Appendix Roadmap

In the following appendix, we first discuss and analyze *how the heterogeneous client domains influence the personalization performance* in Appendix A. Then, we provide a series of convergence proofs related to our pFedSV algorithms, including 1) the convergence proof of dynamic top-k download mechanism in Appendix B, 2) the personalized performance convergence proof in Appendix C and 3) the generalization bounds analysis in Appendix D. Finally, according to the discussion section in our main content, we will provide more details about how to tackle the communication overhead in Appendix E and model privacy issues in Appendix F, which are raised by the model downloading. Besides, more experiments extension are shown in Appendix G, where we extend to more complex dataset (CIFAR-100), model (VGG-19) and larger scale client population (200 clients).

A The Influences of Heterogeneous Client Domains for Personalization

In this section, we explain what are heterogeneous client domains and how they influence the personalization performance in an agnostic federated learning system. In a typical federated learning system, clients have personalized domains or tasks, so they need to look for other clients in this agnostic system, who have the same or similar class labels. The collaboration among these clients, which called domain-relevant clients, can promote the personalized performance, while other domain-irrelevant clients will degrade the performance.

Take the example of image classification problem on MNIST dataset, we constructed a FL scenario with 5 clients $\{A, B, C, D, E\}$, assume that the personalized task of client A now is the even number classification, i.e., $\{0, 2, 4, 6, 8\}$. The label distribution of other clients are: $B : \{0, 2, 4\}$, $C : \{6, 8\}$, $D : \{1, 3, 5\}$ and $E : \{7, 9\}$. It's very clear that class labels owned by client B and C are overlapping with client A. Thus, they ($B\&C$) are A's domain-relevant clients, while the other two clients ($D\&E$) are domain-irrelevant. Now, we generate the personalized model targeting client A with FedAvg algorithm in the following two cases. In case (a), we aggregate the models of all 5 clients to generate a personalized model for client A, which refers to collaboration mixed with domain-irrelevant clients (denoted by IR in Fig. 1). In case (b), we only aggregate the models from client A, B and C to generate a personalized model of client A, which refers to collaboration with only domain-relevant clients (denoted by Re in Fig. 1).

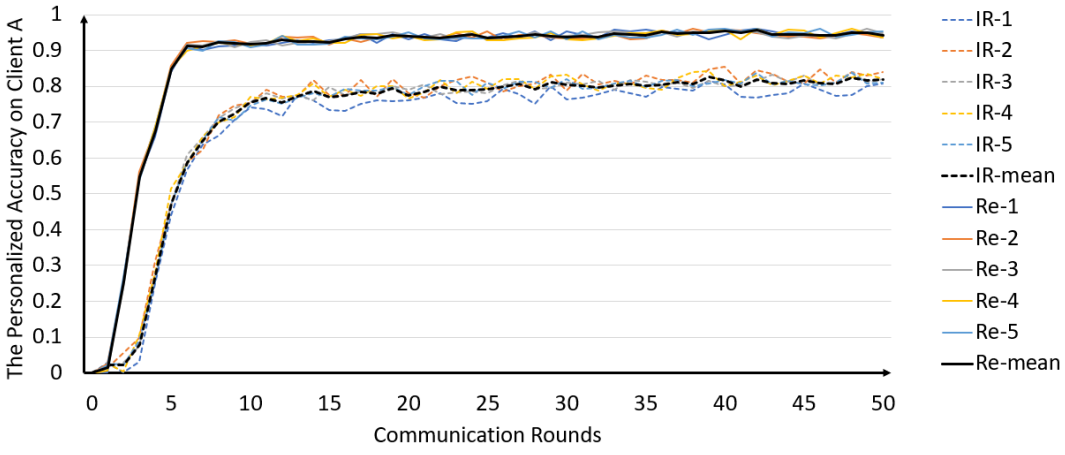

Figure 1: The influence of domain relevance on the personalized performance of client A (MNIST).

Fig. 1 shows the personalized model accuracy of client A in 50 communication rounds on MNIST dataset for both cases, where we conduct 5 times of repeat experiments on each case (shown by different colorful lines) and the black lines (Solid & dashed) are the mean accuracy of all repeat experiments. We can clearly observe that the personalized accuracy of client A converges rapidly in a few communication rounds when collaborating with other domain-relevant clients only. However,

when those domain-irrelevant clients are mixed within the collaboration, they will degrade the final personalized accuracy of client A. This results illustrate that each client should identify their own domain-relevant clients in an agnostic federated learning system for collaboration.

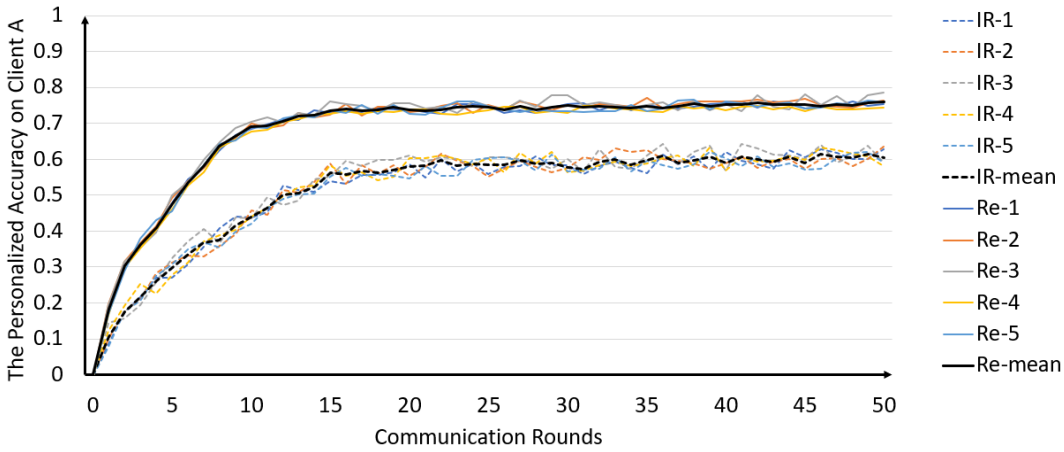

Figure 2: The influence of domain relevance on the personalized performance of client A (CIFAR-10).

We also conduct more experiments on the CIFAR-10 dataset to further validate the influence of domain relevance, where CIFAR-10 has 10 different labels including $\{airplane, automobile, bird, cat, deer, frog, dog, horse, ship, truck\}$. These labels can be roughly divided into two main categories: 1) **animal** labels with $\{bird, cat, deer, frog, dog, horse\}$ and 2) **industrial product** labels with $\{airplane, automobile, ship, truck\}$.

We still use the 5 clients setting as follows: $A : \{bird, cat, deer, frog, dog, horse\}$, $B : \{bird, cat, deer\}$, $C : \{frog, dog, horse\}$, $D : \{airplane, automobile\}$ and $E : \{ship, truck\}$, while the remaining experiment settings are the same as above. Now the personalized task of client A is the animal classification in CIFAR-10, and the results shown in Fig. 2 demonstrate the same conclusion about the domain relevance influence before.

B The Convergence Proof of Dynamic Top-k Download Mechanism

The convergence proof of dynamic top-k download mechanism ensures that our algorithm can quickly and accurately help each client identify their respective domain-related clients in an agnostic federated learning system for personalization, which can be achieved within a short-term period (just a few communication rounds in the beginning).

Table 1: The example of the pathological data Non-IID setting with 10 clients on MNIST dataset.

Client index	A	B	C	D	E
Label Distribution	$[3, 6]$	$[4, 9]$	$[1, 0]$	$[2, 7]$	$[8, 5]$
Client index	F	G	H	I	J
Label Distribution	$[8, 1]$	$[6, 7]$	$[5, 3]$	$[4, 9]$	$[0, 2]$

Assume that there are total n clients with 100% participation, the local data distributions of these clients follow the pathological data Non-IID setting, where each client is randomly assigned m types of labels. An example on MNIST dataset (10 labeled digits from 0 to 9) with $m = 2$ is shown in Table 1 for reference. The domain heterogeneity is defined as each client has different label distribution, while the domain relevance is defined as there are same class labels between

different clients. Therefore, we can observe from ground truth of Table 1 that each client has m other domain-relevant clients in this setting from an omniscient perspective.

Suppose the initial model download number for each client is k. Then, we provide the convergence proof of our dynamic top-k download mechanism. Take the personalization process of client A as an example, there are two conditions for the settings of hyperparameters m and k, which is $m < k$ or $m > k$ and we will explain them one by one.

When $m < k$: In the first round, each client will randomly download k copies of other clients' models from the server-side and there are various (C_n^k) possible model combinations.

- **For the best case**, other m domain-related clients' models are all included in the initial k copies, that is for $\forall i \in \{m\}$, we have $i \in \{k\}$. Thus, we can identify all domain-relevant clients of client A in the first round, where the SV of the domain-relevant clients is positive and the domain-irrelevant clients are negative.

- **For the worst case**, none of the m domain-related models is included in the first k copies, that is for $\forall i \in \{m\}$, we have $i \notin \{k\}$. Next, we proof the maximum number of rounds that is required to identify the m domain-relevant clients from all n clients when the worst case occurs in each round. For the first round, since k copies of models are all from the domain-irrelevant clients, their SV will be negative in the evaluation process, which makes their relevance score be negative after updating. Therefore, according to the top-k rule, these clients will not be selected in the next round because the relevant scores of other clients who have never been selected are the initial 0, which is larger than negative scores. The worst case will continue until a certain round t, which satisfies $tk > n - m - 1$ (1 is client A itself). It means that in round t, we have excluded all domain-irrelevant clients with negative SV, and the remaining clients are all domain-relevant clients. Since $k > m$ (they are both integers), we have $(t+1)k = tk+k > n-m-1+k > n+(k-m-1) \geq n$, which means that we must be able to find all domain-relevant clients in the next round $t+1$. Finally, we prove that it takes at most $\lceil \frac{n-m-1}{k} \rceil + 1$ round to identify all other domain-relevant clients.

When $m > k$, by the similar logic, we have the following proofs.

- **For the best case**, since $m > k$, we cannot include all m domain-relevant clients in the first round with only k downloaded models. Therefore, the process will continue until all clients are scanned by once. Thus, we need $\lceil \frac{m}{k} \rceil$ round to identify all domain-relevant clients.

- **For the worst case**, we need $\lceil \frac{n-m-1}{k} \rceil$ rounds to exclude all domain-irrelevant clients and then we still need up to $\lceil \frac{m}{k} \rceil$ rounds to identify all domain-relevant clients. Finally, it takes at most $\lceil \frac{n-m-1}{k} \rceil + \lceil \frac{m}{k} \rceil$ rounds.

Normally, to ensure efficient traversal, we will set a large value of k at the beginning. Although a large k leads to a large communication overhead in the beginning, it can help the client rapidly scan all other clients and converge to a specific value $k = m$, which is equal to the number of other domain-relevant clients. The issue of communication overhead will be further improved later in Appendix E.

C The Personalized Performance Convergence Analysis of pFedSV.

The personalized performance convergence analysis for each client is the same, so we only focus on one client $i \in \{N\}$. Consider a scenario where each client parallel performs E local SGD step to update their own model. Then, they will communicate with the server to download the model for personalized model aggregation, which is denoted as the synchronization step. First, we analyze the case on pFedSV that all other clients (including both domain-relevant or domain-irrelevant clients) participate in the aggregation step to generate the personalized model.

C.1 Additional Notation

Let θ_t^k be the model parameter of k-th client in t-th step. Let E be the local update epoch number. Let $\mathcal{I}_E$ be the set of synchronization steps, i.e., $\mathcal{I}_E = \{nE \mid n = 1, 2, \cdots\}$. If $t+1 \in \mathcal{I}_E$, i.e., the model update with all participants can be described as:

$$\mathrm{v}_{t+1}^k = \theta_t^k - \eta_t \nabla \mathcal{L}_k(\theta_t^k, \xi_t^k) \tag{1}$$

$$\theta_{t+1}^k = \begin{cases} \mathrm{v}_{t+1}^k, & if \quad t+1 \notin \mathcal{I}_E \\ \sum_{k=1}^N p_k \mathrm{v}_{t+1}^k, & if \quad t+1 \in \mathcal{I}_E \end{cases} \tag{2}$$

Here, an additional variable v_{t+1}^k is introduced to represent the immediate result of one step SGD update from θ_{t+1}^k. We regard θ_{t+1}^k as the parameter obtained after communication steps.

In following analysis, we define two virtual sequences $\bar{\mathrm{v}}_{t,i} = \sum_{k=1}^N p_k \mathrm{v}_{t+1}^k$ and $\bar{\theta}_{t,i} = \sum_{k=1}^N p_k \theta_{t+1}^k$. $\bar{\mathrm{v}}_{t+1,i}$ comes from an single step of SGD from $\bar{\theta}_{t,i}$. For convenience, we also define $\bar{g}_{t,i} = \sum_{k=1}^N p_k \nabla \mathcal{L}_k(\theta_t^k)$ and $g_{t,i} = \sum_{k=1}^N p_k \nabla \mathcal{L}_k(\theta_t^k, \xi_t^k)$. Thus, we have $\bar{\mathrm{v}}_{t+1,i} = \bar{\theta}_{t,i} - \eta_t g_{t,i}$ and $\mathbb{E} g_{t,i} = \bar{g}_{t,i}$.

C.2 Key Lemmas

To clearly show our proof, it is necessary to define some lemmas before the main theorem. The proof of these lemmas can be found in [1] and we only focus on the main theorem.

Lemma 1 *The results of one step SGD. Assume the assumption* 1 *and* 2 *hold. we have*

$$\mathbb{E}||\bar{\mathrm{v}}_{t+1,i} - \theta_i^*|| \leq (1 - \eta_t \mu)\mathbb{E}||\bar{\theta}_{t,i} - \theta_i^*||^2 + \eta_t^2 \mathbb{E}||g_{t,i} - \bar{g}_{t,i}||^2 + 6L\eta_t^2 \Gamma + 2\mathbb{E}\sum_{k=1}^N p_k ||\bar{\theta}_{t,i} - \theta_k^t||^2$$

where $\Gamma = \mathcal{L}_i^* - \sum_{k=1}^N p_k \mathcal{L}_k^* \geq 0$.

Lemma 2 *Bounding the variance. Assume Assumption* 3 *holds. It follows that*

$$\mathbb{E}||g_{t,i} - \bar{g}_{t,i}||^2 \leq \sum_{k=1}^N p_k^2 \sigma_k^2$$

Lemma 3 *Bounding the divergence of* $\{\theta_{t,i}^k\}$. *Assume Assumption* 4 *holds, that* η_t *is non-increasing and* $\eta_t \leq 2\eta_{t+E}$ *for all* $t \geq 0$. *It follows that*

$$\mathbb{E}\left[\sum_{k=1}^N p_k ||\bar{\theta}_{t,i} - \theta_k^t||^2\right] \leq 4\eta_t^2 (E-1)^2 G^2.$$

C.3 Full Proof of Theorem 1

Proof. No matter whether $t+1 \in \mathcal{I}_E$ or $t+1 \notin \mathcal{I}_E$, we always have the following equation: $\bar{\theta}_{t+1,i} = \bar{\mathrm{v}}_{t+1,i}$. Let $\Delta_t = \mathbb{E}||\bar{\theta}_{t,i} - \theta_i^*||^2$. From Lemma 1, Lemma 2 and Lemma 3, it follows that

$$\Delta_{t+1} \leq (1 - \eta_t \mu)\Delta_t + \eta_t^2 B$$

where

$$B = \sum_{k=1}^N p_k^2 \sigma_k^2 + 6L\Gamma + 8(E-1)^2 G^2$$

For a diminishing stepsize, $\eta_t = \frac{\beta}{t+\gamma}$ for some $\beta > \frac{1}{\mu}$ and $\gamma > 0$ such that $\eta_1 \leq \min\{\frac{1}{\mu}, \frac{1}{4L}\}$ and $\eta_t \leq 2\eta_{t+E}$. We prove that $\Delta_t \leq \frac{v}{\gamma+t}$ where $v = \max\{\frac{\beta^2 B}{\beta\mu - 1}, (\gamma+1)\Delta_1\}$.

First, the definition of v ensures that it holds for $t = 1$. Assume the conclusion holds for some t, it follows that

$$\begin{aligned}\Delta_{t+1} &\leq (1-\eta_t\mu)\Delta_t + \eta_t^2 B \\ &\leq \left(1 - \frac{\beta\mu}{t+\gamma}\right)\frac{v}{t+\gamma} + \frac{\beta^2 B}{(t+\gamma)^2} \\ &= \frac{t+\gamma-1}{(t+\gamma)^2}v + \left[\frac{\beta^2 B}{(t+\gamma)^2} - \frac{\beta\mu-1}{(t+\gamma)^2}v\right] \\ &\leq \frac{v}{t+\gamma+1}\end{aligned} \tag{3}$$

Then,according to the L-smoothness of $\mathcal{L}_i(\cdot)$, we have

$$\mathbb{E}[\mathcal{L}_i(\bar{\theta}_{t,i})] - \mathcal{L}_i^* \leq \frac{L}{2}\Delta_t \leq \frac{L}{2}\frac{v}{\gamma+t}.$$

Specifically, if we choose $\beta = \frac{2}{\mu}$, $\gamma = \max\{8\frac{L}{\mu}, E\} - 1$ and denote $\kappa = \frac{L}{\mu}$, then $\eta_t = \frac{2}{\mu}\frac{1}{\gamma+t}$. One can verify that the choice of $\eta_t \leq 2\eta_{t+E}$ for $t \geq 1$. Then we have

$$v = \max\{\frac{\beta^2 B}{\beta\mu - 1}, (\gamma+1)\Delta_1\} \leq \frac{\beta^2 B}{\beta\mu-1} + (\gamma+1)\Delta_1 \leq \frac{4B}{\mu^2} + (\gamma+1)\Delta_1,$$

and

$$\mathbb{E}[\mathcal{L}_i(\bar{\theta}_{t,i})] - \mathcal{L}_i^* \leq \frac{L}{2}\frac{v}{\gamma+t} \leq \frac{\kappa}{\gamma+t}\left(\frac{2B}{\mu} + \frac{\mu(\gamma+1)}{2}\Delta_1\right)$$

D The Generalization Bounds Analysis of pFedSV.

Before the analysis of the generalization bound, we introduce the following notations. In PFL, each client has its own local data distribution $\mathcal{D}_i$ over domain $\Xi := \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^d$ is the input space and $\mathcal{Y}$ is the output space. For the empirical distribution $\hat{\mathcal{D}}_i$ by the given dataset, we assume that each client local model has access to an equal amount (m) of local data samples. For each client, we assume the local model θ as a mapping $\theta : \mathcal{X} \to \mathcal{Y}$. The cross-entropy loss function of task is defined as $\mathcal{L}(\theta(x), y) = \mathcal{L}(\hat{y}, y)$, where $\hat{y} := \theta(x)$. Note that $\mathcal{L}(\hat{y}, y)$ is convex with respect to $\hat{y}$. We denote $\arg\min_{\theta\in\Theta} \mathcal{L}_{\hat{\mathcal{D}}_i}(\theta)$ by $\theta_{\hat{\mathcal{D}}_i}$

According to the Domain Adaptation theory [2], we utilize the domain measurement tools developed below to analyze the generalization bound of the personalized model that is aggregated from an ensemble of other clients' model.

Theorem 1 *(Domain Adaptation) Considering the distribution $\mathcal{D}_S$ and $\mathcal{D}_T$, for every $\theta \in \Theta$ and any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of the samples), there exists:*

$$\mathcal{L}_{\mathcal{D}_T}(\theta) \leq \mathcal{L}_{\mathcal{D}_S}(\theta) + \frac{1}{2}d(\mathcal{D}_S, \mathcal{D}_T) + \lambda, \tag{4}$$

where $\lambda = \mathcal{L}_{\mathcal{D}_S}(\theta^*) + \mathcal{L}_{\mathcal{D}_T}(\theta^*)$, and $\theta^* := \arg\min_{\theta\in\Theta} \mathcal{L}_{\mathcal{D}_S}(\theta) + \mathcal{L}_{\mathcal{D}_T}(\theta)$ corresponds to the optimal joint model that minimize the combined loss.

Now we start the proof of Theorem 1 in the main content by two parts. 1) we first prove that the personalized model aggregated by FedAvg algorithm for each client is better than training with their own local data only. 2) Then, we prove that the aggregation by pFedSV only on other domain-relevant clients is better than FedAvg with all clients participation.

For the first part, we start with the risk of the personalized model of client i, $\mathcal{L}_{\mathcal{D}_i}(\frac{1}{n}\sum_j \theta_{\hat{\mathcal{D}}_j})$, which is aggregated from FedAvg with the participation of all other clients.

Considering the distance between $\mathcal{L}_{\mathcal{D}_i}(\frac{1}{n}\sum_j \theta_{\hat{\mathcal{D}}_j})$ and $\mathcal{L}_{\hat{\mathcal{D}}_i}(\theta_{\hat{\mathcal{D}}_i})$. By the convexity of $\mathcal{L}$ and Jensen inequality, we have

$$\mathcal{L}_{\mathcal{D}_i}(\frac{1}{n}\sum_j \theta_{\hat{\mathcal{D}}_j}) \leq \frac{1}{n}\sum_j \mathcal{L}_{\mathcal{D}_i}(\theta_{\hat{\mathcal{D}}_j}). \tag{5}$$

Using the domain adaptation theory, we transfer from domain $\mathcal{D}_i$ to $\mathcal{D}_j$,

$$\mathcal{L}_{\mathcal{D}_i}(\theta_{\hat{\mathcal{D}}_j}) \leq \mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) + \frac{1}{2}d(\mathcal{D}_j, \mathcal{D}_i) + \lambda_j, \tag{6}$$

where $\lambda_j := \mathcal{L}_{\mathcal{D}_i}(\theta^*) + \mathcal{L}_{\mathcal{D}_j}(\theta^*)$ and $\theta^* := \arg\min_{\theta\in\Theta} \mathcal{L}_{\mathcal{D}_i}(\theta) + \mathcal{L}_{\mathcal{D}_j}(\theta)$.

We can bound the risk with its empirical counterpart through Hoeffding in equality, which gives

$$\Pr\left[\left|\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) - \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j})\right| \geq \epsilon\right] \leq 2\exp\frac{-2m^2\epsilon^2}{\sum_{k=1}^{m}(b-a)^2}, \tag{7}$$

where $[a, b]$ is the range of loss function. In our case, the loss function is bounded in $[0, 1]$ so that $(b-a)^2 \leq 1$. Thus, with the probability at least $1 - \frac{\delta}{n}$, over the draw of m i.i.d. samples S_j from $\mathcal{D}_j$,

$$\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) \leq \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2}{\frac{\delta}{n}}}{2m}}, \tag{8}$$

Thus for n sources, we have

$$\begin{aligned}
&\Pr_{S_1\sim\mathcal{D}_1^m,\cdots,S_n\sim\mathcal{D}_n^m}\left[\bigcap_{j=1}^{n}\left\{\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) \leq \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2}{\frac{\delta}{n}}}{2m}}\right\}\right] \\
&= 1 - \Pr_{S_1\sim\mathcal{D}_1^m,\cdots,S_n\sim\mathcal{D}_n^m}\left[\bigcup_{j=1}^{n}\left\{\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) \geq \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2}{\frac{\delta}{n}}}{2m}}\right\}\right] \\
&\geq 1 - \sum_{j=1}^{n}\Pr_{S_1\sim\mathcal{D}_1^m,\cdots,S_n\sim\mathcal{D}_n^m}\left[\left\{\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) \geq \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2}{\frac{\delta}{n}}}{2m}}\right\}\right] \\
&\geq 1 - \delta.
\end{aligned} \tag{9}$$

Based on the definition of ERM, we have $\mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) \leq \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_i})$, where $\theta_{\hat{\mathcal{D}}_i}$ is the personalized model trained on client i. By using the definition of $\hat{\mathcal{D}}_i$ ($\hat{\mathcal{D}}_i = \frac{1}{n}\sum_j \hat{\mathcal{D}}_j$) and the linearity of expectation, we have

$$\frac{1}{n}\sum_j \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_j}) \leq \frac{1}{n}\sum_j \mathcal{L}_{\hat{\mathcal{D}}_j}(\theta_{\hat{\mathcal{D}}_i}) = \mathcal{L}_{\hat{\mathcal{D}}_i}(\theta_{\hat{\mathcal{D}}_i}). \tag{10}$$

Putting these equations together, we have probability of at least $1-\delta$ over $S_1 \sim \mathcal{D}_1^m, \cdots, S_n \sim \mathcal{D}_n^m$ that

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_i}(\frac{1}{n}\sum_j \theta_{\hat{\mathcal{D}}_j}) &\leq \frac{1}{n}\sum_j \mathcal{L}_{\mathcal{D}_i}(\theta_{\hat{\mathcal{D}}_j}) \\
&\leq \frac{1}{n}\sum_j \left(\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) + \frac{1}{2}d(\mathcal{D}_j, \mathcal{D}_i) + \lambda_k\right) \\
&\leq \frac{1}{n}\sum_j \left(\mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2n}{\delta}}{2m}} + \frac{1}{2}d(\mathcal{D}_j, \mathcal{D}_i) + \lambda_k\right) \\
&\leq \frac{1}{n}\sum_j \mathcal{L}_{\mathcal{D}_j}(\theta_{\hat{\mathcal{D}}_j}) + \sqrt{\frac{\log\frac{2n}{\delta}}{2m}} + \frac{1}{n}\sum_j \left(\frac{1}{2}d(\mathcal{D}_j, \mathcal{D}_i) + \lambda_k\right) \\
&\leq \mathcal{L}_{\mathcal{D}_i}(\theta_{\hat{\mathcal{D}}_i}) + \sqrt{\frac{\log\frac{2n}{\delta}}{2m}} + \frac{1}{n}\sum_j \left(\frac{1}{2}d(\mathcal{D}_j, \mathcal{D}_i) + \lambda_k\right),
\end{aligned} \tag{11}$$

where $\lambda_k = \inf_{\theta\in\Theta}(\mathcal{L}_{\mathcal{D}_i}(\theta) + \mathcal{L}_{\mathcal{D}_j}(\theta))$.

For the second part, we proof that the collaboration with only other domain-relevant clients by pFedSV is better than the collaboration with all clients mixed by FedAvg. According to the domain relevance theory in Appendix.A and the personalized performance convergence analysis in Appendix. C, the collaboration mixed with domain-irrelevant clients will degrade the personalized model performance. Assume the domain-relevant clients set of client i is $\mathcal{R}$. Thus, we have

$$\mathcal{L}_{\mathcal{D}_i}\left(\frac{1}{n}\sum_{j\in\mathcal{R}}\theta_{\hat{\mathcal{D}}_j}\right) \leq \mathcal{L}_{\mathcal{D}_i}\left(\frac{1}{n}\sum_{j\in\mathcal{R}}\theta_{\hat{\mathcal{D}}_j}+\frac{1}{n}\sum_{j\notin\mathcal{R}}\theta_{\hat{\mathcal{D}}_j}\right) = \mathcal{L}_{\mathcal{D}_i}(\frac{1}{n}\sum_{j}\theta_{\hat{\mathcal{D}}_j}). \tag{12}$$

The domain adaptation theory provide insights that for two models ($\theta_{\mathcal{D}_S}$ and $\theta_{\mathcal{D}_{S'}}$) trained on different source domains (S and S'). The higher the relevance between the source and target domains, the better the performance of the models, which means:

$$\mathcal{L}_{\mathcal{D}_T}(\theta_{\mathcal{D}_S}) \leq \mathcal{L}_{\mathcal{D}_T}(\theta_{\mathcal{D}_{S'}}), if\ d(\mathcal{D}_T,\mathcal{D}_S) \leq d(\mathcal{D}_T,\mathcal{D}_{S'})$$

On the other hand, our pFedSV can precisely identify the domain relevance and assign the aggregation weights $w^*_{\hat{\mathcal{D}}_j}$ according the relevance (The higher relevance, the larger weights). Thus, we have

$$d\left(\theta^*_{\mathcal{D}_i}, \sum_{j\in\mathcal{R}} w^*_{\hat{\mathcal{D}}_j}\theta_{\hat{\mathcal{D}}_j}\right) \leq d\left(\theta^*_{\mathcal{D}_i}, \frac{1}{n}\sum_{j\in\mathcal{R}}\theta_{\hat{\mathcal{D}}_j}\right)$$

and then

$$\mathcal{L}_{\mathcal{D}_i}\left(\sum_{j\in\mathcal{R}} w^*_{\hat{\mathcal{D}}_j}\theta_{\hat{\mathcal{D}}_j}\right) \leq \mathcal{L}_{\mathcal{D}_i}\left(\frac{1}{n}\sum_{j\in\mathcal{R}}\theta_{\hat{\mathcal{D}}_j}\right)$$

Finally, we have

$$\mathcal{L}_{\mathcal{D}_i}\left(\sum\nolimits_j w^{t*}_{\hat{\mathcal{D}}_j}\theta^t_{\hat{\mathcal{D}}_j}\right) \leq \mathcal{L}_{\mathcal{D}_i}\left(\frac{1}{n}\sum\nolimits_j \theta^t_{\hat{\mathcal{D}}_j}\right) \leq \mathcal{L}_{\hat{\mathcal{D}}_i}(\theta_{\hat{\mathcal{D}}_i}) + \frac{1}{n}\sum_j\left(\frac{1}{2}d(\mathcal{D}_i,\mathcal{D}_j)+\xi_j\right)+\sqrt{\frac{\log\frac{2n}{\delta}}{2m}} \tag{13}$$

E Communication Overhead

To address the communication overhead issue, we exploit the advantage of common representation between clients [3]. Its key insight is to divide the model into two parts: feature extractor and classifier, which is shown as Fig. 3. Indeed, the kernel of this approach is consistent with multi-task learning in conventional centralized machine learning, by exploiting a common representation to share knowledge [4, 5].

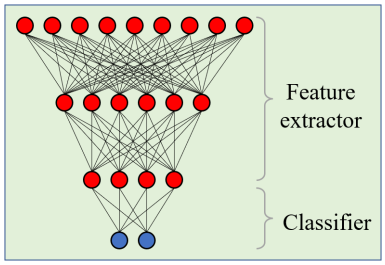

Figure 3: The schematic of common representation for model splitting in our scenario.

According to the latest research [6], they measure the Centered Kernel Alignment (CKA) similarity between the representations from the same layer of different clients' local models, on standard CNN [7]. The observation is clear: comparing different layers in the local models learned on different clients, the similarity of feature extractors among different client local models is very high, while the classifiers have the lowest similarity, which is illustrated in Fig.4.

These result implies that the differences in the common representation part (feature extractor) of each client model are not significant, while the client domain heterogeneity really leads to the difference of the classifiers, and this is exactly what PFL aims at.

Therefore, for the personalization of each client, the most important thing they need to focus on is the classifier of other clients, while the feature extractor part can be shared. Following this insight,

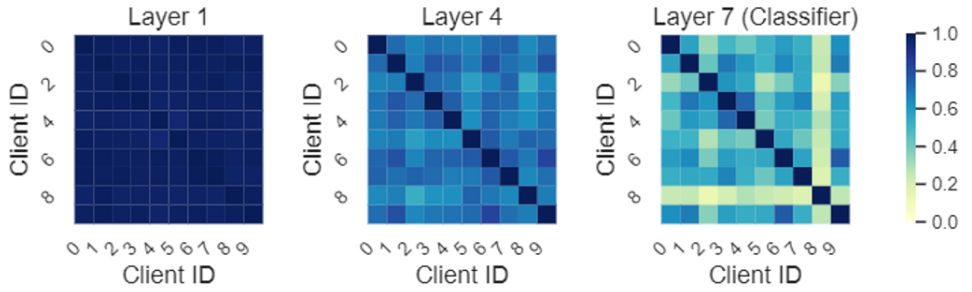

Figure 4: The visualization of the CKA similarity on different layers of different client model pair.

each client only needs to download one global shared feature extractor and several classifiers of other clients to reduce the communication overhead, not the whole model before. And the whole model of other clients can be reconstructed by replacing different classifiers. The modified federated learning architecture is demonstrated in Fig. 5

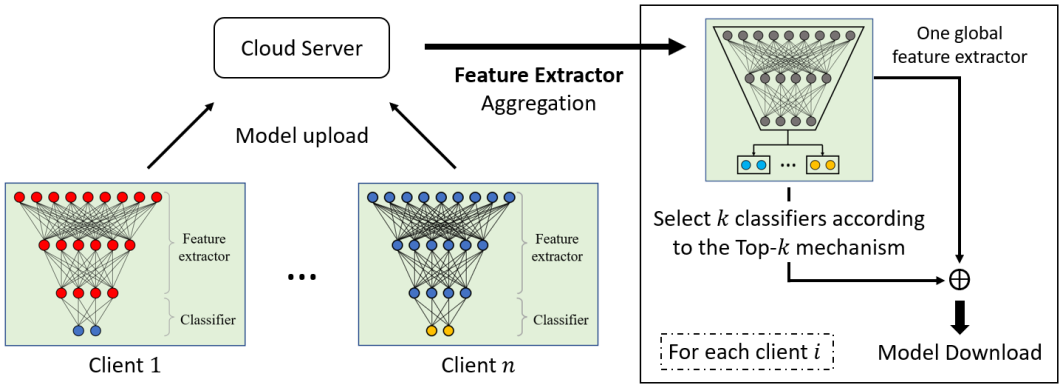

Figure 5: The modified federated learning architecture by exploiting shared feature extractor.

We adopt several baselines compare the communication overhead as follows.

1) **pFedSV (D+C)**: it means we adopt both the **D**ynamic download mechanism and **C**ommon representation in pFedSV to reduce the communication overhead.
2) **pFedSV (D)**: it means we only adopt the **D**ynamic download mechanism in main content to reduce the communication overhead.
3) **FedFomo**: it downloads the whole model of other clients and performs personalization on local-side of each client [8].
4) **FedAMP**: it performs the personalization in the server-side and directly distributes the personalized model to each client, whose communication overhead is equal to FedAvg [9].
5) **FedAvg**: traditional FL algorithm that downloads one global model to each client [10].

All algorithms are implemented with the following setup: total 20 communication rounds, 10 clients with 100% participation in each round and pathological data Non-IID setting.

The extensive experiment results about the communication overhead during the whole FL process on these different algorithms are demonstrated in Fig.6, which is computed based on LeNet-5. We use the number of model parameters that are required in upload and download as the measurement metric.

Beside, to further illustrate the effectiveness of our Top-k dynamic download mechanism and shared common feature extractor in communication overhead reduction, we compute the communication

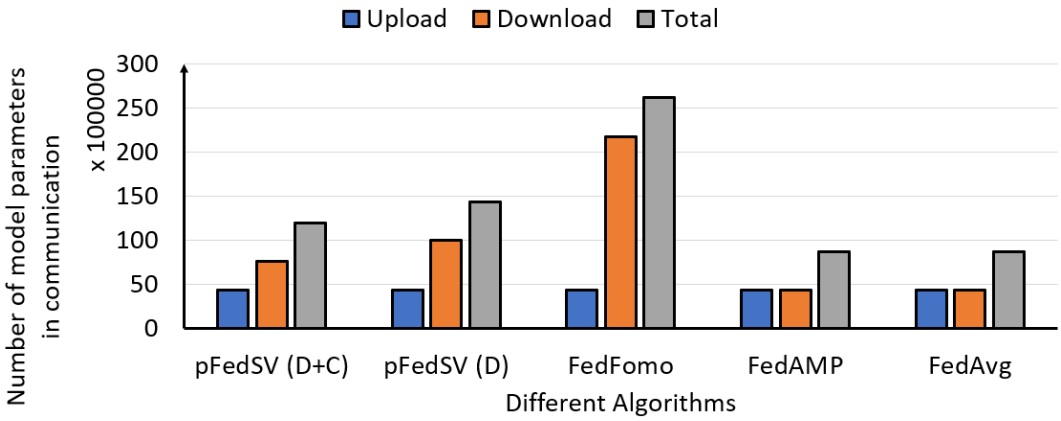

Figure 6: LeNet-5: Communication overhead comparison on LeNet-5 with different algorithms. The y-axis indicates the number of model parameters in the communication.

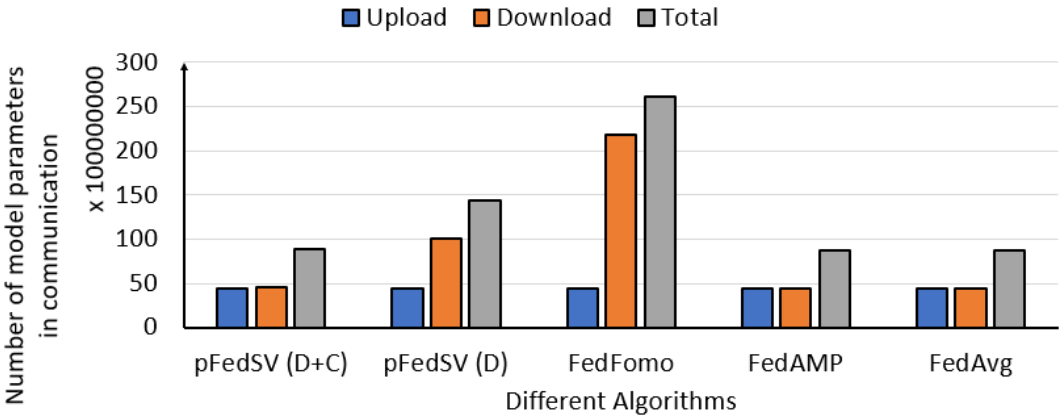

Figure 7: ResNet: Communication overhead comparison on ResNet-V1-34-layer(Plain) with different algorithms. The y-axis indicates the number of model parameters in the communication.

overhead comparison on different models, including ResNet-V1-34-layer(Plain) in Fig. 7 and VGG-19 in Fig. 8.

You can find that the communication overhead of pFedSV at ResNet case is almost the same as traditional FedAvg. The reason is that, as a powerful pre-trained model, The most model parameters in ResNet is the convolutional layer-based feature extractor, and the classifier-related parameters only account for 2.3% of the overall model parameter number. However, in traditional CNN such as LeNet-5, the classifier-related parameters can account for 49.57% of the overall model parameter number. Therefore, with the help of shared feature extractor, the additional communication can be significantly reduced in ResNet case. Moreover, the results on VGG-19 are for your additional reference, where the classifier-related parameters can account for 89.74% of the overall model parameter number in VGG-19.

As expected, the communication overhead of FedFomo is much higher than other algorithms. Our dynamic download mechanism can efficiently reduce it by rapidly identifying the domain-relevant clients and adjusting the model download number, which is illustrated in the main content. Besides, the introduced common representation can further reduce the communication overhead in the download part. Finally, FedAMP has the equal communication overhead as FedAvg. Although our pFedSV (D+C) is not the lowest compared to FedAMP, we can achieve higher personalized performance for each client, which is an acceptable trade-off.

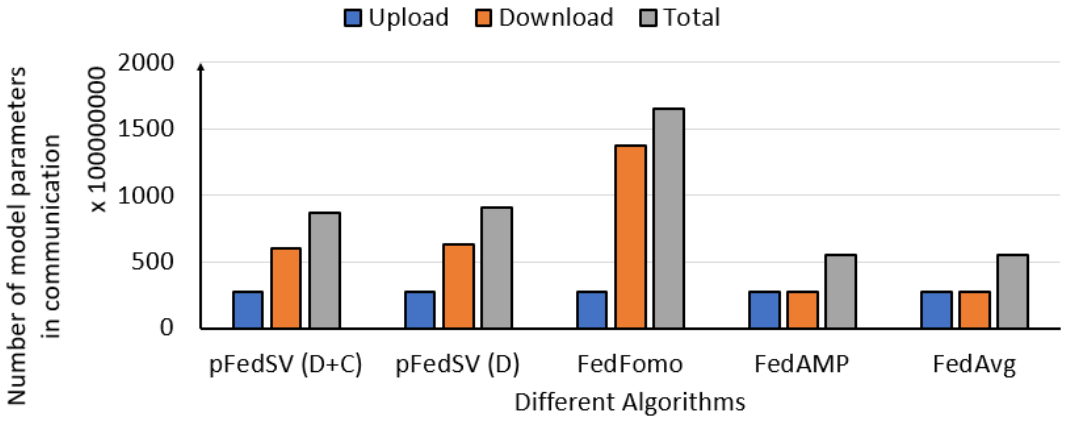

Figure 8: VGG-19: Communication overhead comparison on VGG-19 with different algorithms. The y-axis indicates the number of model parameters in the communication.

F Model Privacy

For the issue of model privacy, we have achieved anonymity by removing any information related to the client's identity from the downloaded models during the entire process of pFedSV. Since the model itself may still imply client private data information in these parameters, we design a more effective privacy protection method, by adopting the (ϵ, δ)-differential privacy (DP) to address the privacy issue in our scenario [11]. We add Gaussian noise into the model parameters after client's local training process, which can guarantee the model with DP. An illustration of DP with adding Gaussian noise is shown in Fig.9.

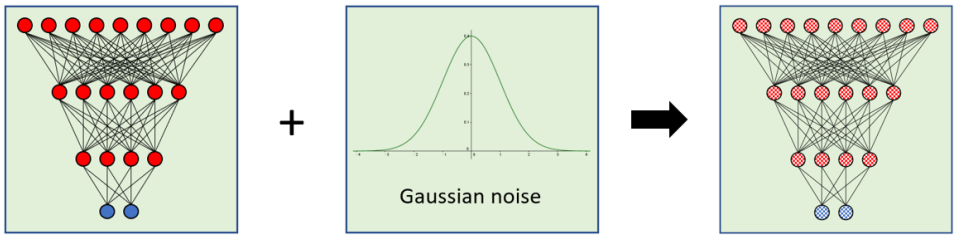

Figure 9: Add Gaussian noise into the model parameters

In brief, DP ensures that, given two nearly identical datasets, querying one dataset produces results with nearly the same probability as querying the other, which is under the control of δ and ϵ. In particular, the DP in our scenario can reduce the connection between the local dataset and the trained model parameters. More noise makes the model more private at the cost of performance and we conduct extensive experiments to illustrate whether pFedSV can retain its performance with more privacy protection (add more noise).

We consider a task with the pathological data Non-IID setting on CIFAR-10 and CIFAR-100 dataset, with 10 clients and 100% participation at each round. We compare pFedSV with the FedAvg under different levels of Gaussian noise σ, and all other parameters are fixed. The results in Table.2 indicates that a higher σ leads to improved privacy (lower ϵ) at the cost of decreased performance (bold in the table). The experiment results in Table 2 of our Appendix have shown that adding aggressive noise will cause accuracy reduction (from 84.73% to 78.29%). However, adopting an appropriate noise ($\delta = 1$) can also protect the model privacy with only a minor impact on accuracy (from 84.73% to 82.16%). We also conduct additional experiments to show that, under an appropriate noise ($\delta = 1$), the performance of our pFedSV+DP can still outperform other personalized baselines, where the results can be found in Table 3.Therefore, the DP-based methods are still effective to solve privacy

Methods	δ	σ	CIFAR-10		CIFAR-100	
			ϵ	Accuracy	ϵ	Accuracy
FedAvg	1×10^{-5}	0	∞	19.68 ± 1.76	∞	5.21 ± 0.41
FedAvg	1×10^{-5}	1	11.28 ± 0.32	17.54 ± 1.37	8.47 ± 0.67	5.03 ± 0.24
FedAvg	1×10^{-5}	2	3.64 ± 0.13	15.97 ± 1.53	2.56 ± 0.19	4.37 ± 0.19
pFedSV	1×10^{-5}	0	∞	84.73 ± 1.67	∞	31.07 ± 1.22
pFedSV	1×10^{-5}	1	$\mathbf{5.97 \pm 0.11}$	$\mathbf{82.16 \pm 1.55}$	$\mathbf{8.42 \pm 0.71}$	$\mathbf{30.59 \pm 1.06}$
pFedSV	1×10^{-5}	2	1.82 ± 0.05	78.29 ± 1.63	1.80 ± 0.16	23.44 ± 0.89

Table 2: The results of pFedSV with DP, which illustrates that we can maintain the personalized accuracy with a reasonable privacy budget.

issues, which are widely validated by many works. In summary, despite adding Gaussian noise into the model parameters will potentially deviate it from the optimal, we can mitigate the privacy violation risk with (ϵ,δ)-differential privacy while maintain an acceptable model performance level.

Methods	MNIST		FMNIST		CIFAR-10	
	10 clients	100 clients	10 clients	100 clients	10 clients	100 clients
Seperate	74.05 ± 2.11	59.81 ± 5.73	60.18 ± 6.42	58.22 ± 6.73	40.53 ± 7.20	36.15 ± 6.88
FedAvg	43.57 ± 3.75	30.15 ± 4.82	40.58 ± 4.16	36.49 ± 5.07	33.81 ± 5.07	26.82 ± 6.43
FedProx	47.49 ± 4.18	44.76 ± 5.49	43.09 ± 4.82	40.34 ± 4.72	35.76 ± 5.18	29.91 ± 5.58
FedAvg+FT	55.72 ± 3.84	50.57 ± 4.26	50.27 ± 4.13	44.83 ± 5.01	42.42 ± 5.29	37.05 ± 5.22
pFedMe	64.39 ± 4.08	58.02 ± 3.51	60.27 ± 3.59	56.81 ± 4.01	50.73 ± 4.29	44.21 ± 5.09
FedFomo	72.54 ± 2.18	63.07 ± 2.54	64.75 ± 3.42	60.49 ± 3.72	53.83 ± 4.57	48.35 ± 5.29
FedAMP	70.15 ± 3.02	60.28 ± 3.11	62.28 ± 2.53	58.94 ± 3.14	51.57 ± 4.03	46.05 ± 4.48
pFedHN	73.35 ± 2.04	62.57 ± 4.11	62.95 ± 3.44	59.55 ± 4.15	52.82 ± 3.88	47.19 ± 5.83
pFedSV(Ours)	$\mathbf{78.17 \pm 1.59}$	$\mathbf{70.76 \pm 2.41}$	$\mathbf{71.47 \pm 1.86}$	$\mathbf{66.63 \pm 2.03}$	$\mathbf{61.18 \pm 1.67}$	$\mathbf{56.76 \pm 1.85}$
pFedSV+DP	$\mathbf{76.58 \pm 1.32}$	$\mathbf{68.43 \pm 1.86}$	$\mathbf{69.24 \pm 2.07}$	$\mathbf{65.31 \pm 1.95}$	$\mathbf{57.94 \pm 2.53}$	$\mathbf{53.28 \pm 1.66}$

Table 3: The MTA with the Dirichlet Non-IID data setting ($\alpha = 0.1$). 10 clients with 100% and 100 clients with 10% participation. We emphasis the results of our pFedSV and the pFedSV+DP by bold.

G Experiment Extension for different Models and Datasets

Our original experiments are conducted on MNIST and Fashion-MNIST datasets with LeNet, CIFAR-10 dataset with AlexNet. To further illustrate the effectiveness of pFedSV on more complex models and tasks, we extend our experiment analysis on CIFAR-100 Dataset with VGG-19 model, where the experiment results are shown in the left part of Table 4. We can find that our pFedSV can still outperform all baselines on more complex dataset and model.

Methods	CIFAR-100		MNIST	FMNIST	CIFAR-10
	10 clients	100 clients	200 clients		
Seperate	35.43 ± 3.87	30.05 ± 5.49	50.37 ± 6.24	48.67 ± 6.49	38.46 ± 6.72
FedAvg	26.17 ± 4.27	20.33 ± 5.27	25.86 ± 7.05	23.81 ± 5.47	24.39 ± 6.28
FedProx	29.62 ± 5.13	23.27 ± 4.69	31.27 ± 6.48	29.44 ± 6.05	26.55 ± 6.13
FedAvg+FT	36.33 ± 3.86	30.55 ± 4.37	36.81 ± 5.37	33.12 ± 6.58	31.07 ± 5.24
pFedMe	40.29 ± 3.57	34.94 ± 3.78	47.53 ± 5.02	45.69 ± 5.82	36.85 ± 5.09
FedFomo	45.91 ± 3.06	37.51 ± 3.09	54.86 ± 4.35	50.36 ± 5.17	41.53 ± 5.47
FedAMP	43.67 ± 3.55	36.40 ± 3.76	52.53 ± 4.19	49.67 ± 5.28	40.28 ± 4.88
pFedHN	45.33 ± 3.45	37.38 ± 3.77	53.49 ± 4.57	50.24 ± 4.86	41.93 ± 5.64
pFedSV(Ours)	$\mathbf{50.46 \pm 2.47}$	$\mathbf{42.25 \pm 3.13}$	$\mathbf{59.22 \pm 3.84}$	$\mathbf{56.72 \pm 5.45}$	$\mathbf{47.61 \pm 4.75}$

Table 4: The left part is experiment extension on complex CIFAR-100 dataset with VGG-19 model. The right part is the experiment extension on larger client scale with total 200 clients and 10% participation in each round. The dataset is Dirichlet Non-IID setting ($\alpha = 0.1$).

Besides, we also further extend our experiment to larger client population (total 200 clients) to demonstrate the effectiveness of pFedSV in practical FL application. The experiment results are shown in the right part of Table 4, where our pFedSV still maintains its best performance on different datasets and modes.

References

[1] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[3] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. *arXiv preprint arXiv:2102.07078*, 2021.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *arXiv preprint arXiv:2106.05001*, 2021.

[7] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[8] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2020.

[9] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7865–7873, 2021.

[10] 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.

[11] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.