INFOCOM 2023 1570829723

Heterogeneity-Aware Federated Learning with Adaptive Client Selection and Gradient Compression

Zhida Jiang[1,2] Yang Xu[*1,2] Hongli Xu[*1,2] Zhiyuan Wang[1,2] Chen Qian[3]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]Suzhou Institute for Advanced Research, University of Science and Technology of China
[3]Department of Computer Science and Engineering, Jack Baskin School of Engineering, University of California, Santa Cru

***Abstract*—Federated learning (FL) allows multiple clients cooperatively train models without disclosing local data. However, the existing works fail to address all these practical concerns in FL: limited communication resources, dynamic network conditions and heterogeneous client properties, which slow down the convergence of FL. To tackle the above challenges, we propose a heterogeneity-aware FL framework, called FedCG, with adaptive client selection and gradient compression. Specifically, the parameter server (PS) selects a representative client subset considering statistical heterogeneity and sends the global model to them. After local training, these selected clients upload compressed model updates matching their capabilities to the PS for aggregation, which significantly alleviates the communication load and mitigates the straggler effect. We theoretically analyze the impact of both client selection and gradient compression on convergence performance. Guided by the derived convergence rate, we develop an iteration-based algorithm to jointly optimize client selection and compression ratio decision using submodular maximization and linear programming. Extensive experiments on both real-world prototypes and simulations show that FedCG can provide up to 5.3× speedup compared to other methods.**

***Index Terms*—Federated Learning, Heterogeneity, Client Selection, Gradient Compression**

I. Introduction

Recently, federated learning (FL) [1] as a novel distributed machine learning paradigm has attracted a lot of attention. In FL, training data are distributed across a large number of edge devices, such as mobile phones, personal computers, or smart home devices. Under the orchestration of the parameter server (PS), these devices (*i.e.*, clients) cooperatively train a global inference model without sharing raw data, which efficiently leverages local computing resources of edge devices and addresses data privacy concerns. With the technical advantages and implemental feasibilities, FL has been applied in a variety of applications, such as next word prediction, extended reality, and smart manufacturing [2].

Despite its practical effectiveness, there are several key challenges unique to the FL setting that make it difficult to train high-quality models. *(1) Limited communication resources.* Since the clients participating in FL need to communicate with the PS iteratively over bandwidth-limited networks, the communication cost is prohibitive and forms a huge impediment to FL's viability, especially when training modern models with millions of parameters [3]. *(2) Dynamic network conditions.* Owing to link instability and bandwidth competition, the communication conditions of wireless channels may fluctuate over time, resulting in dynamics of available bandwidth [4]. For example, a user's smartphone may be allocated higher bandwidth when transmitting model updates at night than during the day. *(3) Heterogeneous client properties.* The heterogeneity of the clients usually includes capability heterogeneity and statistical heterogeneity. The clients may be equipped with different computing chips and located in diverse regions, thus their capabilities vary significantly [5]. The stragglers will delay the aggregation step and make the training process inefficient. Besides, due to different user preferences and contexts, local data on each client are not independent and identically distributed (non-IID). For instance, the images collected by the cameras reflect the demographics of each camera's location. Heterogeneous statistical data will bring the biases in training and eventually cause an accuracy degradation of FL [6].

To improve communication efficiency of FL, a natural solution is to reduce the size of transmitted payload or select only a fraction of clients participating in training. The existing works have adopted quantization [4], [7]–[10] or sparsification [3], [11]–[17] techniques to relax the communication load. But these compression algorithms often assign fixed or identical compression ratios to all clients, which are agnostic to the capability heterogeneity and thereby result in considerable completion time lags. Besides, these compression schemes [18] do not take statistical heterogeneity into account, deteriorating training efficiency in the presence of non-IID data. Another line of studies aims to design client selection (or client sampling) schemes based on heterogeneous client properties [5], [6], [19]–[28], most of which lack joint consideration of capability and statistical heterogeneity. Although some works take two types of heterogeneity into account [29], the derived client selection probabilities are fixed during the training process, which cannot adapt to network dynamics.

In summary, most prior works fail to address all the aforementioned challenges, thereby hindering efficient FL. This motivates us to study the following question: *how to enhance FL by simultaneously addressing the challenges of communication efficiency, network dynamics and client heterogeneity?*

To tackle this problem, we propose a heterogeneity-aware FL framework, called FedCG (Federated Learning with **C**lient selection and **G**radient compression). At each round, the PS selects a diverse subset of clients that carry representative gradient information and then sends the global model to the selected clients. After local training, these clients adopt gra-

dient compression to further boost communication efficiency. FedCG adaptively assigns appropriate compression ratios to selected clients based on their heterogeneous and time-varying capabilities. In this way, each client uploads compressed model updates matching its capabilities to the PS. Finally, the PS aggregates the model updates to obtain the latest global model. Under this framework, our advantages are reflected in two aspects. On one hand, we select a representative client subset such that their aggregated model updates approximate full client aggregation [22]. By encouraging diversity in client selection, FedCG can effectively reduce redundant communication and promote fairness, which modulates the skew introduced by non-IID data. On the other hand, different compression ratios will adapt to dynamic network conditions and heterogeneous capabilities, which contributes to mitigating the straggler effect and thus accelerates the training process.

More importantly, instead of directly combining client selection and gradient compression, we highlight that their decisions are interacted and demonstrate the need for joint optimization. Specifically, the compression ratios should be adapted to the heterogeneous capabilities of the selected clients. Correspondingly, client selection is also bound up with the degree of gradient compression. Selecting clients with over-compressed gradients will impede convergence. As a result, the naive combination of existing client selection and gradient compression schemes cannot adequately address the key challenges of FL and may degrade training performance, which is empirically verified in Section VI.

However, jointly optimizing client selection and compression ratio is non-trivial for the following reasons. *Firstly*, the quantitative relationship between client selection, gradient compression and model convergence is unclear. *Secondly*, it is difficult to determine the proper compression ratios to achieve a delicate trade-off between resource overhead and model accuracy. Things will get even worse while considering the capability heterogeneity across different clients. *Thirdly*, the tightly coupled problem of client selection and compression ratio decision adds additional challenges to algorithm design. In light of the above discussion, we state the key contributions of this paper as follows:

- We propose a novel FL framework, called FedCG, which addresses the challenges of communication efficiency, network dynamics and client heterogeneity by adaptive client selection and gradient compression. We theoretically analyze the impact of client selection and gradient compression on convergence performance.
- Guided by the convergence analysis, we apply submodular maximization to select diverse clients, and determine different compression ratios for heterogeneous clients to achieve the trade-off between overhead and accuracy. We develop an iteration-based algorithm to jointly optimize client selection and compression ratio decision for the tightly coupled problem.
- We evaluate the performance of our proposed framework on both a hardware platform and a simulated environment. Extensive experimental results demonstrate that for both convex and non-convex machine learning models, FedCG can provide up to 5.3× speedup compared to state-of-the-art methods.

The remainder of this paper is organized as follows. Section II reviews related work. Section III introduces our proposed framework and formulates the optimization problem. Section IV provides the convergence analysis of FedCG. Section V designs a joint optimization algorithm for client selection and compression ratio decision. Section VI presents experimental results, and finally Section VII concludes the paper.

II. Related Work

In FL, the iterative communication between the PS and clients will incur considerable costs, particularly when the underlying model is of high complexity [3]. To this end, various works have been devoted to improving communication efficiency by reducing the size of transmitted models/gradients or selecting a subset of clients. On one hand, compression techniques have been adopted to alleviate the transmission burden, including quantization [4], [7]–[10] and sparsification [3], [11]–[17]. The quantization based methods [7]–[10] aim to represent each element with fewer bits, such as QSGD [8] and signSGD [10]. Optimal compression ratio allocation for quantization is considered in [4]. Other studies [3], [11]–[17] apply sparsification to transmit a small subset of gradients so that the communication overhead can be reduced dramatically. However, the aforementioned works assign identical compression ratios to heterogeneous clients and thus the stragglers with poor channel conditions will become the bottleneck of model training. The authors in [18] provide client-specific compression schemes according to communication heterogeneity. However, they do not consider statistical heterogeneity and thus exhibit poor performance in the presence of non-IID data, in terms of model accuracy and convergence rate.

On the other hand, client selection plays a critical role in FL and has been extensively studied in previous works. In the common implementation, clients are selected uniformly at random or proportional to local dataset size [1], [30], which results in poor training performance and long latency due to non-IID data and capability heterogeneity [29]. Considering the statistical property, some sampling methods have investigated different criteria to evaluate the importance of clients, such as local loss [19], test accuracy [20], model updates [6], [21], [22], client correlations [23], and local data variability [24]. However, the above strategies ignore the heterogeneity of clients' capabilities and may suffer from the straggler effect. Other studies [5], [25]–[28] have designed client selection schemes that tackle heterogeneous system resources for fast convergence, but non-IID data still hurt the model accuracy. A very recent work [29] optimizes client selection probabilities while accounting for both data and capability heterogeneity. In this solution, the exchange of complete models incurs exorbitant communication cost and the obtained probabilities cannot be adaptively adjusted as training progresses, which ignores time-varying network conditions and thus exhibits

less flexibility. Compared with the prior works, FedCG can *simultaneously* cope with the challenges of communication efficiency, network dynamics and client heterogeneity by joint optimization of client selection and gradient compression.

III. Preliminaries and Problem Formulation

A. Federated Learning Setup

The goal of FL is to train a high-quality model through a loose federation of clients, which is coordinated by the PS. We suppose there is a set $\mathcal{N} = \{1, 2, ..., N\}$ of clients participating in FL. Each client $n \in \mathcal{N}$ has its local dataset $\mathcal{D}_n$ with the size of $|\mathcal{D}_n|$. The local loss function of client n on the collection of data samples is defined as:

$$F_n(\mathbf{x}) = \frac{1}{|\mathcal{D}_n|} \sum_{\xi \in \mathcal{D}_n} f_n(\mathbf{x}; \xi), \tag{1}$$

where $\mathbf{x}$ is the model parameter vector and $f_n(\mathbf{x}; \xi)$ is the loss function calculated by a specific sample ξ. FL seeks to minimize the global loss function $F(\mathbf{x})$, which translates into the following optimization problem:

$$\min_{\mathbf{x}} \ F(\mathbf{x}) = \sum_{n=1}^{N} p_n F_n(\mathbf{x}), \tag{2}$$

where p_n represents the weight of client n with $\sum_{n=1}^{N} p_n = 1$ and can be set to $p_n = \frac{|\mathcal{D}_n|}{\sum_{i=1}^{N} |\mathcal{D}_i|}$.

As a primitive implementation and the most commonly studied FL algorithm, *FedAvg* [1] has been proposed to solve the problem in Eq. (2). Specifically, the optimization process consists of multiple communication rounds. At each round $k \in \{0, 1, ..., K-1\}$, the PS randomly selects M clients and sends the global model $\mathbf{x}^k$ to the set of selected clients $\mathcal{M}^k \subseteq \mathcal{N}$. By setting $\mathbf{x}_n^{k,0} = \mathbf{x}^k$, each client n independently trains the local model for H iterations:

$$\mathbf{x}_n^{k,j+1} = \mathbf{x}_n^{k,j} - \eta_k \nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j}), \ j = 0, 1, ..., H-1, \tag{3}$$

where η_k is the learning rate at round k, and $\xi_n^{k,j}$ is the sample selected by client n for local iteration j. After local training, each client n sends the model updates $\mathbf{G}_n^k = \sum_{j=0}^{H-1} \nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j})$ to the PS for global aggregation. However, unlike in a cloud data center, FedAvg might face a few fundamental challenges while training models on edge devices, such as limited communication resources, dynamic network conditions and heterogeneous client properties.

B. Heterogeneity-Aware Federated Learning Framework

To address these challenges, we propose a heterogeneity-aware FL framework, called FedCG. As shown in Alg. 1, the training process of our framework includes K rounds, and each round consists of the following phases.

- At the beginning of round k, FedCG adaptively selects a diverse subset of clients $\mathcal{M}^k$ considering statistical heterogeneity and determines different compression ratios for selected clients according to heterogeneous and time-varying capabilities. Then the PS sends the global model $\mathbf{x}^k$ and compression ratio θ_n^k to each client $n \in \mathcal{M}^k$.
- Each client n updates the received model over its local dataset for H iterations. Based on compression ratio θ_n^k, the client n compresses the original model updates $\mathbf{G}_n^k$ to obtain $\tilde{\mathbf{G}}_n^k$. Then the compressed model updates $\tilde{\mathbf{G}}_n^k$ that fit the capabilities of client n are uploaded to the PS.
- Upon receiving all model updates from selected subset, the PS obtains a new global model $\mathbf{x}^{k+1}$ by aggregating compressed model updates and starts the next round.

Algorithm 1: Training process of FedCG

```
for Each round k = 0, 1, ..., K − 1 do
    The PS selects a diverse subset of clients M^k with |M^k| = M;
    The PS determines differnet compression raio θ_n^k for each client n ∈ M^k;
    The PS sends the current global model x^k and compression raio θ_n^k to each client n ∈ M^k;
    for Each client n ∈ M^k in parallel do
        x_n^{k,0} = x^k;
        for Each local iteration j = 0, 1, ..., H − 1 do
            x_n^{k,j+1} = x_n^{k,j} − η_k ∇F_n(x_n^{k,j}; ξ_n^{k,j});
        Compress the model updates G_n^k to obtain G̃_n^k according to compression raio θ_n^k;
        Upload G̃_n^k to the PS;
    The PS updates the global model x^{k+1} = x^k − (η_k / M) Σ_{n∈M^k} G̃_n^k;
```

Next, we detail the two main innovations of our framework, *i.e.*, client selection and gradient compression.

(1) Client Selection. Considering limited communication bandwidth and client availability, we select a fraction of clients to participate in training, which effectively reduces communication overhead. However, the clients located in geographically distinct regions generate data from different distributions (*i.e.*, non-IID) in practice. Many clients may provide similar and redundant gradient information for aggregation, which cannot reflect the true data distribution in the global view. Selecting such clients will waste resources and cause the global model to be biased towards certain clients, thus exacerbating the negative impact of non-IID data on training performance. To this end, we introduce diversity to client selection and select representative clients out of the whole while adhering to resource constraints [22]. We expect to find a diverse subset of clients $\mathcal{M}^k$ whose aggregated model updates approximate the (logically) aggregated updates of all clients. By encouraging diversity in model updates, we reduce redundant communication and increase the impact of under-represented clients that contribute different information, thereby promoting fairness. In this way, FedCG will counterbalance the bias introduced by non-IID data and speed up convergence.

(2) Gradient Compression. Gradient compression is another commonly adopted solution to alleviate network pressure due to its practicality and substantial bandwidth efficiency. Among previous gradient compression techniques, Top-k is a

promising compression operator with empirical and theoretical studies [15], [31]. Only the gradients with larger absolute values are required to transmit for global aggregation, which can sparsify the local gradients to only 0.1% density without impairing model convergence or accuracy [31]. Therefore, we adopt Top-k sparsification in this paper due to its efficiency and simplicity. It is worth mentioning that other compression operators (*e.g.*, Random-k sparsification [12] and quantization [8]) can also be compatible with our framework. Based on compression ratio θ_n^k, the client n selects the gradient elements with larger absolute values from the original model updates $\mathbf{G}_n^k$, and zero-out other unselected gradient elements to obtain compressed model updates $\tilde{\mathbf{G}}_n^k$. Moreover, we apply the error compensation mechanism [18] onto FedCG, which is widely used along with compression to further improve training performance. Error compensation accumulates the error from only uploading compressed gradients, thereby ensuring that all elements of the full gradient have a chance to be aggregated.

The compression ratio θ_n^k can be regarded as a measure of the sparsity, where a smaller θ_n^k corresponds to a more sparse vector and requires less communication and vice versa. More importantly, unlike the existing works unifying the sparsity levels of clients, FedCG assigns *different* compression ratios to the selected clients $\mathcal{M}^k$ considering their heterogeneous and time-varying capabilities. Specifically, the clients with excellent capabilities (*i.e.*, short completion time) are expected to adopt slight gradient compression, while the others with poor capabilities (*i.e.*, long completion time) should compress the gradients more aggressively. As a result, the selected clients will achieve approximately identical per-round completion time. Adaptive gradient sparsification dramatically saves the communication cost and mitigates the impact of stragglers, which provides substantial benefits in improving training efficiency, particularly in the resource-limited and heterogeneous wireless environments envisioned for FL.

C. Problem Formulation

We define the joint optimization problem of client selection and compression ratio decision as below. Let $T_{n,cmp}^k$ denote the computation time required for client n to perform one local iteration. At round k, the local training time of client n is $HT_{n,cmp}^k$ [32], where H is the number of local iterations between two consecutive global synchronizations. Following the prior works [15], we only consider the uplink communication, since the downlink speed in FL is much faster compared with the uplink and the parameter download time is negligible [33]. The communication time of client n at round k can be formulated as:

$$T_{n,com}^k = \frac{\theta_n^k R}{C_n^k}, \tag{4}$$

where R represents the size of original (uncompressed) model updates and C_n^k represents the upload speed of client n at round k. The upload speed changes dynamically as the training progresses. For client n, the total time T_n^k of local training and transmitting parameters at round k is expressed as:

$$T_n^k = HT_{n,cmp}^k + T_{n,com}^k. \tag{5}$$

In the synchronous FL, the per-round time is determined by the "slowest" one among the selected clients. The completion time of round k is defined as:

$$T^k = \max_{n \in \mathcal{M}^k} T_n^k. \tag{6}$$

We aim to select the clients involved in FL and determine the compression ratios for those selected clients. The optimization problem can be formulated as:

$$\min \ F(\mathbf{x}^K)$$
$$s.t. \begin{cases} \sum_{k=0}^{K-1} T^k < T \\ \left|\mathcal{M}^k\right| = M, & \forall k \\ 0 < \theta_n^k \leq 1, & \forall n, \forall k \end{cases} \tag{7}$$

The first inequality guarantees the resource constraint where T denotes the total time budget for given K. The second set of inequalities indicates that the PS selects M clients participating in training at each round. The third set of inequalities bounds the feasible range of compression ratios. The objective of the optimization problem is to minimize the loss function $F(\mathbf{x}^K)$ of model training given the resource constraint.

It is worth noting that our formulation can be extended to other "costs" beyond the completion time (*e.g.*, energy consumption) as well. In fact, it is non-trivial to directly solve the problem in Eq. (7) due to the unclear convergence relationship, accuracy-overhead trade-off, and tightly coupled nature. In the following section, we derive a tractable convergence rate to indicate how the selected clients and compression ratios affect the final convergence. On this basis, we develop a joint optimization algorithm to solve the coupled problem.

IV. Convergence Analysis

In this section, we provide the convergence analysis of the proposed framework. We first state the following assumptions on the local loss functions.

Assumption 1. *$F_1, F_2, ..., F_N$ are all L-smooth, i.e., given $\mathbf{x}$ and $\mathbf{y}$, $F_n(\mathbf{x}) \leq F_n(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^T \nabla F_n(\mathbf{y}) + \frac{L}{2}\left\|\mathbf{x}-\mathbf{y}\right\|^2$.*

Assumption 2. *$F_1, F_2, ..., F_N$ are all μ-strongly convex, i.e., given $\mathbf{x}$ and $\mathbf{y}$, $F_n(\mathbf{x}) \geq F_n(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^T \nabla F_n(\mathbf{y}) + \frac{\mu}{2}\left\|\mathbf{x}-\mathbf{y}\right\|^2$.*

Assumption 3. *The variance of the stochastic gradients on random data samples is bounded, i.e., $\mathbb{E}[\|\nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j}) - \nabla F_n(\mathbf{x}_n^{k,j})\|^2] \leq \sigma^2, \forall n, \forall j, \forall k$.*

Assumption 4. *The stochastic gradients on random data samples are uniformly bounded, i.e., $\|\nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j})\|^2 \leq G^2, \forall n, \forall j, \forall k$.*

These assumptions hold for typical FL models and are common in the convergence analysis literature [4], [22], [29], [30], [34]. Although our convergence analysis focuses on strong convex problems, the experimental results demonstrate that proposed framework also works well for non-convex learning problems. Furthermore, we use $\Gamma = F^* - \sum_{n=1}^N p_n F_n^*$ to quantize the degree of non-IID data distribution on clients, where F^* and F_n^* are the optimal values of F and F_n, respectively. If the data across clients follow IID, then Γ obviously goes to zero as the number of samples grows. Inspired by [30], we flatten local iterations at each communication round and

use $\mathbf{y}_n^{k,j+1} = \mathbf{x}_n^{k,j} - \eta_k \nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j})$ to represent the result of a local iteration on client n. If $j+1 < H$, $\mathbf{x}_n^{k,j+1} = \mathbf{y}_n^{k,j+1}$; otherwise, $\mathbf{x}_n^{k+1,0} = \mathbf{x}_n^{k,0} - \frac{\eta_k}{M} \sum_{n \in \mathcal{M}^k} \mathbf{G}_n^k$ and $\mathbf{y}_n^{k+1,0} = \mathbf{y}_n^{k,H}$. Moreover, we define $\bar{\mathbf{x}}^{k,j} = \sum_{n=1}^{N} p_n \mathbf{x}_n^{k,j}$, $\bar{\mathbf{y}}^{k,j} = \sum_{n=1}^{N} p_n \mathbf{y}_n^{k,j}$, $\bar{\mathbf{x}}^k = \bar{\mathbf{x}}^{k,0}$ and $\bar{\mathbf{y}}^k = \bar{\mathbf{y}}^{k,0}$.

At round k, we need to find a subset $\mathcal{M}^k$ of clients whose aggregated gradients can approximate the full gradients over all the N clients. We assume that there is a mapping $\pi^k : \mathcal{N} \to \mathcal{M}^k$ such that the gradients from client $n \in \mathcal{N}$ can be approximated by the gradients from a selected client $\pi^k(n) \in \mathcal{M}^k$. Let $\mathcal{A}_n^k = \{i \in \mathcal{N} | \pi^k(i) = n\}$ be the set of clients approximated by client $n \in \mathcal{M}^k$ and $\gamma_n^k = |\mathcal{A}_n^k|$. Then we define the approximation error at round k as:

$$\alpha_k = \left\| \frac{1}{N} \sum_{n \in \mathcal{M}^k} \gamma_n^k \nabla F_n(\mathbf{y}_n^{k,0}) - \frac{1}{N} \sum_{n \in \mathcal{N}} \nabla F_n(\mathbf{y}_n^{k,0}) \right\|, \quad (8)$$

which is used to characterize how well the aggregated gradients of selected client subset $\mathcal{M}^k$ approximate the full gradients. Besides, we define the compression error as:

$$\beta_n^k = \|\tilde{\mathbf{G}}_n^k - \mathbf{G}_n^k\|, \quad (9)$$

which indicates the difference between the compressed gradients $\tilde{\mathbf{G}}_n^k$ and the original gradients $\mathbf{G}_n^k$ of client n at round k. The approximation error and compression error are related to client selection and compression ratio decision policies. Their impact on final accuracy will be quantified in the next theorem.

Lemma 1. *Under Assumptions 1-4, the proposed framework ensures*

$$\|\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{y}}^{k+1}\| \leq LGH^2 \eta_k^2 + \alpha_k H \eta_k + \frac{\eta_k}{M} \sum_{n \in \mathcal{M}^k} \beta_n^k.$$

Theorem 1. *Let Assumptions 1-4 and Lemma 1 hold, and $\mathbf{x}^*$ is the optimal model. We have the convergence rate:*

$$\mathbb{E}[\|\mathbf{x}^K - \mathbf{x}^*\|^2] \leq \mathcal{O}(\frac{1}{K}) + \mathcal{O}(\alpha) + \mathcal{O}(\beta),$$

where $\alpha = \max_k \{\alpha_k\}$ and $\beta = \max_{n,k} \{\beta_n^k\}$.

Due to the page limit, the detailed proof of Lemma 1 and Theorem 1 can be found in [35].

Remark: Theorem 1 reveals that the approximation error and compression error have a great impact on the convergence performance. Ideally, when we select all clients to participate in training (*i.e.*, $M = N$) and set the compression ratios of all clients as 1 (*i.e.*, without compression) at each round, the approximate error and compression error become 0. To maximize the final model accuracy for total K rounds, we should minimize the approximation error and compression error under resource constraints.

V. Algorithm Design

In this section, we show how to leverage the derived convergence rate in Theorem 1 to obtain client selection and gradient compression policies for heterogeneous FL systems, which is the crucial design in FedCG. We first introduce the overall joint optimization process (Section V-A) and then detail two core components of the proposed algorithm, *i.e.*, client selection strategy (Section V-B) and compression ratio decision strategy (Section V-C), which are designed by minimizing the approximation error and compression error, respectively.

Algorithm 2: Joint optimization algorithm at round k

1 Initialize $\mathcal{M}^k = \emptyset$ and $\theta_n^k = 0, \forall n$;
2 Initialize $\mathcal{N}' = \mathcal{N}$;
3 **for** *Each iteration* $i = 1, 2, ..., M$ **do**
4 Select a diverse set of clients $\mathcal{M}^{k,i}$ from $\mathcal{N}'$ via submodular maximization in Section V-B;
5 Decide compression ratios $\theta_n^{k,i}$ for selected clients by solving optimization problem in Section V-C;
6 **if** $\sum_{n \in \mathcal{M}^{k,i}} \theta_n^{k,i} > \sum_{n \in \mathcal{M}^k} \theta_n^k$ **then**
7 $\mathcal{M}^k \leftarrow \mathcal{M}^{k,i}$;
8 $\theta_n^k \leftarrow \theta_n^{k,i}$;
9 $n' = \arg\min_{n \in \mathcal{M}^{k,i}} \theta_n^{k,i}$;
10 $\mathcal{N}' \leftarrow \mathcal{N}' - \{n'\}$;
11 **return** $\mathcal{M}^k$ and θ_n^k;

A. Joint Optimization Process

The key insight behind FedCG is that client selection and compression ratio decision interact with each other. We cannot reach the optimal state by independently determining the client subset and compression ratios. Therefore, this coupled property raises the necessity for joint optimization. However, it is usually difficult to optimize both at the same time. While if we fix one decision and then optimize the other, both of which are greatly simplified. To this end, we propose an iteration-based algorithm to jointly optimize client selection and compression ratio decision for the tightly coupled problem.

As shown in Alg. 2, in each iteration, we first apply submodular maximization in Section V-B to select a diverse subset from candidate clients, thereby minimizing the approximation error (Line 4). The aggregated gradients of selected clients are a good approximation of the full gradients from all clients. Then we determine appropriate compression ratios for these clients by solving the optimization problem (14) in Section V-C (Line 5). If the derived solution contributes to the reduction of compression error, we update the current client subset and compression ratios (Lines 6-8). Then, we find the client with the smallest compression ratio in the subset, and remove it from the candidate clients of the next iteration (Lines 9-10). This design prevents the client with over-compressed gradients from participating in FL and affecting model accuracy. The iterative heuristic terminates after M iterations. We finally obtain the client subset $\mathcal{M}^k$ at round k and the corresponding compression ratios θ_n^k, $\forall n \in \mathcal{M}^k$ (Line 11).

Considering the coupled nature of the optimization problem, we apply an iteration-based algorithm to derive an efficient solution for client selection and compression ratio decision. Starting with the initialization, the client subset and com-

pression ratios of each round are optimized alternately via fixed-point iterations, thus optimizing the overall objective. It is worth noting that our algorithm can be completed in M iterations, which does not depend on the total number of clients (*i.e.*, N). Consequently, the algorithm overhead does not increase significantly with system scale and will be evaluated through the experiments in Section VI.

B. Client Selection Strategy

Based on theoretical analysis, we aim to select a subset of clients to minimize approximation error under resource constraints, thereby improving convergence performance. The approximation error reflects how well the aggregated gradients of the selected clients approximate the gradients from all clients. To this end, we introduce diversity to client selection so that the selected clients can be representative of all clients [22]. Submodular functions have been widely adopted to measure diversity [36], [37]. Formally, for any subset $\mathcal{S} \subseteq \mathcal{U} \subseteq \mathcal{N}$ and $z \in \mathcal{N} \setminus \mathcal{U}$, a set function Q is submodular if $Q(\mathcal{S} \cup \{z\}) - Q(\mathcal{S}) \geq Q(\mathcal{U} \cup \{z\}) - Q(\mathcal{U})$, which indicates z is more valuable for a smaller set $\mathcal{S}$ than for a larger set $\mathcal{U}$. The marginal gain of z for a subset $\mathcal{S}$ is denoted as $Q(\mathcal{S} \cup \{z\}) - Q(\mathcal{S})$. All submodular functions have the diminishing return property, *i.e.*, the marginal gain that an element brings to a subset diminishes as more elements are added to the subset. Thanks to the diminishing return property, maximizing submodular functions effectively promotes diversity and reduces the redundancy [37].

Inspired by the above facts, our algorithm minimizes the approximation error by applying submodular maximization to select diverse clients. Based on triangular inequality, we can derive an upper bound of the approximation error:

$$\begin{aligned}&\left\|\frac{1}{N}\sum_{n\in\mathcal{N}}\nabla F_n(\mathbf{y}_n^{k,0}) - \frac{1}{N}\sum_{n\in\mathcal{M}^k}\gamma_n^k\nabla F_n(\mathbf{y}_n^{k,0})\right\| \\ \leq&\frac{1}{N}\sum_{n\in\mathcal{N}}\left\|\nabla F_n(\mathbf{y}_n^{k,0}) - \nabla F_{\pi^k(n)}(\mathbf{y}_{\pi^k(n)}^{k,0})\right\|. \end{aligned} \tag{10}$$

Eq. (10) is minimized when the mapping π^k assigns each $n \in \mathcal{N}$ to a client in $\mathcal{M}^k$ with the most gradient similarity:

$$\begin{aligned}&\left\|\frac{1}{N}\sum_{n\in\mathcal{N}}\nabla F_n(\mathbf{y}_n^{k,0}) - \frac{1}{N}\sum_{n\in\mathcal{M}^k}\gamma_n^k\nabla F_n(\mathbf{y}_n^{k,0})\right\| \\ \leq&\frac{1}{N}\sum_{n\in\mathcal{N}}\min_{i\in\mathcal{M}^k}\left\|\nabla F_n(\mathbf{y}_n^{k,0}) - \nabla F_i(\mathbf{y}_i^{k,0})\right\| = Q(\mathcal{M}^k). \end{aligned} \tag{11}$$

Minimizing the upper bound $Q(\mathcal{M}^k)$ of the approximation error or maximizing $\bar{Q}(\mathcal{M}^k)$ (a constant minus its negation) is essentially equivalent to maximizing a well-known submodular function, *i.e.*, facility location function [38]. Considering the constraint $\left|\mathcal{M}^k\right| = M$, the approximation error minimization problem can be transformed into a submodular maximization problem under cardinality constraint, which is NP-hard [36]. Fortunately, the greedy algorithm has been proven to be effective in solving submodular maximization problem and provides $(1-e^{-1})$-approximation to the optimal solution [39].

At round k, the greedy algorithm for minimizing the approximation error starts with an empty set $\mathcal{M}^k = \emptyset$. From the candidate set $\mathcal{N}$, the client $n \in \mathcal{N} \setminus \mathcal{M}^k$ with the largest marginal gain is constantly added to $\mathcal{M}^k$ until $\left|\mathcal{M}^k\right| = M$:

$$\mathcal{M}^k \leftarrow \mathcal{M}^k \cup \{n^*\}, n^* = \arg\max_{n\in\mathcal{N}\setminus\mathcal{M}^k}[\bar{Q}(\mathcal{M}^k \cup \{n\}) - \bar{Q}(\mathcal{M}^k)]. \tag{12}$$

The computational complexity of the greedy algorithm is $\mathcal{O}(N \cdot M)$. However, in practice, the complexity can be reduced to $\mathcal{O}(N)$ using stochastic greedy algorithms [40], and further improved by lazy evaluation [37] and distributed implementations [41]. Consequently, the algorithm overhead can be negligible compared to the massive overhead for model training and transmission [36], which is empirically verified in Section VI. Besides, it is infeasible for the PS to collect the gradients from all clients for marginal gain calculation. For the clients whose gradients have not been collected at the current round, we estimate the marginal gain with their historical gradient information [22]. In a nutshell, we relate the gradient approximation to submodular maximization. According to the marginal gain of the submodular function, we select the diverse subset of clients to minimize the approximation error between the estimated and the full gradients.

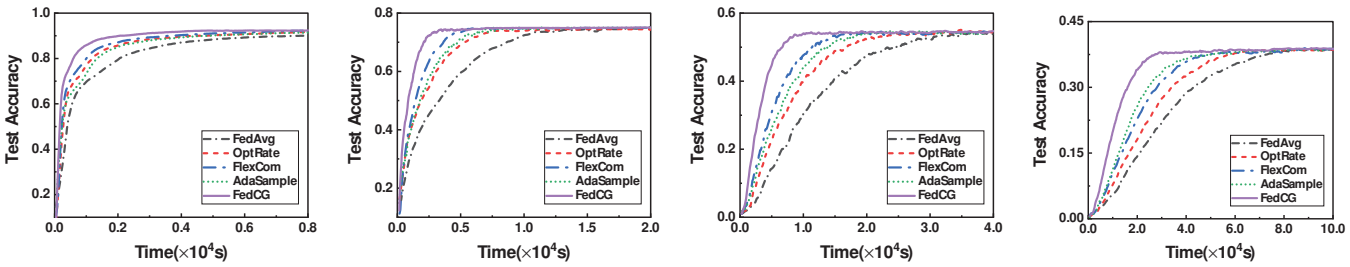
C. Compression Ratio Decision Strategy

We also need to determine different compression ratios for selected clients so as to minimize the compression error, which characterizes the difference between the compressed gradients and the original gradients. The compression error satisfies the following contraction property [12]:

$$\begin{aligned}\mathbb{E}[\|\tilde{\mathbf{G}}_n^k - \mathbf{G}_n^k\|^2] &\leq (1-\theta_n^k)\|\mathbf{G}_n^k\|^2 \\ &\leq (1-\theta_n^k)H^2G^2. \end{aligned} \tag{13}$$

The compression ratio has two contrasting effects on the training process. The larger compression ratio can preserve more information from the original gradients, which reduces the compression error and thus ensures the model accuracy. However, the communication overhead is still high under these circumstances. Conversely, the smaller compression ratio will contribute to reducing communication overhead, but it leads to higher compression error and is more likely to deteriorate the model accuracy. To strike a judicious trade-off between resource overhead and model accuracy, we aim to minimize the compression error under given resource constraints.

However, the PS requires complete information of the entire training process (*e.g.*, network conditions) to determine optimal compression ratios for selected clients. Unfortunately, the communication conditions of wireless links are usually time-varying due to network bandwidth reallocation and clients' mobility, and it is usually impossible to obtain this information in advance. To overcome the unavailability of future information, we divide the long-term optimization problem into a series of one-shot problems. Given the remaining resources, we online determine the compression ratios for the current round. Thus, the compression ratios can be continuously



(a) LR over MNIST (b) AlexNet over CIFAR-10 (c) ResNet9 over CIFAR-100 (d) ResNet18 over Tiny-ImageNet

Fig. 1. Training performance of different methods on the prototype system.

adjusted to accommodate system dynamics without requiring future network conditions as prior knowledge. Accordingly, compression ratio decision problem at round k is expressed as:

$$\min \sum_{n\in\mathcal{M}^k} (1-\theta_n^k)H^2G^2$$
$$s.t. \begin{cases} \sum_{i=0}^{k-1} T^i + (K-k)T^k < T \\ 0 < \theta_n^k \leq 1, \end{cases} \quad \forall n, \forall k \tag{14}$$

Since the above optimization problem is a linear programming (LP) problem, it can be optimally solved using LP solver (*e.g.*, PuLP [42]). By solving the problem in Eq. (14), FedCG assigns different compression ratios to the selected clients according to their heterogeneous and time-varying capabilities. Consequently, these clients adaptively compress and upload the gradients under resource constraints, preventing the ones with poor capabilities from becoming the bottleneck of FL.

VI. Performance Evaluation

A. Experimental Setup

Experimental Platforms. We evaluate the performance of FedCG in both a physical testbed and a simulated environment. The prototype system helps us capture real-world resource overhead (*e.g.*, time and traffic consumption) and the simulation system is used to evaluate larger-scale FL scenarios with manipulative parameters. *(1) Testbed settings:* The hardware prototype system consists of an AMAX deep learning workstation as the PS and 30 commercial embedded devices as the clients. Specifically, the workstation is carrying one 8-core Intel Xeon CPU and 4 NVIDIA GeForce RTX 2080Ti GPUs. The clients include 10 NVIDIA Jetson AGX devices, 10 NVIDIA Jetson NX devices, and 10 NVIDIA Jetson TX2 devices[1], which reflect capability heterogeneity of clients. All devices are interconnected via a commercial WiFi router and we develop a TCP-based socket interface for communication between the PS and clients. *(2) Simulation settings:* We simulate an FL system with 100 virtual clients (each is implemented as a process in the system). To reflect heterogeneous and dynamic network conditions, we fluctuate each client's inbound bandwidth between 10Mb/s and 20Mb/s. Considering that the outbound bandwidth is typically smaller than the inbound bandwidth in a typical WAN, we set it to fluctuate between 1Mb/s and 5Mb/s [34] and randomly change it every round. For computation heterogeneity, the time of one local iteration follows a Gaussian distribution whose mean and variance are from the measurements of the prototype.

[1]https://developer.nvidia.com/embedded/community/support-resources

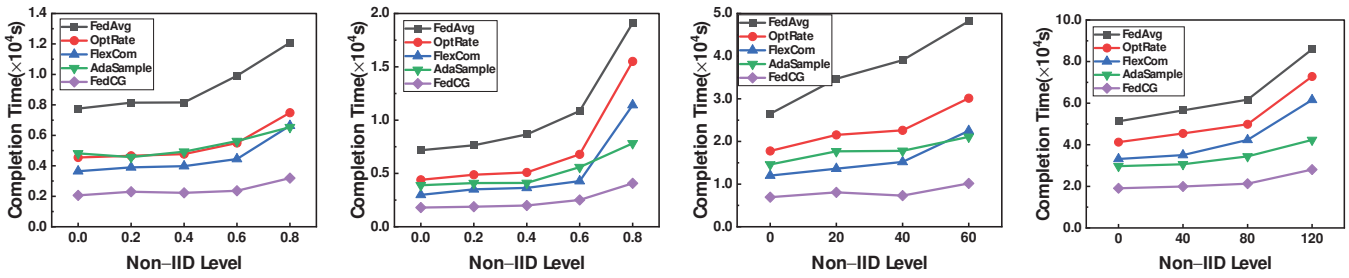
TABLE I
Resource overhead of different methods to achieve the target accuracy.

Datasets	Metrics	FedAvg	OptRate	FlexCom	AdaSample	**FedCG**
MNIST (Acc=90%)	Time(s)	7754.2	4545.1	3647.2	4818.1	**2057.5**
	Traffic(MB)	4368.1	1556.4	898.9	4072.5	**874.7**
CIFAR-10 (Acc=74%)	Time(s)	15932.1	9696.9	5334.3	6967.6	**3261.8**
	Traffic(MB)	15198.6	6192.8	2674.2	11983.6	**2480.3**
CIFAR-100 (Acc=54%)	Time(s)	35047.9	24520.5	17726.2	19722.6	**10068.7**
	Traffic(MB)	32550.9	15581.5	8583.2	34569.6	**8401.7**
Tiny-ImageNet (Acc=37%)	Time(s)	68931.7	54918.1	45543.9	41717.9	**25614.4**
	Traffic(MB)	53271.6	33242.0	21107.4	55849.3	**19958.7**

Datasets and Models. We conduct the experiments over four datasets (*i.e.*, MNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet), which represent a large variety of the small, middle and large training tasks in practical FL scenarios. We adopt the *convex* logistic regression (LR) model for MNIST and *non-convex* deep neural networks (*e.g.*, AlexNet, ResNet9, and ResNet18) for the other three datasets.

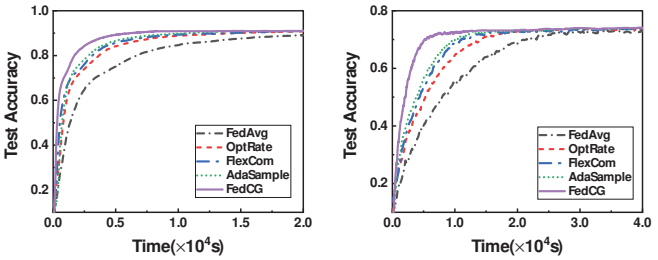
Data Distribution. To simulate various degrees of statistical data heterogeneity, we adopt two different non-IID partition schemes, *i.e.*, latent Dirichlet allocation (LDA) and skewed label, which are widely used in previous works [6], [29], [34]. *(1) LDA for MNIST and CIFAR-10:* ψ (ψ = 0.2, 0.4, 0.6, and 0.8) of the data on each client belong to one class and the remaining $1-\psi$ samples belong to other classes. *(2) Skewed label for CIFAR-100 and Tiny-ImageNet:* Each client lacks ψ classes of data samples, where ψ = 20, 40, and 60 for CIFAR-100, and ψ = 40, 80, and 120 for Tiny-ImageNet. In particular, we use $\psi = 0$ to denote IID data. Except for the experiments on non-IID data, we shuffle the data and uniformly divide them among all clients.

Benchmarks. We compare the proposed framework with four benchmarks. *(1) FedAvg* [1] selects clients uniformly at random and exchanges the entire models between the PS and selected clients. *(2) OptRate* [4] adopts compression to reduce communication overhead and determines identical compression rates for clients at each round to seek the trade-



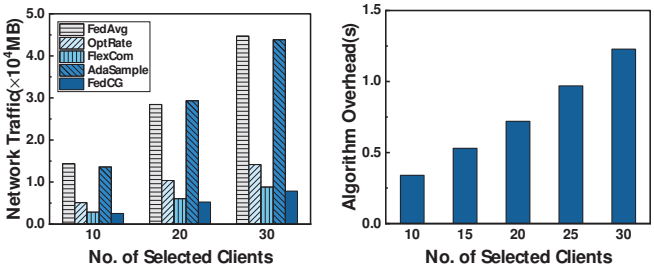
(a) LR over MNIST (b) AlexNet over CIFAR-10 (c) ResNet9 over CIFAR-100 (d) ResNet18 over Tiny-ImageNet

Fig. 2. Completion time under different levels of non-IID data.



(a) LR over MNIST (b) AlexNet over CIFAR-10

Fig. 3. Training performance in dynamic and heterogeneous simulation environments.



(a) Network Traffic (b) Algorithm Overhead

Fig. 4. Effect of the number of selected clients on network traffic and algorithm overhead.

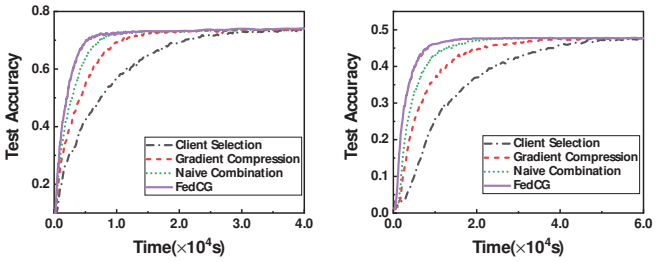
off between overhead and accuracy. *(3) FlexCom* [18] enables flexible compression control and allows clients to compress the gradients to different levels considering the heterogeneity in communication capabilities. *(4) AdaSample* [29] optimizes client sampling probabilities to tackle both system and statistical heterogeneity so as to minimize FL completion time. Since we concentrate on improving the training efficiency of FL regarding resource constraints, training models to achieve state-of-the-art accuracy is beyond the scope of this work. Unless otherwise specified, we select $M = 10$ clients to participate in training and the clients perform $H = 50$ local iterations at each round.

B. Testbed Results

Training Performance. We first compare the training performance of FedCG and other methods on the prototype system. The accuracy results with respect to training time are presented in Fig. 1. We observe that FedCG achieves a comparable accuracy and converges much faster than the other methods for all four datasets. Compared to the benchmarks, FedCG can provide up to 3.8× speedup for LR over MNIST, 4.9× speedup for AlexNet over CIFAR-10, 3.5× speedup for ResNet9 over CIFAR-100, and 2.7× speedup for ResNet18 over Tiny-ImageNet. Furthermore, the accuracy of FedCG always surpasses the other benchmarks after a given time. In particular, our framework achieves 74.94% accuracy after training AlexNet over CIFAR-10 for 10,000s while that of FedAvg, OptRate, FlexCom and AdaSample is 72.49%, 74.01%, 74.83% and 74.81%, respectively. These results demonstrate the advantages of FedCG for both convex and non-convex learning tasks.

Resource Overhead. To validate the efficiency of FedCG, we record the resource overhead of different methods when they attain the target accuracy in Table I, including completion time and traffic consumption. Note that the target accuracy is set as the accuracy that all methods can achieve. As summarized in Table I, compared with the benchmarks, FedCG reduces the training time by 55.4% and network traffic by 50.3% on average for reaching the same accuracy. The reasons for such superior performance are as follows. Compared to FedAvg and AdaSample, the solutions with model compression (*i.e.*, OptRate, FlexCom and FedCG) can save much more network traffic. However, OptRate assigns identical compression ratios to heterogeneous clients, which exacerbates the straggler effect. Although FlexCom and FedCG achieve similar traffic consumption by assigning different compression ratios to clients, FlexCom still produces a long training completion time without considering the computation heterogeneity. In addition, we note that optimizing sampling probabilities like AdaSample can improve training efficiency to a certain extent, but cannot essentially address heterogeneity challenges since the clients with large capability gaps may participate in FL at the same round. By contrast, with the assistance of adaptive client selection and gradient compression, FedCG brings significant savings in both time and traffic consumption, thereby effectively accelerating the training process of FL.

Effect of Non-IID Data. We proceed to investigate how our proposed framework performs under statistical heterogeneity. Fig. 2 depicts the required time for FedCG and benchmarks to reach the target accuracy under different levels of non-IID data. We set the target accuracy of LR, AlexNet, ResNet9 and ResNet18 as 90%, 67%, 51% and 33%, respectively. As shown



(a) AlexNet over CIFAR-10 (b) ResNet9 over CIFAR-100

Fig. 5. Training performance of independent decision, naive combination and joint optimization.

in Fig. 2, all methods suffer from performance degradation with the increasing skewness of data distribution. Nevertheless, FedCG only has the slightest increase in completion time compared to the other benchmarks and exhibits robustness to non-IID data. The advantage of FedCG is attributed to diverse client selection which increases the impact of under-represented clients, thereby promoting fairness and reducing the bias introduced by non-IID data. In addition, the savings of resource overhead in FedCG further enlarge the performance gap as the non-IID level increases. The speedup provided by FedCG is 4.0×, 4.1×, 4.3×, 4.3×, and 4.7× with non-IID level varying from 0 to 0.8 for AlexNet over CIFAR-10, indicating the effectiveness of FedCG for data heterogeneity.

C. Simulation Results

Dynamic and Heterogeneous Environments. To evaluate the performance of FedCG in large-scale FL scenarios, we conduct our experiments by simulations with 100 clients. Fig. 3 plots the accuracy curve of different methods in dynamic and heterogeneous environments. We find that our proposed framework still substantially outperforms the other benchmarks in large-scale FL scenarios and exhibits faster convergence without sacrificing accuracy. For instance, FedCG takes 5,170s to achieve 70% accuracy for AlexNet over CIFAR-10, while the completion time of FedAvg, OptRate, FlexCom and AdaSample are 22,009s, 14,225s, 11,129s and 10,392s, respectively. The corresponding speedups are 4.3×, 2.8×, 2.2× and 2.0×. Such performance gain of FedCG is rooted in appropriate client selection strategy and different compression ratios. This not only excludes the clients with poor capabilities from training but also allows each selected client to transmit compressed gradients fitting its capabilities. Moreover, the decisions including client subset and compression ratio are continuously adjusted during training to adapt to the time-varying capabilities of clients. The above simulation results strongly verify the usability of our design in highly dynamic and heterogeneous environments.

Varying the Number of Selected Clients. We further conduct the simulation experiments to analyze the influence of the number of selected clients (*i.e.*, M) on the training efficiency. Firstly, we compare the traffic consumption of different methods for achieving the target accuracy (*e.g.*, 90%) when the number of selected clients increases from 10 to 30. The results for LR over MNIST are shown in Fig. 4(a). Apparently, network traffic of all methods increases gradually with M ranging from 10 to 30. This is expected because more clients participate in training at each round, which will consume more communication resources to transmit model updates. Nevertheless, FedCG outperforms other methods under different numbers of selected clients and reduces network traffic consumption by about 11.2-82.4% compared with the four benchmarks. Secondly, we measure the decision overhead of the proposed joint optimization algorithm with various numbers of selected clients, which is illustrated in Fig. 4(b). Although the algorithm overhead becomes larger as M increases, the maximum overhead is only 1.2s, which is much smaller than the FL training and transmission time (*e.g.*, hundreds of seconds) and thus can be ignored. These results suggest that the iterative optimization process of the proposed algorithm incurs a small decision overhead and will not hinder the practical deployment of FedCG in FL.

Necessity of Joint Optimization. Instead of simple combination, FedCG aims to achieve efficient FL by joint optimization of client selection and gradient compression. To indicate the importance of the proposed joint optimization algorithm, we compare the training performance of independent decision, naive combination and FedCG. As shown in Fig. 5, it is clear that FedCG consistently converges faster than the other three methods without loss of accuracy. Our framework provides 1.4-4.1× speedup to reach target accuracy (*e.g.*, 70%) for AlexNet over CIFAR-10 and 1.5-4.5× speedup to reach target accuracy (*e.g.*, 45%) for ResNet9 over CIFAR-100. The explanation for this phenomenon is that client selection and compression ratio decision are tightly coupled. Consequently, neither independent decision nor naive combination can handle network dynamics and client heterogeneity well, thus negatively affecting the convergence performance. FedCG overcomes this issue by proposing an iteration-based algorithm and demonstrates impressive performance improvement, which emphasizes the necessity of joint optimization process.

VII. Conclusion

In this paper, we propose a novel framework, called FedCG, to achieve efficient FL with adaptive client selection and gradient compression. Specifically, FedCG selects a diverse set of clients and assigns different compression ratios to selected clients considering their heterogeneous and time-varying capabilities. We jointly optimize client selection and compression ratio decision, which address the challenges of FL on communication efficiency, network dynamics and client heterogeneity. Experimental results demonstrate the advantages and effectiveness of the proposed framework.

Acknowledgment

This research was supported in part by the National Key Research and Development Program of China (Grant No. 2021YFB3301501); in part by the National Science Foundation of China (NSFC) under Grants 61936015, and 62102391 and 62132019; in part by the Open Research of Projects of Zhejiang Lab under Grant 2022QA0AB04. C. Qian was partially supported by NSF grants 1750704 and 2114113.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] H. Xu, K. Kostopoulou, A. Dutta, X. Li, A. Ntoulas, and P. Kalnis, "Deepreduce: A sparse-tensor communication framework for federated deep learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 150–21 163, 2021.

[4] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1459–1468.

[5] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[6] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1698–1707.

[7] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[8] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.

[11] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 440–445.

[12] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[13] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[14] A. M. Abdelmoniem and M. Canini, "Dc2: Delay-aware compression control for distributed machine learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[15] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *International Conference on Learning Representations (ICLR)*, 2018.

[16] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 300–310.

[18] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

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

[20] I. Mohammed, S. Tabatabai, A. Al-Fuqaha, F. El Bouanani, J. Qadir, B. Qolomany, and M. Guizani, "Budgeted online selection of candidate iot clients to participate in federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5938–5952, 2020.

[21] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[22] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *International Conference on Learning Representations (ICLR)*, 2022.

[23] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang, H. Li, and Y. Chen, "Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 102–10 111.

[24] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3095–3099.

[25] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1449–1458.

[26] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2020.

[27] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, 2020, pp. 125–136.

[28] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang, "Resource-efficient and convergence-preserving online participant selection in federated learning," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 606–616.

[29] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1739–1748.

[30] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020.

[31] S. Shi, K. Zhao, Q. Wang, Z. Tang, and X. Chu, "A convergence analysis of distributed sgd with communication-efficient gradient sparsification." in *IJCAI*, 2019, pp. 3411–3417.

[32] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[33] Y. Zhan and J. Zhang, "An incentive mechanism design for efficient edge learning by deep reinforcement learning approach," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2489–2498.

[34] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Transactions on Mobile Computing*, 2022.

[35] Z. Jiang, Y. Xu, H. Xu, Z. Wang, and C. Qian, "Adaptive control of client selection and gradient compression for efficient federated learning," *arXiv preprint arXiv:2212.09483*, 2022.

[36] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6950–6960.

[37] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization techniques*. Springer, 1978, pp. 234–243.

[38] G. Cornuejols, M. Fisher, and G. L. Nemhauser, "On the uncapacitated location problem," in *Annals of Discrete Mathematics*. Elsevier, 1977, vol. 1, pp. 163–177.

[39] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[40] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[41] B. Mirzasoleiman, M. Zadimoghaddam, and A. Karbasi, "Fast distributed submodular cover: Public-private data summarization," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[42] S. Mitchell, M. OSullivan, and I. Dunning, "Pulp: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand*, vol. 65, 2011.