FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning

Minxue Tang[1], Xuefei Ning[2], Yitu Wang[1], Jingwei Sun[1], Yu Wang[2], Hai Li[1], Yiran Chen[1*]
[1]Department of Electrical and Computer Engineering, Duke University
[2]Department of Electronic Engineering, Tsinghua University
[1]{minxue.tang,yitu.wang,jingwei.sun,hai.li,yiran.chen}@duke.edu
[2]foxdoraame@gmail.com [2]yu-wang@tsinghua.edu.cn

Abstract

Client-wise data heterogeneity is one of the major issues that hinder effective training in federated learning (FL). Since the data distribution on each client may vary dramatically, the client selection strategy can significantly influence the convergence rate of the FL process. Active client selection strategies are popularly proposed in recent studies. However, they neglect the loss correlations between the clients and achieve only marginal improvement compared to the uniform selection strategy. In this work, we propose FedCor—an FL 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). Based on the GP model, we derive a client selection strategy with a significant reduction of expected global loss in each round. Besides, we develop an efficient GP training method with a low communication overhead in the FL scenario by utilizing the covariance stationarity. Our experimental results show that compared to the state-of-the-art method, FedCorr can improve the convergence rates by $34\% \sim 99\%$ *and* $26\% \sim 51\%$ *on FMNIST and CIFAR-10, respectively.*

1. Introduction

As a newly emerging distributed learning paradigm, federated learning (FL) [9, 12, 13, 17, 23] has recently attracted attention because of the offered data privacy. FL aims at dealing with scenarios where training data is distributed across a number of clients. Considering limited communication bandwidth and the privacy requirement, in each communication round, FL usually selects only a fraction of clients, and the selected clients will perform multiple iterations of local updating without exposing their own datasets [23]. This special scenario also introduces other challenges that distinguish FL from the conventional distributed learning [2, 35].

One major challenge in FL is the high degree of client-wise data heterogeneity [17], which is the inherent characteristic of a large number of clients. There have been many studies [10, 15, 16, 18, 20, 25, 27, 32] trying to tackle non-IID (independent and identically distributed) and unbalanced data of the clients in FL. Most of these studies [10, 18, 20, 32] focus on amending the local model updates or the central aggregation based on FedAvg [23].

Recently, active client selection arises as a complement of the aforementioned studies, aiming at accelerating the convergence of FL with non-IID data. Some recent studies propose to assign higher probability of being selected to the clients with larger training loss value [4, 6]. However, they neglect the correlations between the clients and consider their losses independently, which leads to only marginal performance improvement. In this paper, we propose a correlation-based active client selection strategy that can effectively alleviate the accuracy degradation caused by data heterogeneity and significantly boost the convergence of FL. Our key idea is mainly based on the following intuitions:

1. Clients do not contribute **equivalently**. For example, training with a large and balanced dataset on a "good" client can reduce the losses of most clients, while training with a small and extremely biased dataset on a "bad" client may increase the losses of other clients.

2. Clients do not contribute **independently**. The influence of selecting one client depends on the other selected clients because their local updates will be aggregated.

A toy experiment shown in Fig. 1 also illustrates the necessity of considering the correlations for client selection. In this experiment, each client has only one data sample, and thus each data point in the figure represents a client. The task is to select two clients (different markers represent the client selections of different strategies) for training a binary

*Corresponding Author

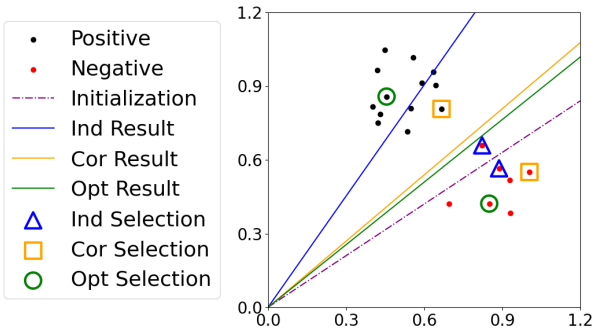

Figure 1. A toy experiment of different client selection strategies.

classifier (shown as the lines). The selection strategy that independently selects two clients with the highest local losses ("Ind Result") fails to reduce the global loss. In contrast, our method considers the correlations between the clients ("Cor Result") and derives a client selection that can achieve an almost lowest global loss ("Opt Result").

Based on the above intuitions, this work proposes FedCor, an FL framework built on a correlation-based client selection strategy, to boost the convergence of FL. Our main contributions are summarized as follows:

1. We model the client loss changes with a Gaussian Process (GP) and propose an interpretable client selection strategy with a significant reduction of the expected global loss in each communication round.

2. We propose a GP training method that utilizes the covariance stationarity to reduce the communication cost. Experiments show that the GP trained with our method can capture the client correlations well.

3. Experimental results demonstrate that FedCor stabilizes the training convergence and significantly improves the convergence rates by $34\% \sim 99\%$ and $26\% \sim 51\%$ on FMNIST and CIFAR-10, respectively.

2. Related Work

An important characteristic of FL [12, 13, 23] is the heterogeneity of clients, which raises new challenges of the training [9, 17, 31]. There are two kinds of heterogeneity in FL: systemic heterogeneity (computation ability, communication bandwidth, etc.) and statistical heterogeneity (non-IID, imbalanced data distribution) [17, 18]. This work mainly focuses on the latter one. A number of methods have been proposed to improve the basic FL algorithm, FedAvg [23], in heterogeneous settings. Some of them manipulate the local training loss like adding regularization terms to stabilized the training [8, 10, 18, 20, 26], while some other works amend the aggregation method to reduce the variance [24, 32].

Complementary to such methods, another way to improve the convergence of FL in non-IID settings is active client selection, which tries to strategically select clients for training in each round in stead of uniformly selecting. Goetz et al. [6] first propose to assign a high selection probability to the clients with large local loss. Cho et al. [4] select C clients with the largest loss among a randomly sampled subset $\mathbb{A} \subseteq \mathbb{U}$ with size $d > C$ to reduce the selection bias. However, neither of them consider the correlations between clients while making the client selection.

3. Preliminary

FL seeks for a global model $\boldsymbol{w}$ that achieves the best performance (e.g., the highest classification accuracy) on all N clients. The global loss function in FL is defined as:

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

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

where $l(\boldsymbol{w}; \xi)$ is the objective loss of data sample ξ evaluated on model $\boldsymbol{w}$. We refer to $l_k(\boldsymbol{w})$ as the local loss of client k, which is evaluated with the local dataset $\mathbb{D}_k$ (of size $|\mathbb{D}_k|$) on client k. The weight $p_k = |\mathbb{D}_k| / \sum_j |\mathbb{D}_j|$ of the client k is proportional to the size of its local dataset.

In consideration of the privacy and communication constraints, FL algorithms usually assume partial client participation and perform local model updates. In particular, in communication round t, only a subset $\mathbb{K}_t$ with size $|\mathbb{K}_t| = C \leq N$ of the overall client set $\mathbb{U}$ is selected to receive the global model $\boldsymbol{w}^t$ 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 aggregates them (usually by averaging [23]) to produce a new global model $\boldsymbol{w}^{t+1}$. We formulate this procedure as follows:

$$\boldsymbol{w}_k^{t+1} = \boldsymbol{w}^t - \eta_t \tilde{\nabla} l_k(\boldsymbol{w}^t), \tag{3}$$

$$\boldsymbol{w}^{t+1}(\mathbb{K}_t) = \frac{1}{C} \sum_{k \in \mathbb{K}_t} \boldsymbol{w}_k^{t+1} \tag{4}$$

$$= \boldsymbol{w}^t - \frac{\eta_t}{C} \sum_{k \in \mathbb{K}_t} \tilde{\nabla} l_k(\boldsymbol{w}^t), \tag{5}$$

where η_t is the learning rate and $\tilde{\nabla} l_k(\boldsymbol{w}^t)$ is the equivalent cumulative gradient [32] in the t-th communication round. More specifically, for an arbitrary optimizer on the client k, it produces $\Delta \boldsymbol{w}_k^{t,\tau} = -\eta \boldsymbol{d}_k^{t,\tau}$ as the local model update at the τ-th iteration in this round, and the cumulative gradient is calculated as $\tilde{\nabla} l_k(\boldsymbol{w}^t) = \sum_\tau \boldsymbol{d}_k^{t,\tau}$.

4. Methodology

In this section, we elaborate our proposed method, i.e., FedCor, that can effectively boost the convergence of FL.

We first formulate our goal of accelerating the convergence of FL as optimization problems that maximize the posterior expectation of loss decrease in Sec. 4.1. Then, Sec. 4.2 demonstrates empirical evidence that the prior distribution of loss changes in each communication round can be modeled as Gaussian Processes (GP). Based on this observation, we utilize GP to solve the optimization problems and obtain an effective client selection strategy for heterogeneous FL in Sec. 4.3. We further analyze the selection criterion of our client selection strategy and give out its intuitive interpretation in Sec. 4.4. Finally, in Sec. 4.5, we describe how we train the GP parameters in communication-constrained FL.

4.1. Problem Formulation

To achieve a fast convergence, we hope to find the client selection strategy which can lead to the maximal global loss decrease after each communication round. Accordingly, we define our target as solving a series of optimization problems, one for each communication round t:

$$\begin{aligned} \min_{\mathbb{K}_t} \quad & \Delta L^t(\mathbb{K}_t) = L(\boldsymbol{w}^{t+1}(\mathbb{K}_t)) - L(\boldsymbol{w}^t) \\ \text{subject to} \quad & \boldsymbol{w}^{t+1}(\mathbb{K}_t) = \boldsymbol{w}^t - \frac{\eta_t}{C} \sum_{k \in \mathbb{K}_t} \tilde{\nabla} l_k(\boldsymbol{w}^t). \end{aligned} \tag{6}$$

It is impractical in FL to search for the best client selection with multiple trials of different client selections since it introduces large communication and computation overhead. Therefore, we need an efficient way to predict the global loss decreases for different client selections and make a decision with very limited trials. To achieve this goal, we first reformulate the optimization problem in Eq. (6) with the following lemma. The proof of this lemma is in Appendix A.2.

Lemma 1. *The optimization problem in Eq. (6) is approximately equivalent to the following probabilistic form.*

$$\min_{\mathbb{K}_t} \quad \mathbb{E}_{\Delta \mathbf{l}^t | \Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t)} \Big[\sum_i p_i \Delta l_i^t \Big] = \sum_i p_i \tilde{\mu}_i^t(\Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t)), \tag{7}$$

where $\Delta \mathbf{l}^t = [\Delta l_1^t, \cdots, \Delta l_N^t]$ is the loss changes of all clients in round t, which is a random variable w.r.t random client selection in round t. $\tilde{\boldsymbol{\mu}}^t(\Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t))$ is the posterior mean of $\Delta \mathbf{l}^t$ conditioned on $\Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t) = [\Delta l_i^t(\mathbb{K}_t)]_{i \in \mathbb{K}_t}$.

The reformulated objective in Eq. (7) tells that if we can predict the loss changes of those clients selected for training ($\Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t)$), we can predict the global loss change with its posterior mean and make decision according to it. Now what we need is a probabilistic model of the loss changes $\Delta \mathbf{l}^t$ to make the prediction and calculate the posterior.

4.2. Modeling Loss Changes with GP

It is a common practice to assume a GP prior over an unknown objective function in Bayesian Optimization [3, 30].

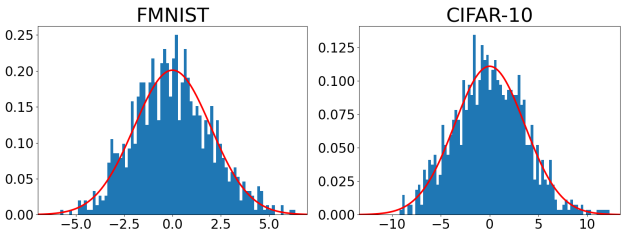

Figure 2. Histograms of the first principle component in Non-IID FL [23]. More details and full results can be found in Appendix D.2.

Our preliminary investigation (partly) shown in Fig. 2 also indicates that the prior distribution of the loss changes in one communication round follow a GP. Specifically, we randomly sample a number of client selections and perform one round of training to get samples of the loss changes. Then, we conduct PCA on these loss change samples and plot histograms of the first several principle components. The red line in the Fig. 2 is the Gaussian PDF with the sample mean and sample variance. And we can see that this Gaussian distribution can approximate the distribution of the samples well. A mathematical explanation of this observation is also given out in Appendix A.1.

Accordingly, we propose to model the loss changes in one communication round t with a GP prior as follows:

$$\Delta \mathbf{l}^t = [\Delta l_1^t, \cdots, \Delta l_N^t] \sim \mathcal{N}(\Delta \boldsymbol{l}^t; \boldsymbol{\mu}^t, \boldsymbol{\Sigma}^t). \tag{8}$$

Remark. In order to efficiently learn the covariance in FL, rather than directly working with the covariance matrix, we embed all clients into a continuous vector space and use a kernel function to calculate the covariance (see Sec. 4.5). Thus, we still use the term GP instead of Multivariate Gaussian Distribution, though the dimension of $\Delta \mathbf{l}^t$ is finite.

A good property of GP is that we can get a closed form of the posterior expectation in Eq. (7), which makes our client selection strategy interpretable. In the next sections, we will propose our client selection strategy based on the GP model, and then give an interpretation of it. We leave the training method for the parameters $(\boldsymbol{\mu}^t, \boldsymbol{\Sigma}^t)$ in GP to Sec. 4.5.

4.3. Client Selection Strategy

While we have get the probabilistic model to calculate the posterior expectation, it is still not determined how to predict the loss changes of the clients selected for training, namely $\Delta \boldsymbol{l}^t_{\mathbb{K}_t}(\mathbb{K}_t)$. Inspired by UCB methods [1, 5, 28], we develop an iterative method that predict the loss change and select one client in each iteration, as shown in Algorithm 1. There are three steps in one iteration:

(i) Prediction. In each iteration, we first make an prediction $\Delta \hat{l}_k^t$ for each client k if it is selected. Generally, the selected client would have a large loss decrease since it directly participate in the model update. Thus, we propose to use the

Algorithm 1 Client Selection Strategy with GP

Require: $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$ of the GP, scale factor $\boldsymbol{\alpha}^t$
Ensure: Client Selection $\mathbb{K}_t$
Initialize $\mathbb{K}_t \leftarrow \emptyset$, $\mathbb{P} \leftarrow \mathbb{U}$.
while $|\mathbb{K}_t| < C$ **do**
  for each client $k \in \mathbb{P}$ **do**
    Predict its loss change if select it: $\Delta\hat{l}_k^t = \mu_k^t - \alpha_k^t \sigma_k^t$.
    Calculate the posterior mean of the loss changes $\tilde{\boldsymbol{\mu}}^t(\Delta\hat{l}_k^t)$.
  end for
  Select the client by $k^* = \arg\min_k \sum_i p_i \tilde{\mu}_i^t(\Delta\hat{l}_k^t)$.
  Add k^* into $\mathbb{K}_t$ and remove it from $\mathbb{P}$.
  $\boldsymbol{\mu}^t \leftarrow \tilde{\boldsymbol{\mu}}^t(\Delta\hat{l}_{k^*}^t)$, $\boldsymbol{\Sigma}^t \leftarrow \tilde{\boldsymbol{\Sigma}}^t(\Delta\hat{l}_{k^*}^t)$.
end while

lower confidence bound as the prediction:

$$\Delta\hat{l}_k^t = \mu_k^t - \alpha_k^t \sigma_k^t; \quad \alpha_k^t = a\beta^{\tau_k^t}, \tag{9}$$

where $\sigma_k^t = \sqrt{\Sigma_{k,k}^t}$, and a is a scale constant. $\beta \in (0,1)$ is an annealing coefficient, and its index τ_k^t denotes how many times client k has been selected. We will discuss this annealing coefficient more in Sec. 4.5.

(ii) Selection. The client k^* is selected to minimize the posterior expectation of the overall loss conditioned on its loss change prediction made in the last step:

$$k^* = \arg\min_k \sum_i p_i \tilde{\mu}_i^t(\Delta\hat{l}_k^t) \tag{10}$$

(iii) Posterior. After selecting the client k^*, we update the GP for the next iteration with the posterior conditioned on the loss change prediction of k^*:

$$\boldsymbol{\mu}^t \leftarrow \tilde{\boldsymbol{\mu}}^t(\Delta\hat{l}_{k^*}^t), \quad \boldsymbol{\Sigma}^t \leftarrow \tilde{\boldsymbol{\Sigma}}^t(\Delta\hat{l}_{k^*}^t). \tag{11}$$

By updating the GP with its posterior, we iteratively add conditions into the probabilistic model to approach the fully conditioned distribution $p(\Delta\boldsymbol{l}^t|\Delta\boldsymbol{l}_{\mathbb{K}_t}^t(\mathbb{K}_t))$, and make the next prediction of the loss change more accurate.

There are some similarities between our method and traditional Bayesian Optimization: Using GP as a prior of the objective function, and using UCB as well as posterior distribution for iterative selection [3, 5, 28]. However, there is a key difference: In each communication round, we determine the client selection with only predictions instead of measurements of the global loss changes, while traditional Bayesian Optimization requires a sequence of measurements as new information to make decisions. The measurements of global loss changes will introduce large communication overhead and are unfeasible in FL.

Algorithm 2 FedCor

Initialize $\boldsymbol{X}_0$ and Global Model $\boldsymbol{w}_0$.
for each round $t = 0, 1, ...$ **do**
  if $t\%\Delta t$==0 **then**
    Uniformly sample S client selections $\mathbb{S}_{t,i}, i = 1, 2, ..., S$.
    for $i = 1, 2, ..., S$ **do**
      $\boldsymbol{w}^{t+1}(\mathbb{S}_{t,i}) \leftarrow \boldsymbol{w}^t - \frac{\eta_t}{C}\sum_{k\in\mathbb{S}_{t,i}} \tilde{\nabla} l_k(\boldsymbol{w}^t)$.
      Collect $\Delta\boldsymbol{l}^t(\mathbb{S}_{t,i}) \leftarrow \boldsymbol{l}(\boldsymbol{w}^{t+1}(\mathbb{S}_{t,i})) - \boldsymbol{l}(\boldsymbol{w}^t)$.
    end for
    Reset $\alpha_k \leftarrow 1, \forall k \in \mathbb{U}$.
  end if
  Update $\boldsymbol{X}_t$ with Eq. (16).
  Select clients $\mathbb{K}_t$ with Algorithm 1 ($\boldsymbol{\mu}^t = \boldsymbol{0}, \boldsymbol{\Sigma}^t = \boldsymbol{X}^{t^T}\boldsymbol{X}^t, \boldsymbol{\alpha}^t = \boldsymbol{\alpha}$).
  $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^{t+1}(\mathbb{K}_t) = \boldsymbol{w}^t - \frac{\eta_t}{C}\sum_{k\in\mathbb{K}_t} \tilde{\nabla} l_k(\boldsymbol{w}^t)$.
  Update $\boldsymbol{\alpha}_{\mathbb{K}_t} \leftarrow \beta\boldsymbol{\alpha}_{\mathbb{K}_t}$.
end for

4.4. Insights into Our Selection Strategy

In this section, we give an intuitive interpretation of our selection strategy and show the benefits of it within a simple case. A more detailed analysis of the selection criterion and convergence of FedCor can be found in Appendix B.

For simplicity, we omit all superscript t in this section. Lemma 2 gives the selection criterion of FedCor in a simple case where we only select two clients, and the proof can be found in Appendix A.3.

Lemma 2. *The selection criterion of FedCor when selecting two clients k_1 and k_2 can be written as*

$$k_1 = \arg\max_k \quad \beta^{\tau_k} \sum_i p_i \sigma_i r_{ik}, \tag{12}$$

$$k_2 = \arg\max_{k'} \quad \frac{\beta^{\tau_{k'}} \Big[\overbrace{\sum_i p_i \sigma_i r_{ik'}}^{(A)} - r_{k_1 k'} \overbrace{\sum_i p_i \sigma_i r_{ik_1}}^{(B)} \Big]}{\sqrt{1 - r_{k'k_1}^2}}, \tag{13}$$

where $r_{ij} = \Sigma_{i,j}/\sigma_i\sigma_j$ is the Pearson correlation coefficient.

(i) Single-Iteration. Eq. (12) has a clear interpretation 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 the clients into consideration, and can conduct better selection compared with those algorithms that only consider the loss of each client independently [4, 6].

(ii) Multi-Iteration. In Eq. (13), term (A) and (B) are the single-iteration selection criterion in Eq. (12) of client k' and k_1, respectively. Since we have maximized (B) when

selecting client k_1, term (B) is usually positive. Therefore, the selection of k' does not only consider its correlations with other clients ($r_{ik'}$), but also prefers the clients that have small correlations $r_{k_1k'}$ with the previous selected client k_1. This criterion penalizes selection redundancy and leads to a client selection with diverse data, which reduces the variance and makes the training process more stable. Since clients with similar data generate similar local updates, selecting redundant clients only brings marginal gains to the global performance or would even drive the optimization into bad local optimum. This selection preference is also demonstrated in Fig. 1, where FedCor chooses one positive and one negative point as the optimal selection does.

4.5. Training GP in FL

As a classical machine learning model, GP has been widely discussed and well studied [33]. There have been many methods to train the parameters in GP, namely, the covariance $\boldsymbol{\Sigma}^t$ in Eq. (8) [1]. Nevertheless, to make the GP training feasible in the communication-constrained FL procedure, we should revise the GP training method to reduce the number of samples and better utilize historical information.

In GP, a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is used to calculate the covariance [33] as $\Sigma^t_{i,j} = K(\boldsymbol{x}^t_i, \boldsymbol{x}^t_j)$, where $\boldsymbol{x}^t_i, \boldsymbol{x}^t_j$ are the features of the data points i and j, respectively. Following this, we assign a trainable embedding in a latent space to each client. The embedding of the k-th client is noted as $\boldsymbol{x}^t_k \in \mathbb{R}^d$ ($d < N$), and we choose the kernel function as

$$K(\boldsymbol{x}^t_i, \boldsymbol{x}^t_j) = {\boldsymbol{x}^t_i}^T \boldsymbol{x}^t_j, \tag{14}$$

which is a homogeneous linear kernel [33]. This low-rank formulation reduces the number of parameters we need to learn, thus making the GP training more data-efficient.

A commonly used GP training method is maximum likelihood evaluation, where we uniformly sample S client selection $\{\mathbb{S}_{t,i} : i = 1, \cdots, S\}$, and maximize the likelihood of the corresponding loss changes $\{\Delta \boldsymbol{l}^t(\mathbb{S}_{t,i}) : i = 1, \cdots, S\}$ to learn the embedding matrix $\boldsymbol{X}^t = [\boldsymbol{x}^t_1, \cdots, \boldsymbol{x}^t_N]$:

$$\boldsymbol{X}^t = \arg\max_{\boldsymbol{X}} \sum_{i=1}^{S} \log p(\Delta \boldsymbol{l}^t(\mathbb{S}_{t,i}) | \boldsymbol{X}). \tag{15}$$

However, to collect each sample $\Delta \boldsymbol{l}^t(\mathbb{S}_{t,i})$, we have to broadcast $\boldsymbol{w}^{t+1}(\mathbb{S}_{t,i})$ to all the clients. And since a large S is usually required for an unbiased estimation in each communication round t, the vanilla training procedure in Eq. (15) 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. Thus, we hypothesise that the covariance also changes slowly in the concerned time range. To verify this, we use a large number of samples to evaluate the covariance $\boldsymbol{\Sigma}^t$ in each communication round, and calculate the cosine similarity between $\boldsymbol{\Sigma}^t$ and $\boldsymbol{\Sigma}^{t+\Delta t}$. We set $\Delta t = 10$ for FMNIST and $\Delta t = 50$ for CIFAR-10. As shown in Fig. 3, we can see that the similarity keeps very high (> 0.97 for FMNIST and > 0.95 for CIFAR-10) during the whole FL training process.

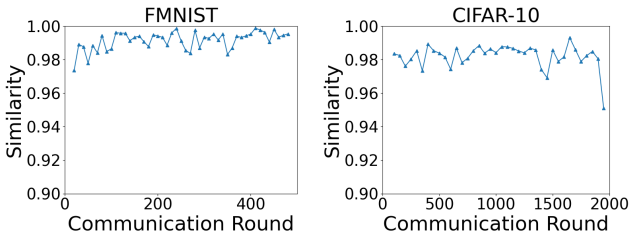

Figure 3. Covariance Stationarity in Non-IID FL [23]. Full experiment results and more details can be found in Appendix D.3.

Accordingly, we do not need to update $\boldsymbol{X}^t$ in every round but inherit the embedding matrix $\boldsymbol{X}^{t-1}$ from the last round and train it only every Δt rounds. Furthermore, we can reuse historical samples for GP training to reduce the number of samples S that we need to collect in each GP training round. We summarize our update rule of $\boldsymbol{X}^t$ as follows:

$$\boldsymbol{X}^t = \begin{cases} \boldsymbol{X}^{t-1}, & t\%\Delta t \neq 0; \\ \arg\max_{\boldsymbol{X}} \Phi_t(\boldsymbol{X}), & t\%\Delta t = 0, \end{cases} \tag{16}$$

where

$$\Phi_t(\boldsymbol{X}) = \sum_{m=0}^{M} \sum_{i=1}^{S} \gamma^m \log p(\Delta \boldsymbol{l}^{t-m\Delta t}(\mathbb{S}_{t-m\Delta t,i}) | \boldsymbol{X}). \tag{17}$$

M is the number of reused historical samples, and $\gamma < 1$ is the discount factor to weight the historical samples. Our method is able to reduce the communication overhead with a large Δt and $S = 1$, while guaranteeing the performance.

As we only update the covariance $\boldsymbol{\Sigma}$ 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 global model to overfit on their data, which may hinder the convergence of FL. In practice, we reset τ_k to 0 after each GP training round to achieve the fastest convergence while avoiding overfitting on some clients.

We summarize our overall framework FedCor in Algorithm 2. It is noteworthy that our method is orthogonal to existing FL optimizers that amend the training loss or the aggregation scheme, e.g., FedAvg [23] and FedProx [18]. So our method can be combined with any of them.

[1] We do not train $\boldsymbol{\mu}$ and set it to $\mathbf{0}$, since it does not affect the selection strategy as we can see in Lemma 2 and Appendix B.

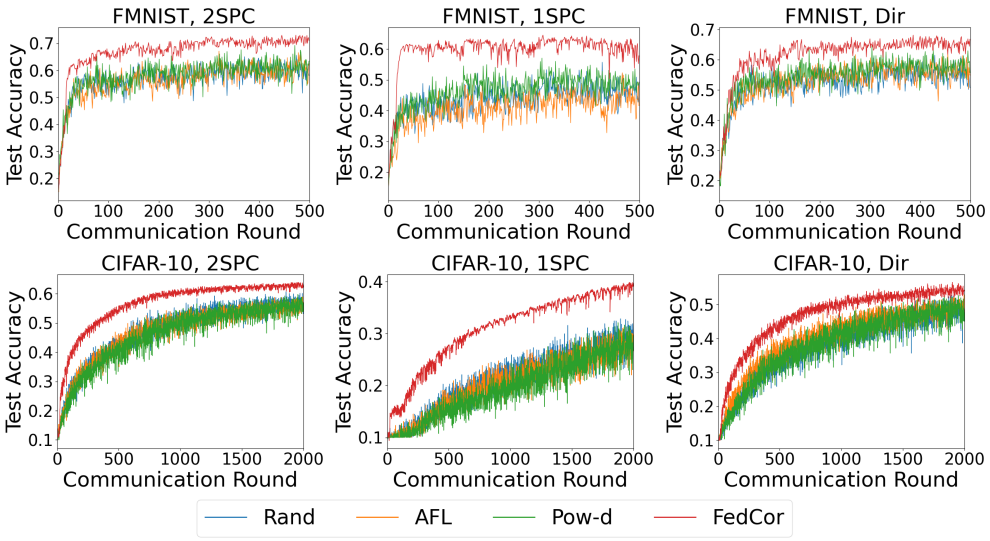

Figure 4. Test accuracy on FMNIST and CIFAR-10 under three heterogeneous settings (2SPC, 1SPC and Dir). All experiments in one figure share the same hyperparameters except for the client selection strategy.

Method	FMNIST			CIFAR-10		
	2SPC(69%)	1SPC(62%)	Dir(64%)	2SPC(62%)	1SPC(36%)	Dir(54%)
Rand	295.8 ± 92.0	N/A	141.0 ± 73.0	1561.2 ± 236.2	1750.4 ± 190.3	N/A
AFL	218.6 ± 117.3	N/A	169.0 ± 166.1	N/A	1845.2 ± 28.8	1524.4 ± 267.9
Pow-d	126.6 ± 78.2	167.2 ± 72.3	123.0 ± 101.0	1558.2 ± 227.0	1752.2 ± 186.2	1355.2 ± 151.3
FedCor (Ours)	$\mathbf{94.8 \pm 18.4}$	$\mathbf{84.0 \pm 53.1}$	$\mathbf{68.8 \pm 27.5}$	$\mathbf{1033.4 \pm 123.7}$	$\mathbf{1269.2 \pm 70.6}$	$\mathbf{1076.8 \pm 262.8}$

Table 1. The number of communication rounds for each selection strategy to achieve target test accuracies (specified in parentheses) under three heterogeneous settings (2SPC, 1SPC and Dir). 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 target accuracy with some random seeds within the maximal number of communication rounds (500 for FMNIST and 2000 for CIFAR-10).

5. Experiments

5.1. Experiment Settings

We conduct experiments on two datasets, FMNIST [34] and CIFAR-10 [14]. For FMNIST, we adopt an MLP model with two hidden layers, and this model achieves an accuracy of 85.92% with centralized training. For CIFAR-10, we adopt a CNN model with three convolutional layers followed by one fully connected layer, and this model can achieve an accuracy of 73.84% with centralized training. More details on the model construction and training hyperparameters can be found in Appendix C.1. For each dataset, we experiment with three different heterogeneous data partitions on $N = 100$ clients as follows.

(i) 2 shards per client (2SPC): This setting is the same as the non-IID setting in [23]. We 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. We select $C = 5$ clients in each round within this setting.

(ii) 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. We select $C = 10$ clients in each round within this setting.

(iii) Dirichlet Distribution with $\alpha = 0.2$ (Dir): We inherit and slightly change the setting from [7] 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 C.2. We select $C = 5$ clients in each round within this setting.

We divide the training process of FedCor into two phases: (i) Warm-up phase: We uniformly sample client selection $\mathbb{K}_t$ and collect the loss values of all the clients in $\mathbb{U}$ to train the GP in each round, i.e., $\Delta t = 1$ and $S = 1$. We set the length of the warm-up phase to 15 for FMNIST and 20 for CIFAR-10. (ii) Normal phase: After the warm-up phase, we

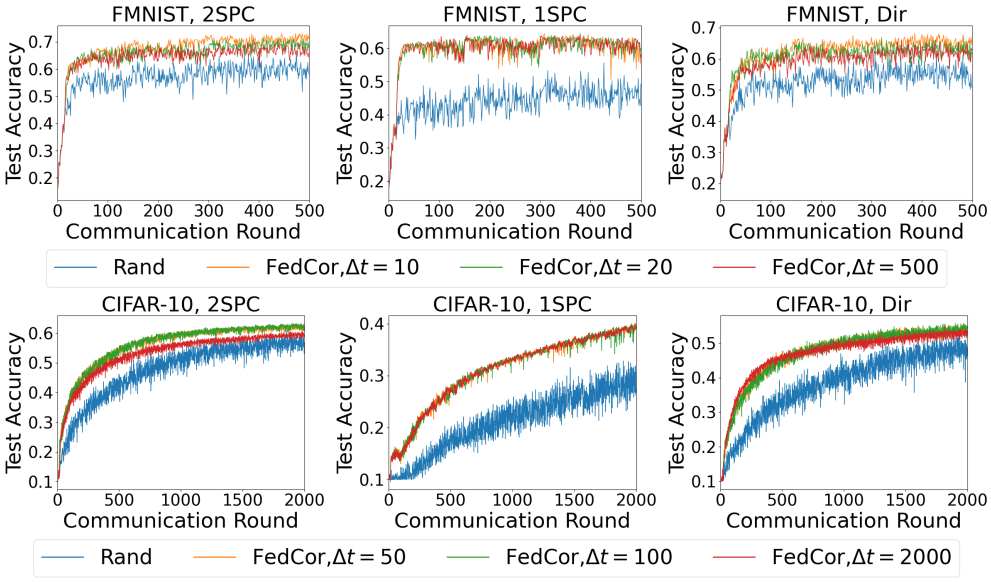

Figure 5. Test accuracy with different GP training interval Δt on FMNIST and CIFAR-10 under 2SPC, 1SPC and Dir.

follow Algorithm 2 to select clients and update the GP.

In all the experiments, we use FedAvg [23] as the FL optimizer. We present the average results using five random seeds in all experiments. We will first show that our method can achieve faster and more stable convergence, compared with three baselines: random selection (Rand), Active FL (AFL) [6] and Power-of-choice Selection Strategy (Pow-d) [4]. Then, we will give ablation studies on the GP training interval Δt as well as the annealing coefficient β. Finally, we visualize the client embeddings $\boldsymbol{X}$ with t-SNE [21] and show that FedCor can effectively capture the correlations.

5.2. Convergence under Heterogeneous Settings

We compare the convergence rate of our method FedCor with the other baselines on both FMNIST and CIFAR-10, and demonstrate the results in Figure 4. We set the GP update interval $\Delta t = 10$ and the annealing coefficient $\beta = 0.95$ for FMNIST experiments, and $\Delta t = 50$ and $\beta = 0.9$ for CIFAR-10 experiments.

As shown in Figure 4, FedCor achieves the highest test accuracy and the fastest convergence in all experiments. While other active client selection strategies show only slight or even no superiority compared with the fully random strategy, our method clearly outperforms all baselines, especially under the extremely heterogeneous setting when data on each client contains only one label (1SPC). Furthermore, the learning curves of FedCor are more smooth and less noisy than those of other methods, meaning that FedCor 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 FedCor achieves the specified accuracy $34\% \sim 99\%$ and $26\% \sim 51\%$ faster than Pow-d on FMNIST and CIFAR-10, respectively.

5.3. Results with Larger GP 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 5. 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 $\Delta t = 2000$ for CIFAR-10), FedCor still achieves faster convergence than the random selection strategy. These results indicate that the correlations learned by the GP model are stable, which supports our assumption in Section 4.5. In a word, one can largely reduce the communication overhead by training the GP model with a very low frequency while guaranteeing the convergence rate and accuracy under the communication-bounded FL setting.

5.4. Influence of Annealing Coefficient

We also conduct experiments with different annealing coefficient β that controls how "concentrated" the client selection is. We perform FedCor with $\Delta t = 10$ and $\beta = 0.5, 0.75, 0.9$ for FMNIST, and $\Delta t = 50, \beta = 0.9, 0.95, 0.99$ for CIFAR-10. The learning curves as well as the client selection frequencies under 2SPC setting are

Figure 6. Test accuracy and client selection frequency with different annealing coefficient β on FMNIST and CIFAR-10 under the 2SPC setting. The frequency is represented as the number of times each client is selected during the whole training process.

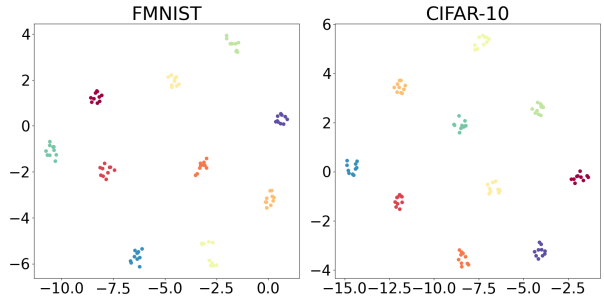

Figure 7. Visualization of client embedding under the 1SPC setting.

shown in Fig. 6, and we leave the full results under the 1SPC and Dir settings to Appendix D.1. We observe that when using a smaller β, the overall client selections appear to be more "uniform", while the learning curves are almost invariant. Notice that this does not mean that FedCor with small β is equivalent to uniform sampling, instead, FedCor still achieves consistent improvements compared to uniform sampling. And Sec. 4.4 havs discussed the reason: FedCor not only considers the benefit that each client brings to the federation, but also considers the correlations among the clients to select the best group of clients. The experimental results here show that it is more important to select a good "group" of clients than just good individuals.

5.5. Visualization of Client Embedding

To obtain an insight into the correlations learned by the GP model, we show the t-SNE [21] plot of the client embeddings learned in the warm-up phase under the 1SPC setting. In Fig. 7, each embedding is labeled with the only data label on the corresponding client. We normalize the length of embedding vectors to 1 so that the distance between two embeddings can reveal the correlation. We can see that the embeddings of clients with the same label are clustered together, which demonstrates that FedCor has captured the correlations between clients correctly in the warm-up phase.

6. Conclusion and Future Work

This work proposes FedCor, an FL framework with a novel client selection strategy for heterogeneous settings. FedCor is based on the intuition that it is crucial to utilize the correlations between clients to achieve a faster and more stable convergence in heterogeneous FL. Specifically, we model the client correlations with a GP, and design an effective and interpretable client selection strategy based on it. We also develop a efficient method to train the GP with a low communication overhead. Experimental results on FMNIST and CIFAR-10 show that FedCor effectively accelerates and stabilizes the training process under highly heterogeneous settings. In addition, we verify that FedCor captures the client correlation correctly using only the loss information. How to extend FedCor to the other tasks and further utilize the captured correlations is an interesting direction for future work. Besides, our method focuses on the cross-silo federated learning scenario [9], and how to extend it to the cross-device scenario is a meaningful topic.

Acknowledgement

This research was generously supported in part by Gift from Amazon, etc. Any opinions, conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of Amazon and its contractors.

References

[1] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002. 3

[2] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011. 1

[3] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. 3, 4

[4] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020. 1, 2, 4, 7, 14, 17, 18

[5] Dennis D Cox and Susan John. A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992. 3, 4

[6] Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. Active federated learning. *arXiv preprint arXiv:1909.12641*, 2019. 1, 2, 4, 7, 18

[7] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 6, 18

[8] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020. 2

[9] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 1, 2, 8

[10] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019. 1, 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 18

[12] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 1, 2

[13] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 1, 2

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[15] Ang Li, Jingwei Sun, Pengcheng Li, Yu Pu, Hai Li, and Yiran Chen. Hermes: an efficient federated learning framework for heterogeneous mobile clients. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 420–437, 2021. 1

[16] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020. 1

[17] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1, 2

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 1, 2, 5

[19] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019. 14, 15, 17

[20] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019. 1, 2

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7, 8

[22] Stephan Mandt, Matthew Hoffman, and David Blei. A variational analysis of stochastic gradient algorithms. In *International conference on machine learning*, pages 354–363, 2016. 11

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 5, 6, 7, 18

[24] Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. *arXiv preprint arXiv:2102.03198*, 2021. 2

[25] Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *arXiv preprint arXiv:2006.08907*, 2020. 1

[26] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019. 2

[27] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in neural information processing systems*, pages 4424–4434, 2017. 1

[28] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012. 3, 4

[29] TensorFlow team. Tensorflow convolutional neural networks tutorial. https://www.tensorflow.org/tutorials/images/cnn, 2016. 17

[30] Ngo Anh Vien, Heiko Zimmermann, and Marc Toussaint. Bayesian functional optimization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[31] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021. 2

[32] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020. 1, 2

[33] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006. 5

[34] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[35] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019. 1