FedGP: Correlation-Based Active Client Selection for Heterogeneous Federated Learning

Minxue Tang [1] **Xuefei Ning** [2] **Yitu Wang** [3] **Yu Wang** [4] **Yiran Chen** [5]

Abstract

Client-wise heterogeneity is one of the major issues that hinder effective training in federated learning (FL). Since the data distribution on each client may differ dramatically, the client selection strategy can largely influence the convergence rate of the FL process. Several recent studies adopt active client selection strategies. However, they neglect the loss correlations between the clients and achieve marginal improvement compared to the uniform selection strategy. In this work, we propose FedGP—a federated learning framework built on a correlation-based client selection strategy, to boost the convergence rate of FL. Specifically, we first model the loss correlations between the clients with a Gaussian Process (GP). To make the GP training feasible in the communication-bounded FL process, we develop a GP training method utilizing the historical samples efficiently to reduce the communication cost. Finally, based on the correlations we learned, we derive the client selection with an enlarged reduction of expected global loss in each round. Our experimental results show that compared to the latest active client selection strategy, FedGP can improve the convergence rates by $1.3 \sim 2.3\times$ and $1.2 \sim 1.4\times$ on FMNIST and CIFAR-10, respectively.

1. Introduction

As a newly emerging distributed optimization framework, federated learning (FL) (Konečnỳ et al., 2015; 2016; McMahan et al., 2017; Kairouz et al., 2019) has recently attracted attentions because of its communication efficiency and data privacy. FL aims at dealing with scenarios where training data is distributed on a number of edge devices (i.e., clients). Considering limited communication bandwidth and privacy requirement, in each communication round, FL usually allows only partial participation of the clients and multi-iteration local updating without any exposure of the datasets (McMahan et al., 2017). This special scenario also introduces other challenges that distinguish FL from the normal distributed optimization (Kairouz et al., 2019; Li et al., 2020b).

One major challenge in FL is the high degree of client-wise heterogeneity (Li et al., 2020b), which is the natural characteristic of the large number of clients. There have been many studies (Smith et al., 2017; Li et al., 2018; Liang et al., 2019; Karimireddy et al., 2019; Wang et al., 2020; Reisizadeh et al., 2020; Li et al., 2020a) trying to tackle non-identical and unbalanced dataset distribution as well as distinct hardware capabilities of the clients in FL. Most of them focus on improving the local updating procedure or the central aggregation based on a popular FL algorithm named FedAvg (McMahan et al., 2017). Nevertheless, few studies have explored the possibility of improving the client selection strategy instead of using a uniform selection. In this paper, we show that an appropriate client selection can alleviate the accuracy deterioration caused by the heterogeneity and boost the convergence of FL.

Our idea is mainly based on the following intuitions:

1. Clients are NOT independent. As all clients share one global model, training with the dataset on one client inevitably causes loss changes on other clients.
2. Clients are NOT equivalent. For example, training with the large and diverse dataset on a “good” client can decrease the losses of most clients, while training with the small and extreme dataset on a “bad” client can even cause the losses of other clients to increase.

According to these intuitions, we believe that it is important to utilize the correlations between the clients to properly decide a client selection. In this work, we propose FedGP, an FL framework built on a correlation-based client selection strategy, to boost the convergence rate of FL. Our contributions are:

1. We model the client loss changes with a Gaussian Process (GP) and propose a feasible training method to

[1]minxue.tang@duke.edu [2]foxdoraame@gmail.com [3]yitu.wang@duke.edu [4]yu-wang@mail.tsinghua.edu.cn [5]yiran.chen@duke.edu.

train the GP model with a low communication overhead. We demonstrate that our model can capture the client correlations well using only the loss information.

2. Based on the GP model, we propose a sequential method to derive a client selection with an enlarged reduction of the expected global loss in each round.

3. Experimental results show that FedGP can stabilize the FL process and boost the convergence rates by $1.3 \sim 2.3\times$ and $1.2 \sim 1.4\times$ on FMNIST and CIFAR-10, respectively.

2. Preliminary and Related Work

In FL (Konečnỳ et al., 2015; 2016; McMahan et al., 2017; Kairouz et al., 2019), we seek for a global model $\boldsymbol{\omega}$ that achieves the best performance on all N clients. The global loss function in FL is defined as

$$L(\boldsymbol{\omega}) = \sum_{k=1}^{N} \frac{|\mathcal{D}_k|}{\sum_j |\mathcal{D}_j|} l(\boldsymbol{\omega}; \mathcal{D}_k) = \sum_{k=1}^{N} p_k l_k(\boldsymbol{\omega}) \tag{1}$$

$$l_k(\boldsymbol{\omega}) = l(\boldsymbol{\omega}; \mathcal{D}_k) = \frac{1}{|\mathcal{D}_k|} \sum_{\xi \in \mathcal{D}_k} l(\boldsymbol{\omega}; \xi), \tag{2}$$

where $l(\boldsymbol{\omega}; \xi)$ is the objective loss for sample ξ evaluated on model $\boldsymbol{\omega}$. We refer to $l_k(\boldsymbol{\omega})$ as the local loss of client k, which is evaluated with the local dataset $\mathcal{D}_k$ (of size $|\mathcal{D}_k|$) on client k. The weight $p_k = |\mathcal{D}_k| / \sum_j |\mathcal{D}_j|$ corresponding to the client k is proportional to the size of its local dataset. The goal of FL is to find the optimal model parameters that minimize the global loss:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} L(\boldsymbol{\omega}) = \arg\min_{\boldsymbol{\omega}} \sum_{k=1}^{N} p_k l_k(\boldsymbol{\omega}). \tag{3}$$

There have been many studies trying to solve Eq. 3 under the FL setting (e.g., Li et al., 2018; Liang et al., 2019; Karimireddy et al., 2019; Wang et al., 2020; Cho et al., 2020), and most of them are derived from a standard FL algorithm FedAvg (McMahan et al., 2017). To satisfy the communication constraint, these algorithms usually assume partial client participation and multi-iteration local updates. In particular, in communication round t, only a subset $\mathcal{K}_t$ with size $|\mathcal{K}_t| = C \leq N$ of the client set $\mathcal{U}$ is selected to receive the global model $\boldsymbol{\omega}^t$ and conduct training with their local dataset for several iterations independently. After the local training, the server collects the trained models from these selected clients and average them to produce a new global model $\boldsymbol{\omega}^{t+1}$. We formulate this procedure as follows

$$\boldsymbol{\omega}_k^{t+1} = \boldsymbol{\omega}^t - \eta_t \tilde{\nabla} l_k(\boldsymbol{\omega}^t) \tag{4}$$

$$\boldsymbol{\omega}^{t+1} = \frac{1}{C} \sum_{k \in \mathcal{K}_t} \boldsymbol{\omega}_k^{t+1} \tag{5}$$

$$= \boldsymbol{\omega}^t - \frac{\eta_t}{C} \sum_{k \in \mathcal{K}_t} \tilde{\nabla} l_k(\boldsymbol{\omega}^t), \tag{6}$$

where η_t is the learning rate and $\tilde{\nabla} l_k(\boldsymbol{\omega}^t)$ is the equivalent cumulative gradient (Wang et al., 2020) in the t-th communication round. More specifically, for an arbitrary optimizer on the client k, it produces $\Delta \boldsymbol{\omega}_k^{t,\tau} = -\eta \boldsymbol{d}_k^{t,\tau}$ as the local model update at the τ-th iteration in this round, and the cumulative gradient is calculated as

$$\tilde{\nabla} l_k(\boldsymbol{\omega}^t) = \sum_{\tau} \boldsymbol{d}_k^{t,\tau}. \tag{7}$$

A main challenge that FL faces is to overcome the client-wise heterogeneity, especially non-identical and unbalanced data distribution. There have been some studies (Li et al., 2018; Liang et al., 2019; Karimireddy et al., 2019; Wang et al., 2020) that focus on improving the local update procedure, i.e., amending the calculation of the equivalent cumulative gradient $\tilde{\nabla} l_k$ during local training to achieve a better performance. However, few studies have tried improving the performance in the heterogeneous setting through an active client selection, and most of the previous studies sample $\mathcal{K}_t$ uniformly. Ribero & Vikalo (2020) model the local training process of each client as an Ornstein-Uhlenbeck process and use a threshold to determine whether a client needs to upload its model to the server. Yang et al. (2019) analyze the convergence rate of different scheduling policies in wireless FL. However, these studies only consider identical data distribution. To improve the convergence rate under heterogeneous data settings, Goetz et al. (2019) assign a high selection probability to those clients with large local losses, and Cho et al. (2020) select C clients with the largest losses among a randomly sampled subset $\mathcal{A} \subseteq \mathcal{U}$ with size $d > C$. In these studies, client losses are modeled independently, and the selection criterion neglects the correlations between the clients. In contrast, our work considers the correlations between the clients and derives a client selection that can enlarge the reduction of the expected global loss.

3. Methodology

In this section, we describe our method FedGP that aims at boosting the convergence of the FL. We first formulate our goal under the FL setting as an optimization problem in Section 3.1. Then in Section 3.2, we model the correlations between loss changes of clients using a Gaussian Process. Then, we give a practical method to train the Gaussian Process in the FL setting in Section 3.3. Finally in Section 3.4, we exploit the GP model to solve the optimization problem and get a client selection in each round. The whole framework is summarized in Algorithm 2.

3.1. Problem Formulation

We seek to find the optimal client selection to achieve the fastest convergence rate. In other words, we aim to achieve the maximal decrease of the global loss defined in Eq. 1 after each communication round. Therefore, we define our target as solving a series of optimization problems, one for each communication round t as:

$$\begin{aligned} \min_{\mathcal{K}_t} \quad & \Delta L^t = L(\boldsymbol{\omega}^{t+1}) - L(\boldsymbol{\omega}^t) \\ \text{subject to} \quad & \boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \frac{\eta_t}{C} \sum_{k \in \mathcal{K}_t} \tilde{\nabla} l_k(\boldsymbol{\omega}^t). \end{aligned} \tag{8}$$

We cannot measure the loss changes before we actually decide a client selection, thus we need to learn a prediction model to predict the loss changes of possible client selections. In the following sections, we will show how we predict the loss changes and find a client selection with an enlarged reduction of expected global loss.

3.2. Modeling Loss Changes with Gaussian Processes

To find a client selection that accelerates the FL training process, we introduce a Gaussian Process (GP) model to model the loss changes of clients in each round. GP has been widely used in active learning due to its strong ability to handle correlations between sample points (see, e.g., Seo et al., 2000; Kapoor et al., 2007; Gotovos, 2013; Shahriari et al., 2015). Here, we make two assumptions under which we can model the loss changes with a GP.

Assumption 3.1 *In any communication round t, if we uniformly sample client selection $\mathcal{K}_t \sim U\big(\{\mathcal{K} \subseteq \mathcal{U} : |\mathcal{K}| = C\}\big)$, the global model update $\Delta\boldsymbol{\omega}^t = \boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^t$ will follow a Gaussian Distribution*

$$\Delta\boldsymbol{\omega}^t \sim \mathcal{N}(-\eta_t \tilde{\boldsymbol{g}}^t, \frac{\eta_t^2 BB^T}{C}), \tag{9}$$

where $\tilde{\boldsymbol{g}}^t = \mathbb{E}_k[\tilde{\nabla} l_k(\boldsymbol{\omega}^t)]$ is the mean cumulative gradient of all the clients in $\mathcal{U}$, and B is a constant matrix.

Assumption 3.2 *In any communication round t, the global model update $\Delta\boldsymbol{\omega}^t = \boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^t$ is always small enough that we can approximate the loss change Δl_k^t of any client k as a linear function of $\Delta\boldsymbol{\omega}^t$*

$$\begin{aligned} \Delta l_k^t &= {\boldsymbol{g}_k^t}^T \Delta\boldsymbol{\omega}^t \\ \boldsymbol{g}_k^t &= \nabla l_k(\boldsymbol{\omega}^t). \end{aligned} \tag{10}$$

Assumption 3.1 is inspired by the work (Mandt et al., 2016) who assumes the stochastic gradients in SGD are Gaussian, and therefore the parameter update after one iteration follows a Gaussian Distribution. Note that in the FL procedure, the form in Eq. 6 is very similar to that in the SGD update. The only difference is that the average gradients within one mini-batch is replaced by the average cumulative gradients of the selected clients. Therefore, it is reasonable to make this assumption similar to (Mandt et al., 2016). Under the above two assumptions, we can model the loss changes of all the clients with a GP as depicted in Lemma 3.1.

Lemma 3.1 *In any communication round t, if we uniformly sample client selection $\mathcal{K}_t \sim U\big(\{\mathcal{K} \subseteq \mathcal{U} : |\mathcal{K}| = C\}\big)$, for any subset $\mathcal{S} \subseteq \mathcal{U}$, the loss changes $\Delta\boldsymbol{l}_{\mathcal{S}}^t = [\Delta l_{i_1}^t, \cdots, \Delta l_{i_{|\mathcal{S}|}}^t]^T, i_1, \cdots, i_{|\mathcal{S}|} \in \mathcal{S}$ will follow a Multivariate Gaussian Distribution*

$$\Delta\boldsymbol{l}_{\mathcal{S}}^t \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{S}}^t, \Sigma_{\mathcal{S}}^t) \tag{11}$$

$$\boldsymbol{\mu}_{\mathcal{S}}^t = -\eta_t {G_{\mathcal{S}}^t}^T \tilde{\boldsymbol{g}}^t \tag{12}$$

$$\Sigma_{\mathcal{S}}^t = \frac{\eta_t^2}{C} {G_{\mathcal{S}}^t}^T BB^T G_{\mathcal{S}}^t \tag{13}$$

$$G_{\mathcal{S}}^t = \left[\boldsymbol{g}_{i_1}^t, \cdots, \boldsymbol{g}_{i_{|\mathcal{S}|}}^t\right]. \tag{14}$$

We remove the subscript $\mathcal{S}$ to simplify the corresponding representation for the client set $\mathcal{U}$ as

$$\Delta\boldsymbol{l}^t \sim \mathcal{N}(\boldsymbol{\mu}^t, \Sigma^t). \tag{15}$$

Now that we have introduced a GP to model the loss changes, the next step is to determine the GP parameters $\boldsymbol{\mu}^t, \Sigma^t$. In the next section, we describe our method of GP training in the communication-bounded FL process.

3.3. Training Gaussian Processes in FL

As a classical machine learning model, GP has been widely discussed and well studied (Williams & Rasmussen, 2006). There have been many methods to train the parameters in GP, namely, the covariance Σ^t in Eq. 15[1]. Nevertheless, to make the GP training feasible in the communication-bounded FL procedure, we must revise the GP training method to better utilize historical data.

In GP, a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is used to calculate the covariance (Williams & Rasmussen, 2006) as

$$\Sigma^t(i,j) = K(\boldsymbol{x}_i^t, \boldsymbol{x}_j^t), \tag{16}$$

where $\boldsymbol{x}_i^t, \boldsymbol{x}_j^t$ are the features of the data points i and j, respectively. Note that in FL, we only have the indexes of each client, which are not suitable to be used as the input features. Thus, we assign a trainable embedding in a latent

[1] We do not train $\boldsymbol{\mu}$ and set it to 0. We will show that the value of $\boldsymbol{\mu}$ does not influence the selection decision in Section 3.4.

space to each client. The embedding of the k-th client is noted as $\boldsymbol{x}_k^t \in \mathbb{R}^d$, and we can write the kernel function as

$$K(\boldsymbol{x}_i^t, \boldsymbol{x}_j^t) = {\boldsymbol{x}_i^t}^T \boldsymbol{x}_j^t, \tag{17}$$

which is a homogeneous linear kernel (Williams & Rasmussen, 2006). The homogeneous linear kernel is an intuitive choice according to Eq. 13, where $\Sigma_{ij}^t = {\boldsymbol{x}_i^t}^T \boldsymbol{x}_j^t \propto {\boldsymbol{g}_i^t}^T BB^T \boldsymbol{g}_j^t$ by setting $\boldsymbol{x}_k^t \propto B^T \boldsymbol{g}_k^t, \forall k \in \mathcal{U}$.

Then, we maximize the likelihood of observing $\Delta \boldsymbol{l}^{t,i} \sim \mathcal{N}(0, {X^t}^T X^t)$ to learn the embedding matrix X^t:

$$X^t = \arg\max_{X^t} \sum_i \log p(\Delta \boldsymbol{l}^{t,i} | X^t) \tag{18}$$

where $X^t = [x_1^t, \cdots, x_N^t]$ is the embedding matrix and $\Delta \boldsymbol{l}^{t,i} = [\Delta l_1^{t,i}, \cdots, \Delta l_N^{t,i}]^T$ is the observed loss changes using a certain client selection $\mathcal{K}_i$. To get the loss changes $\Delta \boldsymbol{l}^{t,i}$ using one client selection $\mathcal{K}_i$ and evaluate the likelihood, we have to broadcast the current and updated model to all the clients in $\mathcal{U}$. And since multiple client selections are needed for an unbiased estimation of the parameters in GP, the vanilla GP training process in Eq. 18 introduces a high communication overhead.

Actually, the correlations between loss changes of different clients mainly arise from similarities between their datasets, which are invariant during the FL process. This inspires us to develop a practical training method for the GP model to reduce the communication overhead. We assume that the covariance Σ^t keeps approximately invariant in the concerned time range so that we can reuse historical samples for GP training. Furthermore, we don't need to update X^t in every round. Instead, we inherit the embedding matrix X^{t-1} from the last round and train it only every $\Delta t > 1$ rounds. In each GP update round, we randomly sample one client selection $\mathcal{K}^r$, and update X^t by maximizing the likelihood. The update of X^t in every FL round can be written as

$$X^t = \begin{cases} X^{t-1}, & t\%\Delta t \neq 0 \\ \arg\max_X \sum_{m=0}^{M} \gamma^m \log p(\Delta \boldsymbol{l}^{t-m\Delta t} | X), & t\%\Delta t = 0, \end{cases} \tag{19}$$

where M is the number of reused historical samples. $\gamma < 1$ is the discount factor to weight the samples from the past round. Using our GP training method, we are able to reduce the communication overhead to only broadcasting the model to all the clients in $\mathcal{U}$ every Δt rounds.

3.4. Client Selection Strategy

In this section, we show how we use the GP to predict and make the selection decision. We develop a sequential method to efficiently derive a solution of the combinatorial optimization problem in Eq. 8 that has a vast solution space ($\sim \mathcal{O}(N^C)$).

We first rewrite the optimization target in Eq.8 as a conditioned expectation form:

$$\min_{\mathcal{K}_t} \mathbb{E}_{\Delta \boldsymbol{\omega}^t | \mathcal{K}_t} \Big[\sum_i p_i \Delta l_i^t \Big] \tag{20}$$

$$\approx \min_{\mathcal{K}_t} \mathbb{E}_{\Delta \boldsymbol{\omega}^t | \Delta \boldsymbol{l}_{\mathcal{K}_t}^t} \Big[\sum_i p_i \Delta l_i^t \Big] \tag{21}$$

$$= \min_{\mathcal{K}_t} \sum_i p_i \tilde{\mu}_i^t (\Delta \boldsymbol{l}_{\mathcal{K}_t}^t), \tag{22}$$

where $\tilde{\boldsymbol{\mu}}^t(\Delta \boldsymbol{l}_{\mathcal{K}_t}^t)$ is the mean of the posterior conditioned on the loss changes $\Delta \boldsymbol{l}_{\mathcal{K}_t}^t$ of the selected client set $\mathcal{K}_t$. Note that we make the approximation in Eq. 21 by replacing the condition $\mathcal{K}_t$ with $\Delta \boldsymbol{l}_{\mathcal{K}_t}^t$. This approximation holds if the loss changes on $\mathcal{K}_t$ ($\Delta \boldsymbol{l}_{\mathcal{K}_t}^t$) could only be achieved by the client selection $\mathcal{K}_t$. We anticipate that this is usually the case since that the losses of a client group usually decrease the most when they are included in the client selection. Hence, no client selections other than $\mathcal{K}_t$ can achieve the same loss decreases on $\mathcal{K}_t$.

To efficiently solve the optimization problem in Eq. 22, we develop a sequential method that selects the clients one by one. In each selection step, the client is selected to greedily maximize the posterior expectation of the overall loss decrease. More specifically, according to the intuition that the selected client would have a larger loss decrease, we first assume that the loss changes of the selected clients would be lower than the means of the GP. And the excess of the loss changes over the GP mean is related to the variance to some extent

$$\Delta \hat{l}_k^t = \mu_k^t - \alpha_k^t \sigma_k^t \tag{23}$$

$$\alpha_k^t = a \beta^{\tau_k^t}, \tag{24}$$

where $\sigma_k^t = \sqrt{\Sigma^t(k,k)}$, and a is a scale constant. $\beta \in (0,1)$ is an annealing coefficient, and its index τ_k^t denotes how many times that client k has been selected after the last GP update. Then, after selecting one client, we calculate the GP posterior distribution conditioned on this selected client, and utilize the posterior distribution for the following selection steps.

The selection strategy is summarized in Algorithm 1, and Figure. 1 illustrates a toy example of the selection procedure. Our overall FL framework FedGP is summarized in Algorithm 2. It is noteworthy that our method is orthogonal to any existing FL optimizer, e.g., FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2018), so it can be easily combined with any of them.

Insights into the Selection Strategy: Single-Step For simplicity, we omit the superscript t in the following dis-

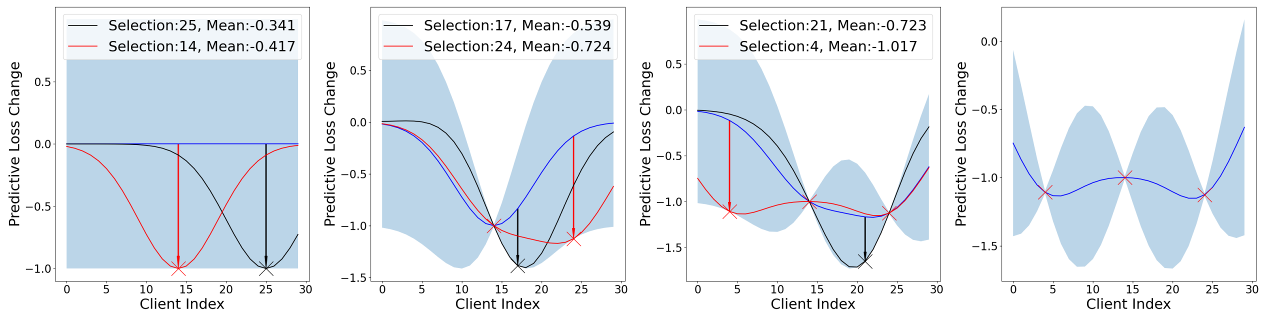

Figure 1. A toy example of our selection procedure where $\Sigma(i,j) \propto e^{(i-j)^2/25}$. From left to right, we sequentially select 3 clients from 30 clients in total. Blue lines and shades refer to the mean and standard deviation of the GP, using which we will predict the loss changes and select a new client. The red and black lines are the posterior mean conditioned on the predictive loss changes (marked as the crosses, lower than the prior mean shown by the arrows) of two candidates, while the red one is the optimal selection. The values of the selection criterion in Eq. 22 of the candidate clients are shown in the legends. After each selection step, we update the mean and covariance of the GP with the posterior conditioned on the selected client. The last figure shows the final client selection (red crosses) and its related predictive loss changes (blue line).

Algorithm 1 Client Selection Strategy with GP

Require: $\boldsymbol{\mu}^t$ and Σ^t of the GP, scale factor $\boldsymbol{\alpha}^t$
Ensure: Client Selection $\mathcal{K}_t$

Initialize $\mathcal{K}_t \leftarrow \emptyset, \mathcal{S} \leftarrow U$.
while $|\mathcal{K}_t| < C$ **do**
  Predict loss change for each client in S as $\Delta\hat{l}_k^t = \mu_k^t - \alpha_k^t \sigma_k^t$, where $\sigma_k^t = \sqrt{\Sigma^t(k,k)}$.
  Calculate the posterior mean $\tilde{\boldsymbol{\mu}}^t(\Delta\hat{l}_k^t)$ and select the client $k^* = \arg\min_k \sum_i p_i \tilde{\mu}_i^t(\Delta\hat{l}_k^t)$.
  Add k^* into $\mathcal{K}_t$ and remove it from S.
  $\boldsymbol{\mu}^t \leftarrow \tilde{\boldsymbol{\mu}}^t(\Delta\hat{l}_{k^*}^t), \Sigma^t \leftarrow \tilde{\Sigma}^t(\Delta\hat{l}_{k^*}^t)$.
end while

Algorithm 2 FedGP

Initialize X and Global Model $\boldsymbol{\omega}$.
for each round $t = 0, 1, \ldots$ **do**
  if $t\%\Delta t$==0 **then**
    Uniformly sample clients $\mathcal{K}^r$.
    Train $\mathcal{K}^r$ with the FL optimizer and collect the loss change $\Delta \boldsymbol{l}^t$.
    Reset $\alpha_k \leftarrow 1, \forall k \in \mathcal{U}$.
  end if
  Update X with Eq. 19.
  Select clients $\mathcal{K}_t$ with Algorithm 1.
  Train $\mathcal{K}_t$ and update $\boldsymbol{\omega}$ with the FL optimizer.
  Update $\boldsymbol{\alpha}_{\mathcal{K}_t} \leftarrow \beta \boldsymbol{\alpha}_{\mathcal{K}_t}$.
end for

cussions. By substituting the loss change guess $\Delta\hat{l}_k$ from Eq. 23 into the posterior expression of the GP, we can calculate the weighted sum of the posterior mean as

$$\sum_i p_i \tilde{\mu}_i(\Delta\hat{l}_k) \tag{25}$$

$$= \sum_i p_i \mu_i + \sum_i p_i \frac{\Sigma(i,k)}{\sigma_k^2}(\Delta\hat{l}_k - \mu_k) \tag{26}$$

$$= \sum_i p_i \mu_i - a\beta^{\tau_k} \sum_i p_i \sigma_i r_{ik}, \tag{27}$$

where $\Sigma(i,k) = \sigma_i \sigma_k r_{ik}$, and $r_{ik} \in [-1,1]$ is the correlation coefficient measuring the loss correlation between clients i and k. The first item in Eq. 27 and the factor a are constant for all k, thus the selection strategy becomes

$$k^* = \arg\max_k \beta^{\tau_k} \sum_i p_i \sigma_i r_{ik}. \tag{28}$$

Eq. 28 has a clear interpretation that it tends to select the client that has large correlations with other clients (r_{ik}), so that other clients can benefit more from training on the selected client. Our selection criterion takes the correlations between clients into consideration, and can conduct better selection compared with those algorithms that only consider the loss of each client independently (Goetz et al., 2019; Cho et al., 2020).

As we only update the covariance Σ every Δt rounds, the annealing factor β^{τ_k} can prevent us from making the same selection during the Δt rounds. Repeatedly training with the same group of clients would cause the training process to get trapped in a local optimum, and therefore their loss decrease would vanish.

Insights into the Selection Strategy: Multi-Step Here, we analyze the expression of the multi-step criterion and demonstrate that the sequential selection strategy encourages diversity while selecting clients. The posterior covariance conditioned on one selected client k can be written

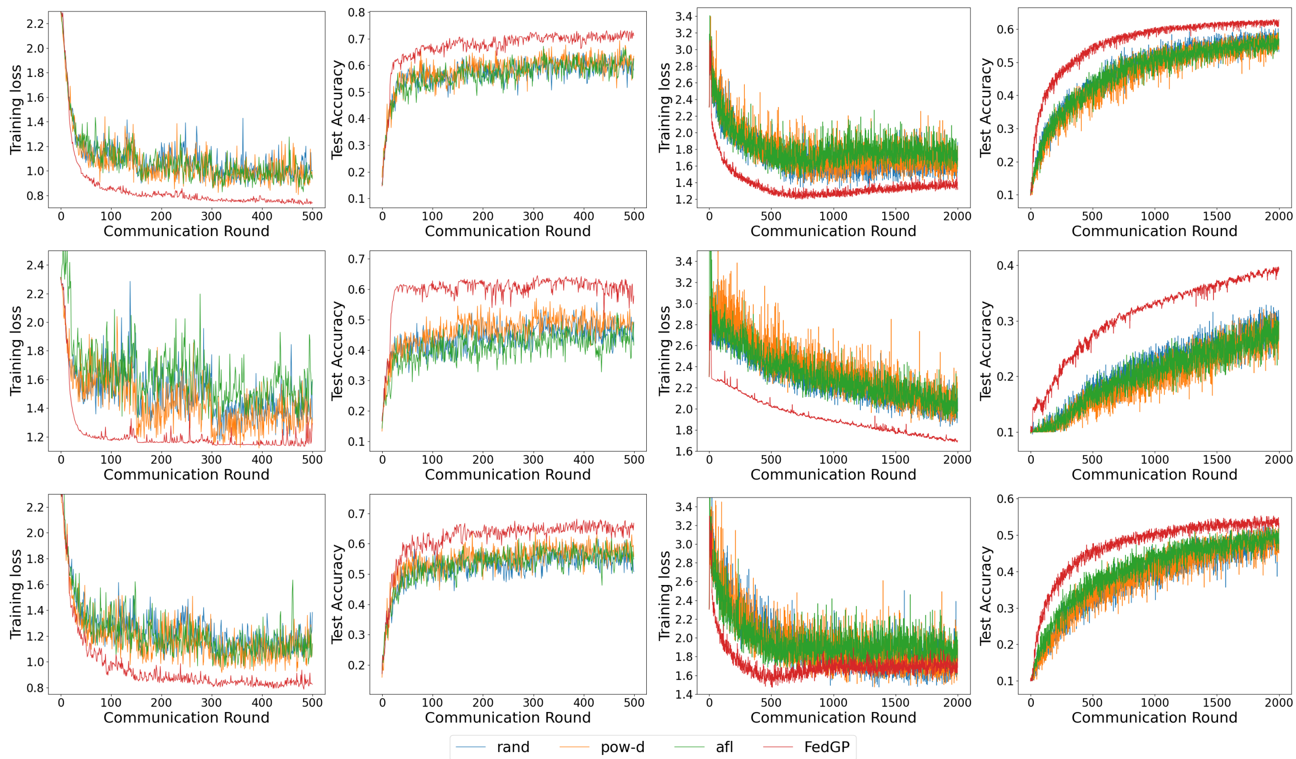

Figure 2. Global loss and test accuracy on FMNIST (left two columns) and CIFAR-10 (right two columns) under three heterogeneous settings (top: 2SPC; median: 1SPC; bottom: Dir). All experiments in one figure share the same hyperparameters except for the method of client selection. In some experiments on CIFAR-10, the slight loss increase in later training rounds is due to the weight decay.

as

$$\tilde{\Sigma}(i,j) = \Sigma(i,j) - \frac{\Sigma(i,k)\Sigma(k,j)}{\sigma_k^2} \tag{29}$$

$$= \sigma_i \sigma_j (r_{ij} - r_{ik} r_{kj}) \tag{30}$$

$$\tilde{\sigma}_i = \sqrt{\tilde{\Sigma}(i,i)} \tag{31}$$

$$= \sigma_i \sqrt{1 - r_{ik}^2}. \tag{32}$$

Under the condition of selecting client k, the selection criterion in Eq. 28 for the next client k' becomes

$$\beta^{\tau_{k'}} \sum_i p_i \frac{\tilde{\Sigma}(i,k')}{\tilde{\sigma}_{k'}} \tag{33}$$

$$= \frac{\beta^{\tau_{k'}}}{\sqrt{1 - r_{k'k}^2}} \Big[\underbrace{\sum_i p_i r_{ik'} \sigma_i}_{\text{(A)}} - r_{kk'} \underbrace{\sum_i p_i r_{ik} \sigma_i}_{\text{(B)}} \Big]. \tag{34}$$

The first term (A) in Eq. 34 is the prior selection criterion without the condition, and the second term (B) is exactly the selection criterion of client k. Since we have maximized (B) in Eq. 28 when selecting client k, this term is usually positive. Therefore, besides the property we have discussed in the single-step selection (large correlations with other clients $r_{ik'}$), the clients that have small correlations $r_{kk'}$ with the previous selected client k are preferred to maximize Eq. 34. This criterion penalizes selection redundancy and leads to a client selection with diverse datasets, which makes the training process more stable and robust. Since clients with similar datasets generate similar local updates, selecting redundant clients only brings marginal gains to the global model performance or even drives the optimization into local optimum.

4. Experiments

4.1. Experiment Settings

We conduct experiments on two datasets, FMNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky et al., 2009). For FMNIST, we adopt an MLP model with two hidden layers. For CIFAR-10, we adopt a CNN model with three convolutional layers followed by one fully connected layer. More details on the model construction and training hyperparameters can be found in the Appendix A.1. For each dataset, we experiment with three different heterogeneous data partition on $N = 100$ clients as follow:

- **2 shards per client (2SPC)**: This setting is the same as the non-IID setting in (McMahan et al., 2017). We

Table 1. The number of communication rounds for each selection strategy to achieve specified test accuracies under three heterogeneous settings. The values in the parentheses in the second row are the specified test accuracy, e.g., 2SPC(0.69) means that the specified accuracy in the heterogeneous setting 2SPC is 69%. The results consist of the mean and the standard deviation over 5 random seeds. N/A means that the corresponding selection strategy cannot achieve the specified accuracy with some random seeds within the maximal number of communication rounds (500 for FMNIST and 2000 for CIFAR-10).

Method	FMNIST			CIFAR-10		
	2SPC(0.69)	1SPC(0.62)	Dir(0.64)	2SPC(0.62)	1SPC(0.36)	Dir(0.53)
Rand	295.8 ± 92.0	N/A	141.0 ± 73.0	1561.2 ± 236.2	1750.4 ± 190.3	1386.4 ± 350.1
AFL	218.6 ± 117.3	N/A	169.0 ± 166.1	N/A	1845.2 ± 28.8	1334.4 ± 277.5
Pow-d	126.6 ± 78.2	167.2 ± 72.3	123.0 ± 101.0	1558.2 ± 227.0	1752.2 ± 186.2	1254.2 ± 119.1
FedGP (Ours)	$\mathbf{94.8 \pm 18.4}$	$\mathbf{84.0 \pm 53.1}$	$\mathbf{68.8 \pm 27.5}$	$\mathbf{1309.0 \pm 196.6}$	$\mathbf{1265.6 \pm 40.5}$	$\mathbf{1024.6 \pm 261.5}$

sort the data by their labels and divide them into 200 shards so that all the data in one shard share the same label. We randomly allocate these shards to clients, and each client has two shards. Since all the shards have the same size, the data partition is balanced. That is to say, all the clients have the same dataset size.

- **1 shard per client (1SPC)**: This setting is similar to the 2SPC setting, and the only difference is that each client only has one shard, i.e., each client only has the data of one label. This is the data partition with the highest heterogeneity, and it is also balanced.
- **Dirichlet Distribution with $\alpha = 0.2$ (Dir)**: We inherit and slightly change the setting from (Hsu et al., 2019) to create an unbalanced data partition. We sample the ratio of the data with each label on one client from a Dirichlet Distribution parameterized by the concentration parameter $\alpha = 0.2$. More details can be found in the Appendix A.2.

We divide the training process of FedGP into two phases:

1. Warm-up phase: We uniformly sample client selection $\mathcal{K}^t$ and collect the loss values of all the clients in $\mathcal{U}$ to train the GP in each round, i.e., $\Delta t = 1$. We set the length of the warm-up phase to 15 in all experiments.
2. Normal phase: After the warm-up phase, we follow Algorithm 2 to select clients and update the GP model.

In all the experiments, we use FedAvg (McMahan et al., 2017) as the FL optimizer. We present the average results using five random seeds in all figures. We will first show that our method can achieve faster and more stable convergence, compared with three baselines: random selection (Rand), Active FL (AFL) (Goetz et al., 2019) and Power-of-choice Selection Strategy (Pow-d) (Cho et al., 2020). Then, we will give an ablation study on the GP training interval Δt to show the stability of the correlations between clients. Finally, we visualize the client embeddings X with t-SNE (Maaten & Hinton, 2008) and show that FedGP can effectively capture the correlations between clients.

4.2. Convergence Rate under Various Heterogeneous Settings

We compare the convergence rate of our method FedGP with the other baselines on both FMNIST and CIFAR-10, and demonstrate the results in Figure 2. We set the GP update interval $\Delta t = 10$ and the annealing coefficient $\beta = 0.95$ for FMNIST experiments, and $\Delta t = 50$ and $\beta = 0.97$ for CIFAR-10 experiments.

As shown in Figure 2, while other biased client selection methods show only slight or even no superiority compared with the fully random selection, our method outperforms all baselines consistently, especially in the extremely heterogeneous setting when data on each client contains only one label (1SPC). We can see from the loss curves that FedGP achieves faster convergence. Also, FedGP achieves higher test accuracy during the whole training process, as illustrated by the accuracy curves. Furthermore, the learning curves of FedGP are more smooth and less noisy than those of other methods, meaning that FedGP reduces the variance and makes the federated optimization more stable.

Table 1 shows the numbers of communication rounds for each selection strategy to achieve a specified test accuracy. We can see that FedGP can achieve the given test accuracy within minimal numbers of communication rounds under all experiment settings.

4.3. Results with Larger Training Interval

Collecting training data in the GP update rounds brings communication overhead, since we need to broadcast the model to all the clients. Thus, it is important to investigate the minimal GP update frequency. We vary the GP training interval and show the accuracy curves in Figure 3. We set $\Delta t = 10, 20, 500$ with $\beta = 0.95, 0.95, 0.99$ for the experiments on FMNIST, and $\Delta t = 50, 100, 2000$ with $\beta = 0.97, 0.97, 0.999$ for the experiments on CIFAR-10, respectively. As shown in the figures, the performance degrades very slightly with larger training intervals. It is noteworthy that even if we do not update the GP model after the warm-up phase (noted as $\Delta t = 500$ for FMNIST, and

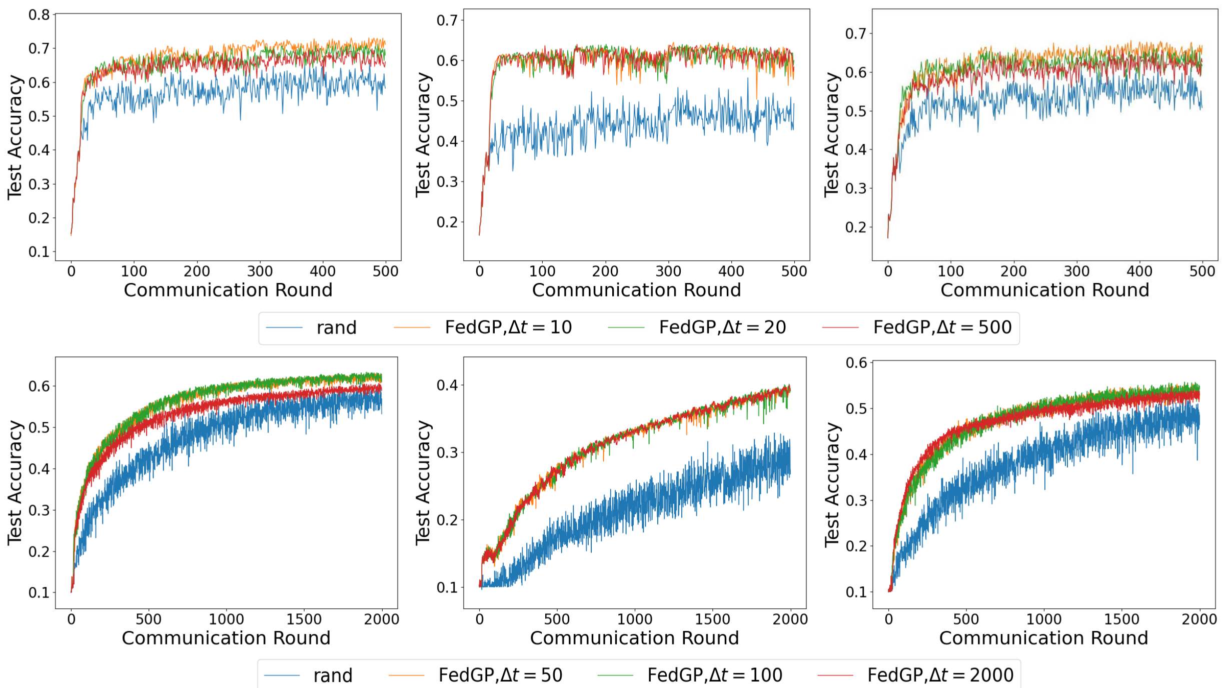

Figure 3. Test accuracy with different GP training interval Δt on FMNIST (top) and CIFAR-10 (bottom) under three heterogeneous settings (left: 2SPC; median: 1SPC; right: Dir).

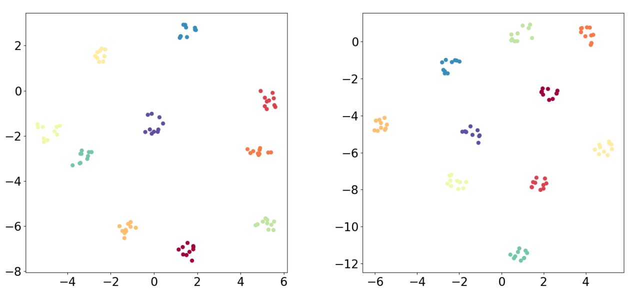

Figure 4. Visualization of client embedding (left: FMNIST; right: CIFAR-10). Each point represents a client, and the color of the point represents the label of the data on this client. We normalize the vector length to 1 so that the distance between two points can reveal the correlation.

$\Delta t = 2000$ for CIFAR-10), FedGP still achieves faster convergence than the random selection strategy. This indicates that the correlations learned by the GP model are stable, which supports our assumptions in Section 3. Therefore, one only needs to train the correlation model in the warm-up phase, which can reduce the communication overhead.

4.4. Visualization of Client Embedding

To obtain an insight into the correlations learned by the GP model, we show the t-SNE (Maaten & Hinton, 2008) plot of the normalized client embeddings in Figure 4. We plot the client embeddings learned in the warm-up phase under the 1SPC setting, and each embedding can be labeled using the only data label on the corresponding client. We see that after the warm-up phase, the GP model has already captured the correlations between the clients, and the embeddings of clients with the same data label are clustered together. This indicates that besides accelerating the FL process, FedGP is an effective tool to learn the correlations between clients without accessing their private data.

5. Conclusion

In this work, we propose FedGP, a framework with a novel client selection strategy for heterogeneous FL. We reveal that it is crucial to utilize the correlations between clients to achieve a faster and more stable convergence in heterogeneous FL. We first model the client correlations with a Gaussian Process (GP), and then develop a practical method to train the GP model with a low communication overhead. Finally, we design an efficient client selection strategy utilizing the GP model, and we give a mathematical interpretation of the selection criterion. Experimental results on FMNIST and CIFAR-10 show that FedGP effectively accelerates and stabilizes the training process under highly heterogeneous settings. In addition, we examine that FedGP captures the client correlation well using only the loss information. How to further utilize the captured correlations is an interesting direction for future work.

References

Cho, Y. J., Wang, J., and Joshi, G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

Goetz, J., Malik, K., Bui, D., Moon, S., Liu, H., and Kumar, A. Active federated learning. *arXiv preprint arXiv:1909.12641*, 2019.

Gotovos, A. Active learning for level set estimation. Master's thesis, Eidgenössische Technische Hochschule Zürich, Department of Computer Science,, 2013.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8. IEEE, 2007.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Konečnỳ, J., McMahan, B., and Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, A., Sun, J., Wang, B., Duan, L., Li, S., Chen, Y., and Li, H. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020a.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020b.

Liang, X., Shen, S., Liu, J., Pan, Z., Chen, E., and Cheng, Y. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Mandt, S., Hoffman, M., and Blei, D. A variational analysis of stochastic gradient algorithms. In *International conference on machine learning*, pp. 354–363, 2016.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Reisizadeh, A., Farnia, F., Pedarsani, R., and Jadbabaie, A. Robust federated learning: The case of affine distribution shifts. *arXiv preprint arXiv:2006.08907*, 2020.

Ribero, M. and Vikalo, H. Communication-efficient federated learning via optimal client sampling. *arXiv preprint arXiv:2007.15197*, 2020.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. Gaussian process regression: Active data selection and test point rejection. In *Mustererkennung 2000*, pp. 27–34. Springer, 2000.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2015.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in neural information processing systems*, pp. 4424–4434, 2017.

TensorFlow team. Tensorflow convolutional neural networks tutorial, 2016. URL https://www.tensorflow.org/tutorials/images/cnn.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yang, H. H., Liu, Z., Quek, T. Q., and Poor, H. V. Scheduling policies for federated learning in wireless networks. *IEEE transactions on communications*, 68(1):317–333, 2019.

A. Experiment Details

A.1. Model Parameters

Hyperparameters in FMNIST We follow (Cho et al., 2020) to construct the neural model on FMNIST: An MLP model with two hidden layers with 64 and 30 units, respectively. Under all three heterogeneous settings, we set the local batch size $B = 64$ and the number of local epochs $E = 3$. The learning rate η_0 is set to 0.005 initially, and halved at the 150-th and 300-th rounds. An SGD optimizer with a weight decay of 0.0001 and no momentum is used. We allocate data to $N = 100$ clients, and set the participation fraction $C = 0.1$ for the 1SPC setting, and $C = 0.05$ for the 2SPC and Dir settings.

Hyperparameters in CIFAR-10 We use a CNN with three convolutional layers (TensorFlow team, 2016) with 32, 64 and 64 kernels, respectively. And all convolution kernels are of size 3×3. Finally, the outputs of convolutional layers are fed into a fully-connected layer with 64 units. Under all three heterogeneous settings, we set the local batch size $B = 50$ and the number of local epochs $E = 5$. We use a learning rate $\eta = 0.01$ without learning rate decay, and a weight decay of 0.0003 for the SGD optimizer. The total number of clients and the client participation fraction are the same as those in FMNIST.

Hyperparameters for FedGP We set the dimension of client embedding $d = 15$ for all experiments. In Eq. 19, we set the number of reused history samples $M = 1$, and the discount factor $\gamma = \theta^{\Delta t}$ where $\theta = 0.9$ for experiments on FMNIST and $\theta = 0.99$ for experiments on CIFAR-10. And in each GP update round t, we use X^{t-1} as the initialization and use an Adam optimizer (Kingma & Ba, 2014) with learning rate 0.01 to optimize for X^t.

Hyperparameters for other baselines We use the same parameters $\alpha_1 = 0.75, \alpha_2 = 0.01$ and $\alpha_3 = 0.1$ as those in the paper (Goetz et al., 2019) for Active Federated Learning. And we set $d = 2NC$ for Power-of-choice Selection Strategy, which is empirically shown to be the best value of d in a highly heterogeneous setting in the paper (Cho et al., 2020).

Note that we implement the random selection strategy as uniformly sampling clients from $\mathcal{U}$ without replacement (McMahan et al., 2017), while Cho et al. (2020) implement the random selection strategy as sampling clients with replacement. Thus, our implemented random selection strategy achieves better performances than their implementation.

A.2. Dirichlet Distribution for Data Partition

We follow the idea in (Hsu et al., 2019) to construct the Dir heterogeneous setting, while we make some modifications to get an unbalanced non-identical data distribution.

For each client k, we sample the data distribution $\boldsymbol{q}_k \in \mathbb{R}^{10}$ from a dirichlet distribution independently, which could be formulated as

$$\boldsymbol{q}_k \sim \mathrm{Dir}(\alpha \boldsymbol{p}), \tag{35}$$

where $\boldsymbol{p}$ is the prior label distribution and $\alpha \in \mathbb{R}_+$ is the concentration parameter of the dirichlet distribution. We group $\boldsymbol{q}_k$ of all the clients together and get a fraction matrix $Q = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n]$. We denote the size of dataset on each client as $\boldsymbol{x} = [x_1, \cdots, x_N]^T$ and we get it from a solution of a quadratic programming:

$$\min_{\boldsymbol{x}} \quad \boldsymbol{x}^T \boldsymbol{x} \tag{36}$$

$$\text{subject to} \quad Q\boldsymbol{x} = \boldsymbol{d} \tag{37}$$

$$\boldsymbol{x} \in \mathbb{R}^N_{++}, \tag{38}$$

where $\boldsymbol{d}$ is the number of data with each label. We minimize $\|\boldsymbol{x}\|_2$ to avoid the cases where data distribution is over-concentrated on a small fraction of clients. In that case, the client selection problem might become trivial, since we can always ignore those clients with a small dataset and select those with a large dataset.

B. Extra Experimental Results

B.1. Ablation Study: Annealing Coefficient β

We also conduct experiments on FMNIST with different annealing coefficient β. We setup our experiments under three heterogeneous settings as in Section 4, with three different annealing coefficient $\beta = 0.95, 0.75, 0.5$. The test accuracy curve is shown in Figure 5. We can see that within a large range, the value of annealing coefficient only slightly influence the convergence rate as well as the final accuracy. Recalling the results of different GP training intervals Δt in Section 4.3, we can see that our method is not sensitive to the hyperparameters Δt and β.

We present the selected frequency of each client in Figure 6, and we can see that with a smaller β, the selected frequency tends to be "uniform". However, this does not mean that our selection strategy is equivalent to the uniformly random selection. Our sequential selection strategy introduces dependencies between selected clients as discussed in the multi-step insights in Section 3.4, which makes our selection strategy prefer some combinations of selected clients to others, while the uniformly random selection treats all the combinations equally. The advantage shown in Figure 5 compared to the uniformly random strategy demonstrates that selecting a good combination of clients, not only a good individual, is important.

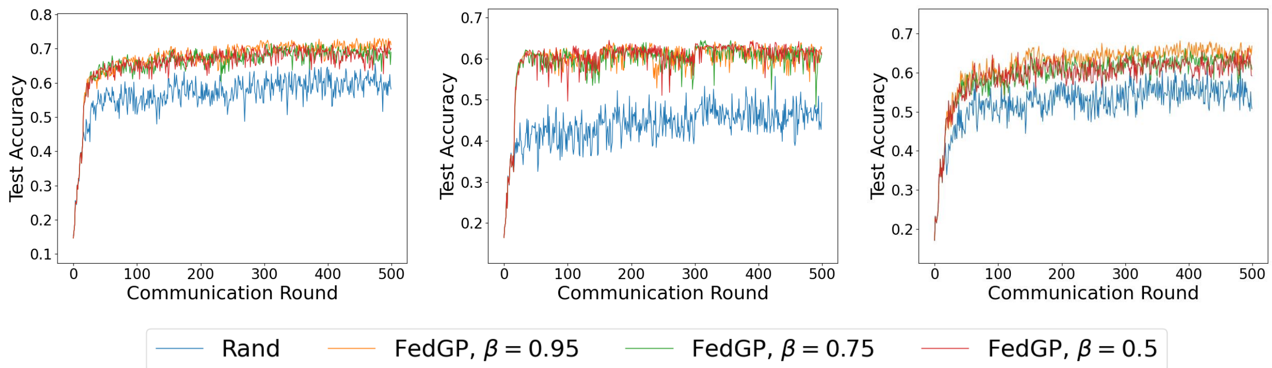

Figure 5. Test accuracy with different annealing coefficient β on FMNIST under three heterogeneous settings (left: 2SPC; median: 1SPC; right: Dir).

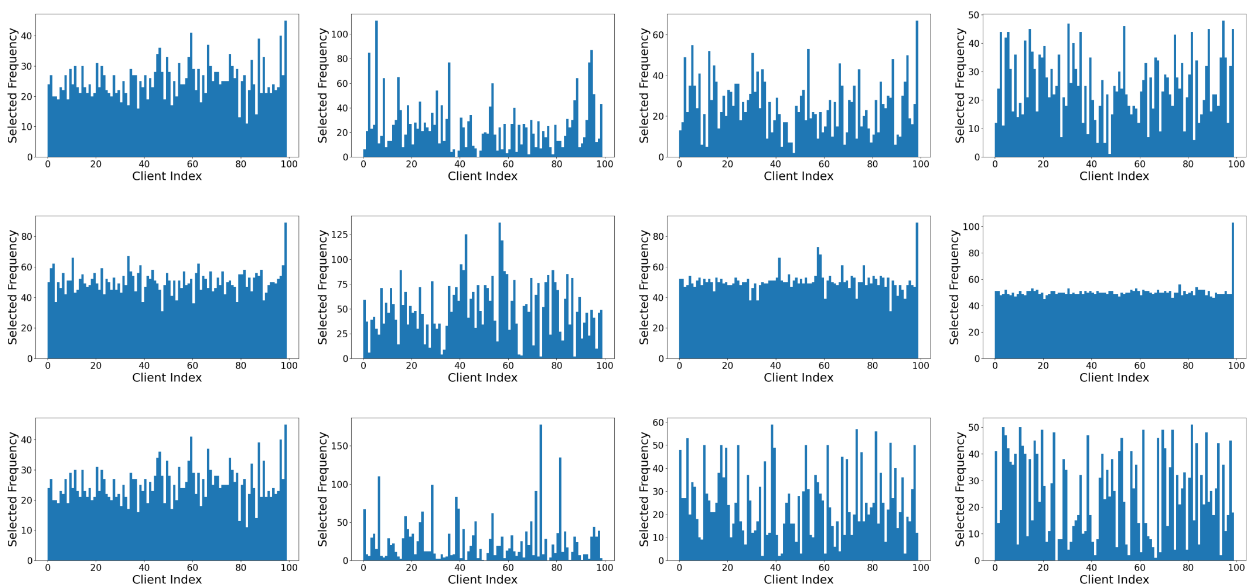

Figure 6. Selected Frequency of each client with different annealing coefficient β on FMNIST under three heterogeneous settings (top: 2SPC; median: 1SPC; bottom: Dir). The pictures on the left first column show the selected frequency of each client with the uniform selection strategy. The pictures on the 2-nd to the 4-th columns show the selected frequencies of FedGP with $\beta = 0.95, 0.75, 0.5$, respectively.