat{h}_g) - \min_{h \in H} \mathcal{L}_{\mathcal{D}_i}(h) \leq 2\sqrt{\frac{\log \frac{1}{\delta}}{N_i}} + 4\sum_{g=1}^{G} \frac{N_g^i}{N_i} \Re_{\mathcal{C}_g, N_g}(\mathcal{H}) + 2\sum_{g=1}^{G} \frac{N_g^i}{N_i} \operatorname{disc}(\mathcal{D}_g^i, \mathcal{C}_g)$$

□

D. Additional Analysis

D.1. Feature T-SNE map of pre-train model

In this section, we examine features outputted from various layers of a pre-trained ViT model. As illustrated in Fig. 5, features from different classes processed by the early layers of a pre-trained ViT are uniformly distributed on the manifold, indicating shared information across classes. In later layers, the features become more specialized and cluster together, thereby introducing higher heterogeneity in FL. As a result, it validates our motivation in introducing shared prompts into lower layers for common information and group prompts into higher layers for specialized information (Section 3.1).

D.2. The influence of Momentum Ratio

In this section, we perform an ablation study on the two momentum ratios in Eq. 7. We use the CIFAR-100 dataset with $s = 10$ as a case study.

Key Momentum Ratio. We conduct an analysis of the key momentum ratio and begin by applying centralized K-Means to the training data to generate cluster labels for the data. Subsequently, we assess the congruence between the groups learned through our distributed approach and the clusters identified by K-Means. To be specific, we first calculate the normalized contingency matrix [50] between cluster results from centralized K-Means and *SGPT* and obtain the overlapping ratio $Acc_{overlap}$ by summing the maximum values of each row in this matrix. Then we evaluate the quality of the centralized K-Means result, Q_{kmeans}, by calculating the ratio of data from the same class clustered into the same group. Finally, we calculate $\frac{Acc_{overlap}}{Q_{kmeans}}$ to obtain the congruence score. Fig. 6a demonstrates that `Select` can match the performance of centralized K-Means. Notably, with a momentum setting of $\alpha_k = 0.5$, it achieved its highest congruence score at 86.8%.

Group Momentum Ratio. Based on the optimal key momentum ratio, we study the influence of group momentum ratio α_g on both the global accuracy and the worst local accuracy. Fig. 6b illustrates that an increase in α_g enhances the worst local accuracy, as a higher α_g incorporates more knowledge from previous rounds. As to the global accuracy, initially, it improves attributing to enhanced stability, but it starts to decline when α_g exceeds 0.5, indicating an over-rely on information from previous rounds. Therefore, the optimal group momentum ratio is $\alpha_k = 0.5$. We also report the best baseline performances with two vertical dashed lines, *SGPT* can outperform baselines across all momentum ratios.

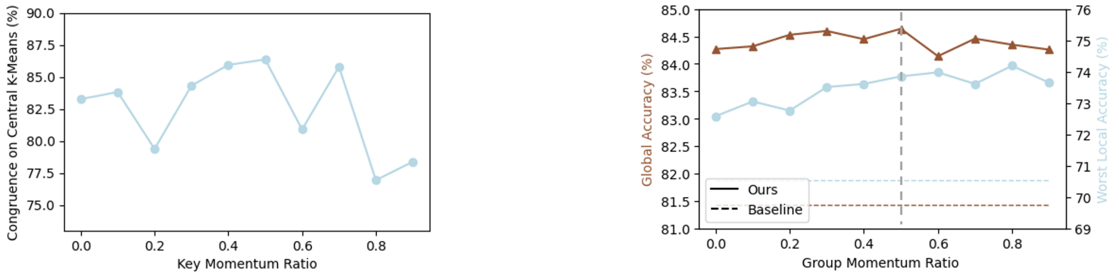
Consequently, without further declarations, we set the momentum ratio at 0.5 for both the key α_k and group prompt α_g.

E. Description of Heterogeneity

In this section, we describe the details of label heterogeneity and feature heterogeneity settings and provide examples of them.

E.1. Label Heterogeneity

For CIFAR-100, we follow [28, 38] to apply the "Pathological Partition", where each client is randomly assigned s classes. The sample rate on client i of selected class s is obtained by $a_{i,c}/\sum_j a_{j,c}$, where $a_{i,c} \sim U(.4, .6)$. Considering that s equals



(a) Influence of key momentum ratio α_k: We assess the congruence between the groups distributively learned by `Select` and those identified by centralized K-Means. Optimal congruence is observed at a key momentum ratio of $\alpha_k = 0.5$.

(b) Influence of group momentum ratio α_g: We study the influence of the group momentum ratio α_g on global accuracy and worst local accuracy. We highlight the sweet spot using a horizontal dashed line and the best baseline performance with two vertical dashed lines.

Figure 6. Ablation study on key momentum ratio α_k and group momentum ratio α_g.

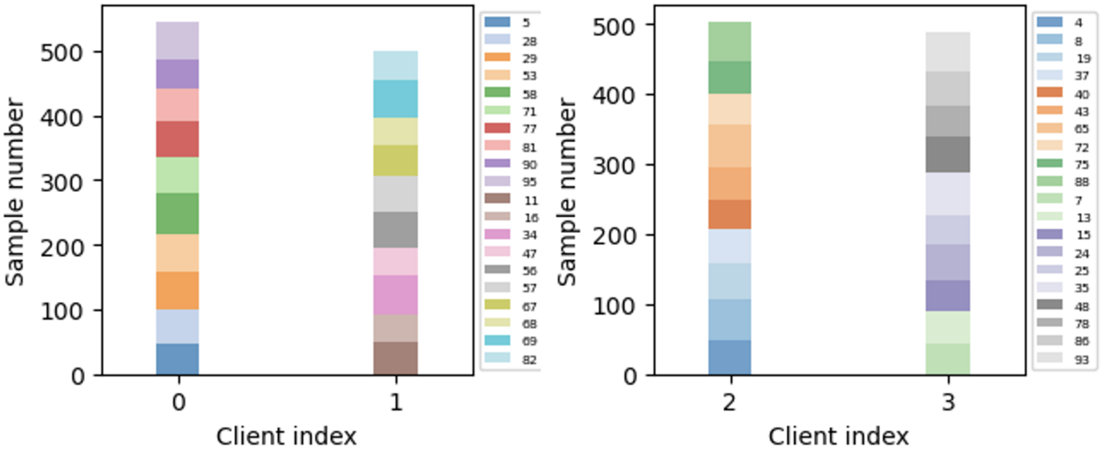


Figure 7. Examples of data distribution with $s = 10$ for four clients are presented. Due to the large number of classes (100 classes in CIFAR-100), every two clients are plotted on the same figure, with different colors indicating different classes.

10, we illustrate the data distribution for four clients out of a total of 100 in Fig. 7. As depicted, $a_{i,c}$ affects the data size of a class and introduces class imbalance within a client. Simultaneously, each client possesses 10 classes, the combination varies across clients, thereby introducing label heterogeneity. As s decreases, the variety of classes available to each client becomes limited, resulting in a restricted label distribution for each client and an increase in the number of samples per class. Regarding the Five Datasets, we allocate the data and ensure each client receives data from one dataset. Figure 8 presents image samples from five clients with each client representing one of the five datasets. This approach introduces label heterogeneity due to the unique nature of each dataset.

E.2. Feature Heterogneity

For conducting clients with *feature heterogeneity*, we follow the methodologies outlined in the latest benchmark for feature heterogeneity [56] as well as in the widely-referenced paper [31]. According to these sources, we assign a data domain to each client, with the total number of clients (M) set to 4 for Office-Caltech10 and 6 for DomainNet respectively. Image examples from these datasets are displayed in Fig.9 and Fig.10 for DomainNet and Office-Caltech10, respectively. As illustrated, different clients receive data from the same classes but sourced from various domains, thereby introducing feature heterogeneity.

Figure 8. Examples of images from five clients with each client representing one of the five datasets.

Figure 9. Examples of images from different clients with DomainNet.

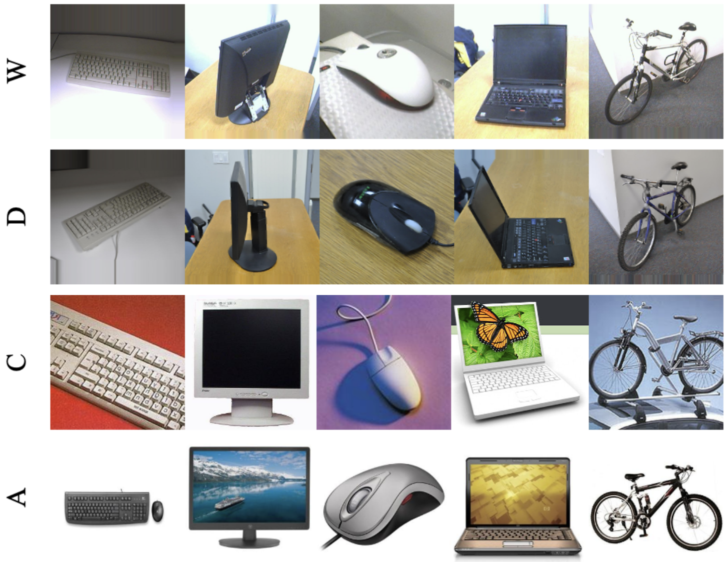


Figure 10. Examples of images from different clients with Office-Caltech10 dataset.