Federated Learning With Client Selection and Gradient Compression in Heterogeneous Edge Systems

Yang Xu, *Member, IEEE*, Zhida Jiang, Hongli Xu, *Member, IEEE*, Zhiyuan Wang, Chen Qian, *Senior Member, IEEE*, and Chunming Qiao, *Fellow, IEEE*

***Abstract*—Federated learning (FL) has recently gained tremendous attention in edge computing and Internet of Things, due to its capability of enabling distributed clients to cooperatively train models while keeping raw data locally. However, the existing works usually suffer from limited communication resources, dynamic network conditions and heterogeneous client properties, which hinder efficient FL. To simultaneously tackle the above challenges, we propose a heterogeneity-aware FL framework, called FedCG, with adaptive client selection and gradient compression. Specifically, FedCG introduces diversity to client selection and aims to select a representative client subset considering statistical heterogeneity. These selected clients are assigned different compression ratios based on heterogeneous and time-varying capabilities. After local training, they upload sparse model updates matching their capabilities for global aggregation, which can effectively reduce the communication cost and mitigate the straggler effect. More importantly, instead of naively combining client selection and gradient compression, we highlight that their decisions are tightly coupled and indicate the necessity of joint optimization. We theoretically analyze the impact of both client selection and gradient compression on convergence performance. Guided by the convergence rate, we develop an iteration-based algorithm to jointly optimize client selection and compression ratio decision using submodular maximization and linear programming. On this basis, we propose the quantized extension of FedCG, termed Q-FedCG, which further adjusts quantization levels based on gradient innovation. Extensive experiments on both real-world prototypes and simulations show that FedCG and its extension can provide up to 6.4× speedup.**

***Index Terms*—Edge computing, federated learning, capability heterogeneity, statistical heterogeneity.**

I. Introduction

In RECENT years, the prosperity of edge computing and Internet of Things (IoT) results in the blowout of data being generated at the network edge [1]. The massive data from edge devices are of paramount importance for training machine learning models to improve the quality of services, also known as *edge intelligence* [2]. However, conventional centralized training paradigm requires local raw data to be aggregated in a central server for further processing, which is impractical due to limited network bandwidth and privacy concerns. Driven by such realistic issues, federated learning (FL) [3] has emerged as a distributed privacy-preserving training paradigm. In FL, multiple clients collaboratively train machine learning models under the orchestration of the parameter server (PS), while without explicitly sharing local data. The inherently distributed nature of FL makes it suitable for IoT and easy to implement in mobile edge networks. Such a paradigm can efficiently leverage local computing resources of edge devices and protect user privacy. With these technical advantages, FL can cover a wide range of intelligent applications, such as smart city, smart healthcare, transportation and automated systems [4].

Despite its practical effectiveness, there are several key challenges for the FL setting that make it difficult to train high-quality models in edge systems. *(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 [5]. *(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 [6]. 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

Manuscript received 5 March 2023; revised 15 August 2023; accepted 24 August 2023. Date of publication 28 August 2023; date of current version 4 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB3301500, in part by the National Science Foundation of China (NSFC) under Grants 61936015, 62132019, and 62102391, in part by the Jiangsu Province Science Foundation for Youths under Grant BK20210122 [DOI: 10.1109/INFOCOM53939.2023.10229029]. A preliminary version of this paper titled "Heterogeneity-Aware Federated Learning with Adaptive Client Selection and Gradient Compression" was accepted by IEEE INFOCOM 2023. Recommended for acceptance by X. Liu. *(Corresponding author: Hongli Xu.)*

Yang Xu, Zhida Jiang, Hongli Xu, and Zhiyuan Wang are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China, and also with Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, Jiangsu 215123, China (e-mail: xuyangcs@ustc.edu.cn; zdjiang@mail.ustc.edu.cn; xuhongli@ustc.edu.cn; cswangzy@mail.ustc.edu.cn).

Chen Qian is with the Department of Computer Science and Engineering, Jack Baskin School of Engineering, University of California, Santa Cru, CA 95064 USA (e-mail: cqian12@ucsc.edu).

Chunming Qiao is with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: qiao@buffalo.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2023.3309497, provided by the authors.

Digital Object Identifier 10.1109/TMC.2023.3309497

heterogeneity. The clients may be equipped with different computing chips and located in diverse regions, thus their capabilities vary significantly [7]. 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 typically not independent and identically distributed (non-IID). For instance, the images collected by the cameras reflect the demographics of each camera's location. Statistical heterogeneity will bring the biases in training and eventually cause an accuracy degradation of FL [8], [9].

To improve communication efficiency of FL, a natural solution is to reduce the size of transmitted payload. The existing works have adopted quantization [6], [10], [11], [12], [13], [14], [15], [16], [17] or sparsification [5], [18], [19], [20], [21], [22], [23], [24], [25] techniques to relax the communication load. But most 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 [26] do not take statistical data 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 [7], [8], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], most of which lack joint consideration of capability and statistical heterogeneity. Although some works take two types of heterogeneity into account [37], [38], the derived client selection solution cannot effectively adapt to network dynamics. Considering time-varying network conditions, previous literature focuses on dynamic resource management [39], [40], [41], compression control [21], [22] and topology construction [42], [43]. However, they overlook capability or statistical heterogeneity, leading to performance degradation in heterogeneous edge systems.

In summary, the 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 of FedCG, 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 Top-k sparsification 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 sparse 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 [30]. By encouraging diversity in client selection, FedCG can effectively reduce redundant communication and modulate 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 significantly 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. *First*, the quantitative relationship between client selection, gradient compression and model convergence is unclear. *Second*, the compression ratios have two contrasting effects on the training process. 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. *Third*, the tightly coupled problem of client selection and compression ratio decision adds additional challenges to algorithm design. To this end, we provide the convergence analysis and then design an iteration-based algorithm to jointly optimize client selection and compression ratio decision.

On the basis of FedCG, we further integrate quantization techniques into the heterogeneity-aware FL framework and propose the quantized extension, termed Q-FedCG. Q-FedCG not only adopts Top-k sparsification to transmit only a subset of gradient elements but also reduces the number of bits representing each element. Transmitting the sparse gradients in lower precision numerical formats can relieve the communication bottleneck from different perspectives. Since the selected clients may contribute differently to global convergence due to non-IID data, Q-FedCG performs multi-level quantification for clients based on gradient innovation. The gradients with significant innovation are allowed to transmit with more bits while the gradients with small innovation are represented with lower quantization resolution [17]. In this way, Q-FedCG can further reduce unnecessary communication and thus expedite the training process of FL. The key contributions of this paper are summarized 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. Moreover, we provide the quantized extension of FedCG.
- 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 and its extension can provide up to 6.4× speedup compared to state-of-the-art methods.

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

II. Preliminaries and Problem Formulation

In this section, we start by introducing the basics of FL. Then, we describe our proposed FL framework and finally formulate the optimization problem of client selection and gradient compression.

A. Federated Learning Basics

The goal of FL is to train a high-quality model through a loose federation of clients, which is coordinated by the PS. The main notations of this paper are summarized in Table I. We suppose there is a set $\mathcal{N} = \{1, 2, \ldots, N\}$ 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* [3] has been proposed to solve the problem in (2). Specifically, the optimization process consists of multiple communication rounds. At each round $k \in \{0, 1, \ldots, K-1\}$, 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

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

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.

TABLE I
Main Notations

Notation	Semantics
$\mathcal{N}$	Set of clients
N	Total number of clients
$\mathcal{D}_n$	Local dataset of client n
F_n	Local loss function of client n
p_n	Weight of client n
K	Total number of communication rounds
H	Number of local iterations at each round
$\mathcal{M}^k$	Set of selected clients at round k
M	Number of selected clients
$\mathbf{x}^k$	Global model at round k
$\mathbf{x}_n^{k,j}$	Local model of client n at iteration j of round k
$\xi_n^{k,j}$	Data sample of client n at iteration j of round k
η_k	Learning rate at round k
θ_n^k	Compression ratio of client n at round k
$\mathbf{G}_n^k$	Model updates of client n at round k
$\tilde{\mathbf{G}}_n^k$	Compressed model updates of client n at round k
T^k	Completion time of round k
T	Total time budget
Γ	Degree of non-IID data distribution
π^k	Mapping from set $\mathcal{N}$ to set $\mathcal{M}^k$
$\mathcal{A}_n^k$	Set of clients approximated by client $n \in \mathcal{M}^k$
γ_n^k	Cardinality of set $\mathcal{A}_n^k$
α_k	Approximation error at round k
β_n^k	Compression error of client n at round k
$V(\mathcal{M}^k)$	Upper bound of approximation error
l_n^k	Quantization level of client n at round k
$Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$	Quantized gradients of client n at round k
Δ_n^k	Actual aggregated gradients in Q-FedCG
Ψ^k	Impact of global updates in criterion of Q-FedCG
$\varepsilon_n^k(l)$	Quantization error of client n at round k under level l

B. Heterogeneity-Aware Federated Learning Framework

To address these challenges, we propose a heterogeneity-aware FL framework, called FedCG, as shown in Fig. 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 their 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 $\tilde{\mathbf{G}}_n^k$. The compressed model updates $\tilde{\mathbf{G}}_n^k$ that fit the capabilities of client n are uploaded to the PS.

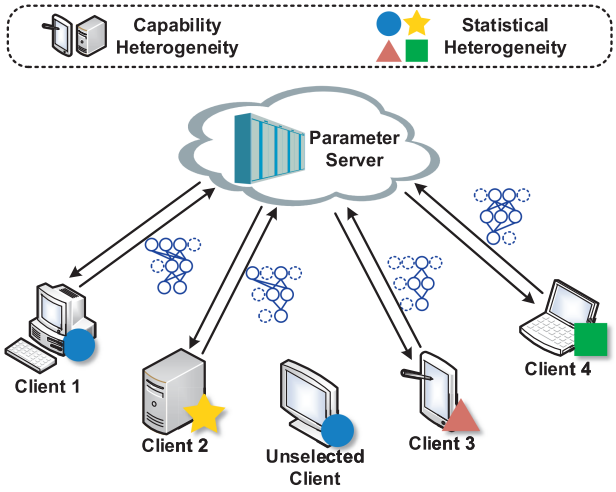


Fig. 1. Overview of FedCG. The PS selects a representative client subset considering statistical heterogeneity. After local training, these selected clients upload model updates of different sizes, which match their capabilities.

- 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.

The complete algorithm of the training process is presented in Algorithm 1. 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 only 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 [30]. Specifically, 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. 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 [25]. Instead of the entire model updates, only a small portion of gradients are required to transmit for global aggregation. Among previous gradient sparsification techniques, Top-k is a promising compression operator, which can sparsify the local gradients to only 0.1% density without impairing model convergence or accuracy [44]. The main idea of Top-k sparsification is based on the fact that gradients with larger absolute values contribute more to model convergence, which has been theoretically and empirically verified in [23], [44], [45]. 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 [19]) 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 [26] 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.

Algorithm 1: Training Process of FedCG.

for *Each round* $k = 0, 1, ..., K-1$ **do**
  The PS selects a diverse subset of clients $\mathcal{M}^k$ with $|\mathcal{M}^k| = M$;
  The PS determines differnet compression raio θ_n^k for each client $n \in \mathcal{M}^k$;
  The PS sends the current global model $\mathbf{x}^k$ and compression raio θ_n^k to each client $n \in \mathcal{M}^k$;
  for *Each client* $n \in \mathcal{M}^k$ *in parallel* **do**
    $\mathbf{x}_n^{k,0} = \mathbf{x}^k$;
    for *Each local iteration* $j = 0, 1, ..., H-1$ **do**
      $\mathbf{x}_n^{k,j+1} = \mathbf{x}_n^{k,j} - \eta_k \nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j})$;
    Select gradient elements with larger absolute values considering error compensation;
    Obtain compressed model updates $\tilde{\mathbf{G}}_n^k$ according to compression raio θ_n^k;
    Upload $\tilde{\mathbf{G}}_n^k$ to the PS;
    Accumulate error for gradient elements that are not uploaded;
  The PS updates the global model $\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{\eta_k}{M} \sum_{n \in \mathcal{M}^k} \tilde{\mathbf{G}}_n^k$;

C. Problem Formulation

We define the joint optimization problem of client selection and compression ratio decision as below. Let $T^k_{n,cmp}$ denote the computation time required for client n to perform one local iteration. At round k, the local training time of client n is $HT^k_{n,cmp}$ [46], where H is the number of local iterations between two consecutive global synchronizations. Following the prior works [23], [47], 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 [48]. The communication time of client n at round k can be formulated as

$$T^k_{n,com} = \frac{\theta^k_n R}{C^k_n}, \tag{4}$$

where R represents the size of original (uncompressed) model updates and C^k_n 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^k_n of local training and transmitting parameters at round k is expressed as

$$T^k_n = HT^k_{n,cmp} + T^k_{n,com}. \tag{5}$$

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

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

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

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

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 (7) due to the following reasons: 1) There are few existing works exploring the quantitative relationship among client selection, gradient compression and training performance. It is unclear how the selected clients and compression ratios affect the final convergence of FL. 2) A small compression ratio incurs a small amount of communication overhead but at the expense of slower convergence rate since the direction of the sparse gradients could be different from the direction of the full gradients. Conversely, a large compression ratio captures the accurate gradients, however, it will increase per-round communication overhead. It is challenging to determine different compression ratios for heterogeneous clients to achieve the trade-off between resource overhead and model accuracy. 3) The decisions of client subset and compression ratio are tightly coupled, i.e., their decision influence each other. Neither independent decision nor naive combination can adequately address the challenges of communication efficiency, network dynamics and client heterogeneity. In the following section, we derive a tractable convergence rate to connect client selection and compression ratio with the training performance. On this basis, we develop a joint optimization algorithm to solve the coupled problem.

III. 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: $F_1, F_2, \ldots, F_N$ are all L-smooth, i.e., given $\mathbf{x}$ and $\mathbf{y}$, $F_n(\mathbf{x}) \le F_n(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^T \nabla F_n(\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2$.

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

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

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

These assumptions hold for typical FL models and are common in the convergence analysis literature [6], [30], [38], [49], [50], [51]. 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 [49], we flatten local iterations at each communication round and use $\mathbf{y}^{k,j+1}_n = \mathbf{x}^{k,j}_n - \eta_k \nabla F_n(\mathbf{x}^{k,j}_n; \xi^{k,j}_n)$ to represent the result of a local iteration on client n. If $j+1 < H$, $\mathbf{x}^{k,j+1}_n = \mathbf{y}^{k,j+1}_n$; otherwise, $\mathbf{x}^{k+1,0}_n = \mathbf{x}^{k,0}_n - \frac{\eta_k}{M}\sum_{n \in \mathcal{M}^k} \tilde{\mathbf{G}}^k_n$ and $\mathbf{y}^{k+1,0}_n = \mathbf{y}^{k,H}_n$. Moreover, we define $\bar{\mathbf{x}}^{k,j} = \sum_{n=1}^N p_n \mathbf{x}^{k,j}_n$, $\bar{\mathbf{y}}^{k,j} = \sum_{n=1}^N p_n \mathbf{y}^{k,j}_n$, $\bar{\mathbf{x}}^k = \bar{\mathbf{x}}^{k,0}$ 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 $\mathcal{A}^k_n = \{i \in \mathcal{N} | \pi^k(i) = n\}$ be the set of clients approximated by client $n \in \mathcal{M}^k$ and $\gamma^k_n = |\mathcal{A}^k_n|$. Then we define the *approximation error* at round k as

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

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

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

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

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

The proof of Lemma 1 is included in Appendix A, available online.

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

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

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

Please refer to Appendix B, available online, for the proof of Theorem 1.

Remark 1: Lemma 1 shows the coupled nature of client selection and gradient compression from a theoretical perspective. The optimal state cannot be achieved by independently determining the client subset and compression ratios. We need to jointly consider the decisions of client subset and compression ratios to improve convergence.

Remark 2: 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, which can be used to guide the algorithm design.

IV. 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 IV-A) and then detail two core components of the proposed algorithm, i.e., client selection strategy (Section IV-B) and compression ratio decision strategy (Section IV-C), which are designed by minimizing the approximation error and compression error, respectively.

A. Joint Optimization Process

The key insight behind FedCG is that client selection and compression ratio decision interact with each other. The 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.

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 4.2;
5 Decide compression ratios $\theta_n^{k,i}$ for selected clients by solving optimization problem in Section 4.3;
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;

As shown in Algorithm 2, in each iteration, we first apply submodular maximization in Section IV-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 IV-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 $\theta_n^k, \forall n \in \mathcal{M}^k$ (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 compression ratios of each round are optimized alternately via fixed-point iterations, thus optimizing the overall objective. Our algorithm can be completed in M iterations, which depend on the number of selected clients. How to determine the optimal value of M is not the main focus of this paper and we refer readers to the related works [46], [52] for more information. The algorithm overhead 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. Submodular functions have been widely adopted to measure diversity [53], [54]. Formally, for any subset $\mathcal{S} \subseteq \mathcal{U} \subseteq \mathcal{N}$ and $z \in \mathcal{N} \setminus \mathcal{U}$, a set function V is submodular if $V(\mathcal{S} \cup \{z\}) - V(\mathcal{S}) \geq V(\mathcal{U} \cup \{z\}) - V(\mathcal{U})$, 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 $V(\mathcal{S} \cup \{z\}) - V(\mathcal{S})$. 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 [54].

Inspired by the above facts, our algorithm minimizes the approximation error by applying submodular maximization to select diverse clients [30]. Based on triangular inequality, we can derive an upper bound of the approximation error

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

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

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

Minimizing the upper bound $V(\mathcal{M}^k)$ of the approximation error or maximizing $\bar{V}(\mathcal{M}^k)$ (a constant minus its negation) is essentially equivalent to maximizing a well-known submodular function, i.e., facility location function [55]. Considering the constraint $|\mathcal{M}^k| = M$, the approximation error minimization problem can be transformed into a submodular maximization problem under cardinality constraint, which is NP-hard since it captures well-known NP-hard problems such as Minimum Vertex Cover [53]. Fortunately, the greedy algorithm has been proven to be effective in solving the above submodular maximization problem and provides a $(1 - e^{-1})$-approximation to the optimal solution [56].

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 $|\mathcal{M}^k| = M$

$$\begin{aligned}\mathcal{M}^k &\leftarrow \mathcal{M}^k \cup \{n^*\}, n^* \\ &= \arg\max_{n \in \mathcal{N} \setminus \mathcal{M}^k} \left[\bar{V} \left(\mathcal{M}^k \cup \{n\} \right) - \bar{V} \left(\mathcal{M}^k \right) \right]. \end{aligned} \tag{12}$$

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 [57], and further improved by lazy evaluation [54] and distributed implementations [58]. Consequently, the algorithm overhead can be negligible compared to the massive overhead for model training and transmission [53], 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 [30]. 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 [19]:

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

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. As a consequence, the compression ratios can be continuously adjusted to accommodate system dynamics without requiring future network conditions as prior knowledge. Accordingly, the compression ratio decision problem at round k is expressed as follows:

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

Since the above optimization problem is a linear programming (LP) problem, it can be optimally solved using LP solver (e.g., PuLP [59]). By solving the problem in (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.

V. Quantized FedCG Framework

In this section, we integrate quantization techniques into the heterogeneity-aware FL framework and propose the quantized extension of FedCG, termed Q-FedCG, which reduces the number of bits representing each element to further relieve the communication cost. We first introduce the training process of Q-FedCG and then show how to adaptively determine quantization levels for heterogeneous clients based on gradient innovation.

A. Quantization Techniques

FL requires the selected clients to communicate with the PS iteratively. In practice, the parameter exchange in FL is often carried out over bandwidth-constrained and delay-sensitive wireless edge networks. The repeated transmission induces notable communication delay, which may be larger than the model training time by orders of magnitude [60], especially for high-capacity models with a huge number of parameters. At each round, the per-client communication cost usually depends on the number of transmitted elements and the number of bits representing each element. In FedCG, we have employed Top-k sparsification to transmit only a subset of gradient elements with large magnitudes. Additionally, reducing the number of bits required to represent each element could be another direction to save communication cost.

Modern computers typically use 32 or 64 bits to represent floating-point numbers, which are considered accurate enough for most algorithms. Quantization techniques [6], [10], [11], [12], [13], [14], [15], [16], [17] limit the number of bits that represent the original values such that the data are transmitted in lower precision numerical formats. Since quantization replaces the full precision of floating-point numbers with a smaller bit width, it can significantly reduce the amount of communication data. For instance, transmitting with 2-bit quantization can theoretically save the communication cost by a factor of 16 compared with 32-bit transmission. Therefore, Q-FedCG will adopt quantization schemes to represent sparse model updates with fewer bits, which relieves the communication bottleneck from another perspective.

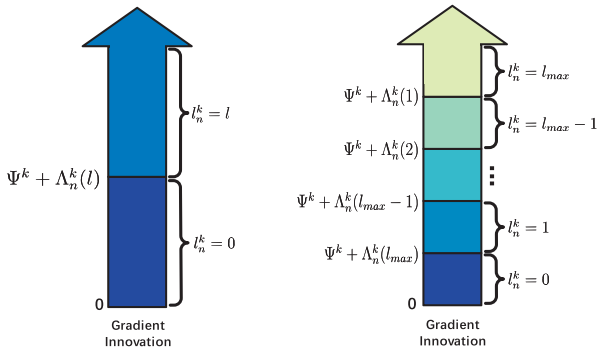
B. Training Process of Q-FedCG

The training process of Q-FedCG is described in Algorithm 3. At round k, the PS first selects a representative client subset $\mathcal{M}^k$ to participate in FL based on statistical heterogeneity. These selected clients update the local models for H iterations. After local training, Q-FedCG not only enables the clients to adaptively compress the model updates $\mathbf{G}_n^k$ using Top-k sparsification but also quantizes the sparse gradients $\tilde{\mathbf{G}}_n^k$ with fewer bits, which is the main difference from the FedCG framework. Most existing quantization schemes either assign identical quantization levels to the clients participating in FL or determine different quantization levels only considering heterogeneous communication resources [15], [16]. However, due to heterogeneous data distributions among clients, local loss functions decrease at different rates and clients contribute differently to global convergence. These existing quantization schemes exhibit inefficiency and poor flexibility for non-IID data. Considering statistical data heterogeneity, we will adjust the quantization level l_n^k for each client n based on gradient innovation, which represents the difference between two consecutive locally computed gradients. If the gradient innovation of the clients is small, it is reasonable to represent their gradients with fewer bits or even discard the redundant gradients, thereby avoiding unnecessary communication [17]. The gradients with significant innovation will be transmitted with more bits (i.e., higher quantization levels) while the gradients with small innovation are quantized by fewer bits (i.e., lower quantization levels). The detailed quantization level strategy will be elaborated in Section V-C.

According to the quantization level l_n^k, the gradient innovation of client n is element-wise quantized by projecting to the closest

Algorithm 3: Training Process of Q-FedCG.

for *Each round* $k = 0, 1, ..., K-1$ **do**
  The PS selects a diverse subset of clients $\mathcal{M}^k$ with $|\mathcal{M}^k| = M$;
  The PS determines differnet compression raio θ_n^k for each client $n \in \mathcal{M}^k$;
  The PS sends the current global model $\mathbf{x}^k$ and compression raio θ_n^k to each client $n \in \mathcal{M}^k$;
  for *Each client* $n \in \mathcal{M}^k$ *in parallel* **do**
    $\mathbf{x}_n^{k,0} = \mathbf{x}^k$;
    for *Each local iteration* $j = 0, 1, ..., H-1$ **do**
      $\mathbf{x}_n^{k,j+1} = \mathbf{x}_n^{k,j} - \eta_k \nabla F_n(\mathbf{x}_n^{k,j}; \xi_n^{k,j})$;
    Compress the model updates $\mathbf{G}_n^k$ to obtain $\tilde{\mathbf{G}}_n^k$ by Top-k sparsification;
    Adjust the quantization level l_n^k based on gradient innovation;
    if $l_n^k \neq 0$ **then**
      Quantize $\tilde{\mathbf{G}}_n^k$ with fewer bits to obtain $Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$;
      Upload the compressed model updates $Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$ to the PS;
  if $l_n^k \neq 0$ **then**
    $\Delta_n^k = Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$;
    $Q_{l_n^{k'}}(\tilde{\mathbf{G}}_n^{k'}) = Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$;
  else
    $\Delta_n^k = Q_{l_n^{k'}}(\tilde{\mathbf{G}}_n^{k'})$;
  The PS updates the global model $\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{\eta_k}{M} \sum_{n \in \mathcal{M}^k} \Delta_n^k$;



(a) Two-Level Quantification (b) Multi-Level Quantification

Fig. 2. Comparison of different quantization level strategies. The two-level criterion fixes the quantization level as l for all uploaded gradients. The multi-level criterion adaptively determines different quantization levels based on gradient innovation.

point in a uniformly discretized grid. The discretized grid contains $2^{l_n^k} - 1$ quantized points and thus each gradient element can be encoded by l_n^k bits in contrast to 32 or 64 bits in most algorithms. A higher quantization level means a finer-grained grid so that the original values can be represented more accurately and vice versa. With the quantization operator $Q(\cdot)$, $Q_{l_n^k}(\tilde{\mathbf{G}}_n^k)$ denotes the compressed model updates after sparsification and quantization and will be uploaded to the PS. Note that $l_n^k = 0$ is the special case, in which the gradient innovation of client n is not significant enough to upload. In the aggregation step, the clients whose quantization levels are zero will reuse previously received gradients $Q_{l_n^{k'}}(\tilde{\mathbf{G}}_n^{k'})$, where k' records the round index when last time the client uploaded the gradients to the PS. Therefore, for client n at round k, the actual gradients Δ_n^k participating in global aggregation are defined as follows:

$$\Delta_n^k = \begin{cases} Q_{l_n^k}(\tilde{\mathbf{G}}_n^k), & l_n^k \neq 0 \\ Q_{l_n^{k'}}(\tilde{\mathbf{G}}_n^{k'}), & l_n^k = 0 \end{cases}.$$

Finally, the PS obtains the latest global model $\mathbf{x}^{k+1}$ by aggregating compressed gradients Δ_n^k. Considering heterogeneous local data and capabilities, Q-FedCG adaptively compresses the model updates using sparsification and quantization or even skips the transmission of gradients with small innovation, which dramatically reduces communication cost from different perspectives and enables efficient FL.

C. Innovation-Based Quantization Level

A key question in Q-FedCG is how many bits should be used to quantize the gradients. The previous works have proposed a two-level quantized method [14], which allows each client to choose from two quantization levels, i.e., 0 and l. The clients can upload nothing or the gradients represented by l bits for each element. As illustrated in Fig. 2(a), the existing solution develops a quantification criterion to decide whether the clients upload the quantized gradients

$$\|Q_l(\tilde{\mathbf{G}}_n^k) - Q_l(\tilde{\mathbf{G}}_n^{k'})\|^2 \geq \Psi^k + \underbrace{3\|\varepsilon_n^k(l)\|^2 + 3\|\varepsilon_n^{k'}(l)\|^2}_{\Lambda_n^k(l)},$$

where $\Psi^k = \frac{1}{\eta_k^2 M^2 I} \sum_{i=1}^{I} \|\mathbf{x}^{k+1-i} - \mathbf{x}^{k-i}\|^2$ indicates the impact of global model updates from previous I rounds, $\varepsilon_n^k(l) = \tilde{\mathbf{G}}_n^k - Q_l(\tilde{\mathbf{G}}_n^k)$ denotes the quantization error of client n at round k under the quantization level of l, and k' is the round index when last time the client uploaded the gradients to the PS.

The above quantification criterion considers both global model updates and quantization error. The client n transmits the gradients with quantization level l only when the difference of the current gradients relative to the previously uploaded gradients exceeds the criterion. Otherwise, the client n will not upload its gradients to the PS, that is, the quantization level is zero. Although they can save the communication cost by skipping less informative quantized gradient transmission, the quantization level is fixed as l for all uploaded gradients. Due to the non-IIDness of local data, the clients' contributions to model convergence are different. Such a fixed and unified quantization strategy is unfair to clients with statistical data heterogeneity. The clients with a large amount of gradient innovation have to compromise against the ones with small innovation, negatively affecting training convergence. Moreover, assigning high quantization resolution to the clients with slight gradient innovation may waste limited communication resources.

To this end, Q-FedCG employs multiple flexible levels instead of two-level quantification criterion. We integrate an innovation-based quantization strategy into our framework, where the quantization levels l_n^k of clients are adaptively calibrated by assessing their contributions to convergence, as suggested in [17]. For the clients with significant gradient innovation, Q-FedCG tends to improve their quantization levels to preserve the precision of the gradients. On the contrary, slowly varying gradients are represented with lower quantization resolution. Specifically, the multi-level quantification criterion is designed as

$$\begin{aligned} &\|Q_{l_{\max}}(\tilde{\mathbf{G}}_n^k) - Q_{l_n^{k'}}(\tilde{\mathbf{G}}_n^{k'})\|^2 \\ &\geq \Psi^k + \underbrace{3\|\varepsilon_n^k(l_{\max} - l_n^k + 1)\|^2 + 3\|\varepsilon_n^{k'}(l_{\max} - l_n^k + 1)\|^2}_{\Lambda_n^k(l_{\max} - l_n^k + 1)}, \end{aligned}$$

where $l_n^k = 1, 2, \ldots, l_{\max}$. Since more bits used for quantization mean less quantization error, we have $\varepsilon_n^k(l_{max}) < \varepsilon_n^k(l_{max} - 1) < \cdots < \varepsilon_n^k(1)$. Therefore, the above criterion divides the range of gradient innovation into $l_{max} + 1$ continuous intervals, as shown in Fig. 2(b). If the gradients of client n have more significant variations, it will be located in the later interval. As a consequence, the client n is assigned a higher quantization level l_n^k and the corresponding gradients can be represented more accurately and vice versa. In this way, Q-FedCG reduces redundant communication and enables the selected clients to transmit adaptively quantized gradients based on different gradient innovation, which is flexible and efficient for data heterogeneity.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed framework on both the physical testbed and simulated environment. We first introduce the experimental settings and then report the experimental results of the prototype system and simulated environment, respectively.

A. Experimental Setup

1) Experimental Platform: The experiments are conducted on 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 edge 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 TX2 devices, 10 NVIDIA Jetson NX devices, and 10 NVIDIA Jetson AGX devices, which reflect capability heterogeneity of clients. Each TX2 device is equipped with an NVIDIA Pascal GPU with 256 CUDA capable cores and a CPU cluster (i.e., a 2-core Denver2 and a 4-core ARM CortexA57). Each NX device carries a 6-core NVIDIA Carmel ARMv8.2 CPU and a 384-core NVIDIA Volta GPU. Each AGX device features an NVIDIA Volta GPU with 512 cores and an 8-core NVIDIA Carmel ARMv8.2 CPU. 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 on an AMAX deep learning workstation, which is equipped with 8 NVIDIA GeForce RTX 3090 GPUs (24 GB GDDR6X memory). To reflect heterogeneous and dynamic network conditions, we fluctuate each client's inbound bandwidth between 10 Mb/s and 20 Mb/s. Considering that the outbound bandwidth is typically smaller than the inbound bandwidth in a typical WAN, we set it to fluctuate between 1 Mb/s and 5Mb/s [50] 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.

2) Datasets and Models: We conduct the experiments over five datasets (i.e., MNIST, EMNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet), which represent a large variety of the small, middle and large training tasks in practical FL scenarios. The MNIST dataset [61] includes a training set with 60,000 samples and a test set with 10,000 samples. The samples in MNIST are 28×28 greyscale images of handwritten digits from 10 classes. The EMNIST dataset [62] totally contains 814,255 samples (731,668 samples for training and 82,587 samples for test) with 62 different classes. The CIFAR-10 dataset [63] is composed of 50,000 color training images and 10,000 testing images labeled in 10 categories. The CIFAR-100 dataset has the same number of images as CIFAR-10 but consists of 100 different classes. The Tiny-ImageNet dataset [64] has 200 classes of images and the dimension of each image is $64 \times 64 \times 3$. We adopt the *convex* logistic regression (LR) model for the MNIST dataset and *non-convex* deep neural networks (e.g., AlexNet, VGG9, VGG16, ResNet9, and ResNet18) for the other four datasets. 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, the clients perform $H = 50$ local iterations at each round.

3) Data Distribution: To simulate various degrees of statistical 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 [8], [38], [50]. *(1) LDA for MNIST and CIFAR-10:* ψ ($\psi = 0.2$, 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 EMNIST, CIFAR-100 and Tiny-ImageNet:* Each client lacks ψ classes of data samples, where $\psi =$ 10, 20, 30, and 40 for EMNIST; $\psi =$ 20, 40, and 60 for CIFAR-100; and $\psi =$ 40, 80, and 120 for Tiny-ImageNet. In particular, we use $\psi = 0$ to denote IID data distribution. Except for the experiments on non-IID data, we shuffle the data and uniformly divide them among all clients.

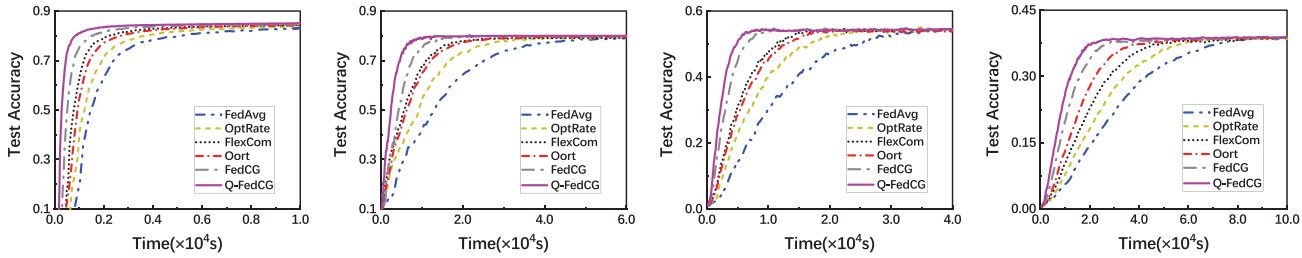
4) Benchmarks and Performance Metrics: We compare the proposed frameworks (FedCG and Q-FedCG) with the following four benchmarks:

- *FedAvg* [3] selects clients uniformly at random and exchanges the entire models between the PS and selected clients.
- *OptRate* [6] adopts quantization techniques to reduce communication overhead and determines identical quantization levels for clients to seek the trade-off between overhead and accuracy.
- *FlexCom* [26] enables flexible compression control and allows clients to compress the gradients to different levels considering the heterogeneity in communication capabilities.
- *Oort* [37] improves time-to-accuracy performance via guided participant selection, which tends to pick clients with high statistical and system utility.

We adopt three widely used metrics for performance evaluation. *(1) Test accuracy* is measured by the proportion between the amount of the correct data through model inference and that of all test data. *(2) Completion time* is defined as the time consumption to reach the target accuracy. We adopt this metric to evaluate the training speed. *(3) Network traffic* denotes the total size of data uploaded from clients to the PS, which is used to quantify the communication overhead of different methods. The test accuracy helps to validate whether FL methods can effectively guarantee model convergence or not. Meanwhile, the completion time and network traffic indicate if the proposed methods are resource-efficient.

B. Testbed Results

1) Training Performance: We first compare the training performance of different methods on the prototype system. The comparison of model accuracy of the proposed framework with other methods is depicted in Fig. 3. We observe that FedCG and Q-FedCG achieve a comparable accuracy and converge to the stationary point more quickly. For example, Q-FedCG accelerates the training process over EMNIST dataset by 5.9×, 4.3×, 2.5× and 3.0× compared with FedAvg, OptRate, FlexCom and Oort. Besides the EMNIST dataset, our framework



(a) VGG9 over EMNIST (b) VGG16 over CIFAR-10 (c) ResNet9 over CIFAR-100 (d) ResNet18 over Tiny-ImageNet

Fig. 3. Training performance of different methods on the prototype system.

TABLE II
COMPLETION TIME AND SPEEDUP OF DIFFERENT METHODS TO ACHIEVE THE TARGET ACCURACY

Datasets	Metrics	FedAvg	OptRate	FlexCom	Oort	FedCG	Q-FedCG
EMNIST (Acc=82%)	Time	7513.4s	5434.8s	3207.2s	3764.8s	2203.9s	1267.2s
	Speedup	5.9×	4.3×	2.5×	3.0×	1.7×	1.0×
CIFAR-10 (Acc=78%)	Time	44765.1s	31924.2s	20679.9s	22350.8s	12779.0s	7879.9s
	Speedup	5.7×	4.1×	2.6×	2.8×	1.6×	1.0×
CIFAR-100 (Acc=54%)	Time	35047.9s	24520.5s	17726.2s	19937.5s	10068.7s	7313.9s
	Speedup	4.8×	3.4×	2.4×	2.7×	1.4×	1.0×
Tiny-ImageNet (Acc=37%)	Time	68931.7s	54918.1s	45543.9s	36401.1s	25614.4s	19327.3s
	Speedup	3.6×	2.8×	2.4×	1.9×	1.3×	1.0×

also shows great advantages for more complex and challenging datasets. Compared to the benchmarks, we can provide up to 5.7× speedup for VGG16 over CIFAR-10, 4.8× speedup for ResNet9 over CIFAR-100, and 3.6× speedup for ResNet18 over Tiny-ImageNet. Meanwhile, the proposed framework can improve the test accuracy of the global model under the same completion time. For example, FedCG attains the accuracy of 79.76% after training 30,000 s over CIFAR-10 while FedAvg, OptRate, FlexCom and Oort achieve 74.05%, 77.90%, 79.18% and 78.94%, respectively.

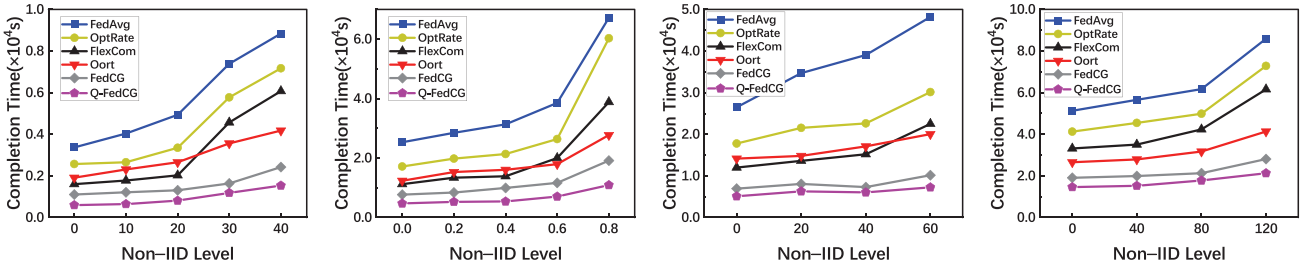
The explanation for the above phenomenon is that our framework jointly optimizes the client selection and gradient compression strategy, which sharply accelerates the training process of FL without sacrificing accuracy. The diversity of client subset prevents optimizations from favoring specific clients. Different compression ratios allow clients to upload compressed gradients of different sizes, which not only reduces the communication burden but also matches the heterogeneous and dynamic capabilities of clients. By contrast, the benchmarks cannot simultaneously handle the challenges of communication efficiency, network dynamics and client heterogeneity, making the overall training process inefficient.

2) Completion Time: To further validate the efficiency of FedCG and Q-FedCG, we record the completion time of different methods to attain the target accuracy and the speedup achieved by our methods in Table II. Note that the target accuracy is set as the accuracy that all methods can reach. As summarized in Table II, it is clear that the proposed framework can drastically reduce time consumption and complete the training tasks fastest compared to benchmarks. For instance, FedCG saves the training time by 71.3%, 58.9%, 43.2% and 49.5% for ResNet9 over CIFAR-100 compared with FedAvg, OptRate, FlexCom and Oort.

The reasons for such superior performance are as follows. FedAvg and Oort transmit over-parameterized models and gradients, thereby slowing down the training process of FL. By contrast, the solutions with compression can save much more completion time by reducing the payload for transmitting. However, OptRate assigns identical compression ratios to heterogeneous clients, which exacerbates the straggler effect. Although FlexCom determines different compression ratios for clients, it still produces a long training completion time without considering the computation heterogeneity. With the assistance of adaptive client selection and gradient compression, FedCG brings significant savings in completion time, thus achieving efficient FL.

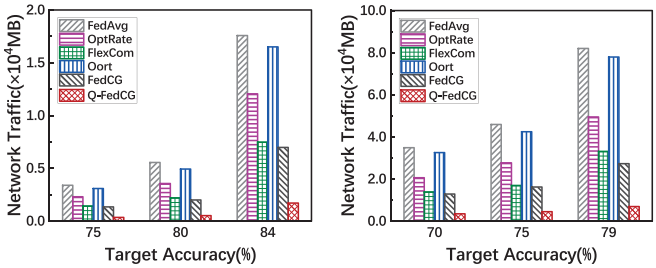
3) Effect of Non-IID Data: We proceed to investigate how our proposed framework performs under statistical heterogeneity. Fig. 4 plots the required time to reach the target accuracy under different levels of non-IID data. We set the target accuracy of VGG9, VGG16, ResNet9 and ResNet18 as 76%, 70%, 51% and 33%, respectively. We make the following three observations from the experimental results. First, all methods suffer from performance degradation with the increasing skewness of data distribution. Nevertheless, FedCG and Q-FedCG always achieve superior performance over benchmarks under different non-IID levels. The proposed framework only has the slightest increase in completion time and exhibits robustness to non-IID data. Another observation is that the performance gain of our methods over the existing studies is enhanced as non-IID level increases. For example, the maximum speedup brought by Q-FedCG increases from 5.4× to 6.2× with non-IID level of CIFAR-10 dataset varying from 0 to 0.8.

Finally, our framework outperforms the benchmarks even under high data skewness. Specifically, Q-FedCG still expedites the training speed by 75.2%, 70.8%, 65.5%, 48.5% and 24.3% than FedAvg, OptRate, FlexCom, Oort and FedCG under the highest non-IID level of Tiny-ImageNet dataset. The advantage is attributed to diverse client selection which increases the impact of under-represented clients. Such a client selection strategy can reduce the bias introduced by non-IID data. On the basis of FedCG, Q-FedCG further calibrates the quantization levels for the selected clients. We assess their contributions to model convergence and allocate higher quantization resolution to



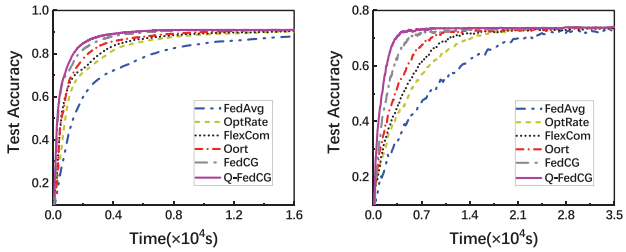
(a) VGG9 over EMNIST (b) VGG16 over CIFAR-10 (c) ResNet9 over CIFAR-100 (d) ResNet18 over Tiny-ImageNet

Fig. 4. Completion time under different levels of non-IID data.



(a) VGG9 over EMNIST (b) VGG16 over CIFAR-10

Fig. 5. Communication overhead of different methods to achieve the target accuracy.



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

Fig. 6. Training performance in dynamic and heterogeneous simulation environments.

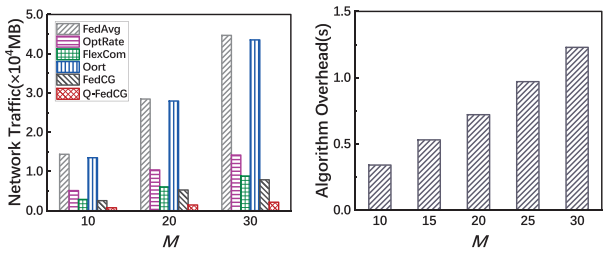
gradients with larger variations, thus effectively reducing redundant communication and accommodating non-IID data. These results imply that our design effectively alleviates the issue of statistical heterogeneity.

4) Communication Overhead: In Fig. 5, we illustrate the required network traffic regarding the specified test accuracy. We observe that our methods take the least amount of communication overhead to achieve the target accuracy. Concretely, compared with four benchmarks, FedCG can mitigate the traffic consumption by 42.2% for VGG9 over EMNIST and 44.6% for VGG16 over CIFAR-10 on average. Similar phenomenon is witnessed in the Q-FedCG framework. Even compared to the FedCG framework, Q-FedCG still saves the communication overhead of 72.1-75.5%. The reason behind these advantages lies in that FedCG and Q-FedCG not only select a representative client subset but also involve adaptive gradient compression strategy. Besides, Q-FedCG further quantizes the model updates for the selected clients on the basis of Top-k sparsification. The sparse local gradients are represented by fewer number of bits. Hence, our quantized extension can achieve higher compression rates than FedCG and reduce the network traffic more aggressively. The above experimental results show the impressive resource efficiency of the proposed framework.

C. Simulation Results

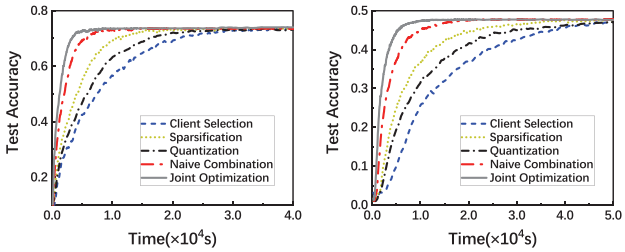
1) Dynamic and Heterogeneous Environments: To evaluate the performance of the proposed framework in large-scale FL scenarios, we conduct our experiments by simulations with 100 clients. Fig. 6 plots the accuracy curve of different methods in dynamic and heterogeneous environments. We find that our proposed framework still substantially outperforms the other benchmarks in large-scale FL scenarios and exhibits faster convergence without sacrificing accuracy. For instance, FedAvg, OptRate, FlexCom and Oort take 25,495 s, 17,391 s, 15,137 s and 10,482 s to reach 72% accuracy for AlexNet over CIFAR-10, while the completion time of FedCG and Q-FedCG are 7,311 s and 3,970 s, respectively. Our methods can provide up to 6.4× speedup. Such performance gain 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 after sparsification and quantization, thereby relieving the straggler problem and expediting the convergence of FL. 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.

2) Effect of Different M Values: We conduct the simulation experiments to analyze the influence of the number of selected clients (i.e., M) on training efficiency. First, 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. 7(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. The proposed methods still outperform the four benchmarks under different values of M and reduce network traffic consumption by up to 95.2%. Second, we measure the



(a) Network Traffic (b) Algorithm Overhead

Fig. 7. Effect of different M values on network traffic and algorithm overhead.



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

Fig. 8. Training performance of independent decision, naive combination, and joint optimization.

decision overhead of joint optimization algorithm with different values of M, which is illustrated in Fig. 7(b). Although the algorithm overhead becomes larger as M increases, the resulting overhead 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 our framework in FL.

3) Necessity of Joint Optimization: Instead of simple combination, our proposed framework aims to achieve efficient FL by joint optimization of client selection and gradient compression. To indicate the importance of joint optimization algorithm, we compare the training performance of independent decision, naive combination and joint optimization. As shown in Fig. 8, it is clear that our method consistently converges faster than independent decision and naive combination without loss of accuracy. The joint optimization method can save 44.5–84.8% of the training time to reach target accuracy (e.g., 72%) for AlexNet over CIFAR-10 and 49.8–84.5% of the training time to reach target accuracy (e.g., 46%) for ResNet9 over CIFAR-100. The explanation for this phenomenon is that neither independent decision nor naive combination can handle network dynamics and client heterogeneity well due to the coupled property, thus negatively affecting the convergence performance. The above experimental results demonstrate the impressive performance improvement of joint optimization process and thus emphasize its necessity.

VII. 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 [5]. 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.

A. Compression Methods

Compression techniques have been adopted to alleviate the transmission burden, including quantization [6], [10], [11], [12], [13], [14], [15], [16], [17] and sparsification [5], [18], [19], [20], [21], [22], [23], [24], [25]. The quantization based methods [6], [10], [11], [12], [13], [14], [15], [16], [17] aim to represent each element with fewer bits. QSGD [11] adjusts the number of bits sent per training round under the premise of ensuring convergence. SignSGD [13] only transmits the sign of each stochastic gradient and performs aggregation using majority vote, thereby enabling 1-bit compression. HeteroSAg [15] allows the clients to determine their quantization resolution according to available communication resources. DAdaQuant [16] as a computationally efficient and robust algorithm combines time- and client-adaptive quantization. The authors in [65] and [66] have focused on vector quantization. Optimal compression ratio allocation for quantization is considered in [6] to seek the balance between model accuracy and communication overhead.

Other studies [5], [18], [19], [20], [21], [22], [23], [24], [25] apply sparsification to transmit a small subset of gradients so that the communication overhead can be reduced dramatically. For example, the authors in [21] and [22] enable dynamic compression control and adapt the communicated volume to time-varying network conditions. DGC [23] only sends gradient elements that exceed a certain threshold, achieving a very high compression ratio (e.g., 0.1%). Adaptive degree of sparsity considering characteristics of FL tasks is studied in [25]. DeepReduce [5] decomposes sparse tensors into values and indices and allows both independent and combined compression. However, most aforementioned works assign identical or fixed compression ratios to heterogeneous clients and thus the stragglers with poor channel conditions will become the bottleneck of model training. The authors in [26] provide client-specific compression schemes according to communication heterogeneity. However, they do not consider statistical heterogeneity and exhibit poor performance in the presence of non-IID data, in terms of model accuracy and convergence rate.

B. Client Selection Methods

Considering limited communication bandwidth and device availability, FL usually selects only a fraction of clients to participate in training. 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 [3], [49], which results in poor training performance and long latency due to non-IID data and capability heterogeneity [38]. Considering the statistical property, some sampling methods have investigated different criteria to evaluate the importance of clients, such as local loss [27], test accuracy [28], model updates [8], [29], [30], client correlations [31], and local data variability [32]. The

clients with "important" data will have higher probabilities to be selected at each round. However, these strategies ignore the heterogeneity of clients' capabilities and may suffer from the straggler effect.

Some works [7], [33], [34], [35], [36] have designed client selection schemes that tackle heterogeneous system resources for fast convergence, but non-IID data still hurt the model accuracy. Oort [37] tends to sample clients with high system and statistical utility while allowing developers to specify their data selection criteria. The authors in [38] optimize 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.

VIII. 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 considering their coupled nature. Besides, we introduce quantization techniques into FedCG and design the Q-FedCG framework, which calibrates the quantization levels of the selected clients based on gradient innovation. Experimental results demonstrate the effectiveness of FedCG and Q-FedCG. Meanwhile, we emphasize some limitations of our work. In extreme scenarios where the network conditions of clients vary significantly within a round, our framework may not provide sufficient performance gains. Improving one-shot round optimization to better accommodate network dynamics could be an interesting direction for future work.

References

[1] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[4] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1/2, pp. 1–210, 2021.

[5] H. Xu, K. Kostopoulou, A. Dutta, X. Li, A. Ntoulas, and P. Kalnis, "DeepReduce: A sparse-tensor communication framework for federated deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 21150–21163.

[6] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1459–1468.

[7] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[8] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1698–1707.

[9] J. Liu, Y. Xu, H. Xu, Y. Liao, Z. Wang, and H. Huang, "Enhancing federated learning with intelligent model migration in heterogeneous edge computing," in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 1586–1597.

[10] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7663–7673.

[11] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.

[12] W. Wen et al., "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1508–1518.

[13] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 560–569.

[14] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2031–2044, Apr. 2022.

[15] A. R. Elkordy and A. S. Avestimehr, "HeteroSAg: Secure aggregation with heterogeneous quantization in federated learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2372–2386, Apr. 2022.

[16] R. Hönig, Y. Zhao, and R. Mullins, "DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 8852–8866.

[17] Y. Mao et al., "Communication-efficient federated learning with adaptive quantization," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–26, 2022.

[18] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 440–445.

[19] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4452–4463.

[20] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[21] X. Zhang, X. Zhu, J. Wang, H. Yan, H. Chen, and W. Bao, "Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks," *Inf. Sci.*, vol. 540, pp. 242–262, 2020.

[22] A. M. Abdelmoniem and M. Canini, "DC2: Delay-aware compression control for distributed machine learning," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[23] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Int. Conf. Learn. Representations*, 2018.

[24] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1306–1316.

[25] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 300–310.

[26] 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 *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

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

[28] I. Mohammed et al., "Budgeted online selection of candidate IoT clients to participate in federated learning," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5938–5952, Apr. 2021.

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

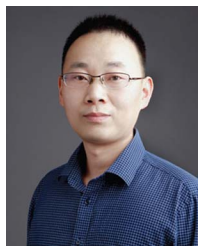
[30] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *Proc. Int. Conf. Learn. Representations*, 2022.

[31] M. Tang et al., "FedCor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10102–10111.

[32] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 3095–3099.
[33] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1449–1458.
[34] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.
[35] Z. Chai et al., "TiFL: A tier-based federated learning system," in *Proc. 29th Int. Symp. High- Perform. Parallel Distrib. Comput.*, 2020, pp. 125–136.
[36] 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 *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 606–616.
[37] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *Proc. 15th USENIX Symp. Operating Syst. Des. Implementation*, 2021, pp. 19–35.
[38] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1739–1748.
[39] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.
[40] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.
[41] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.
[42] H. Xu, M. Chen, Z. Meng, Y. Xu, L. Wang, and C. Qiao, "Decentralized machine learning through experience-driven method in edge networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 515–531, Feb. 2022.
[43] Z. Tang, S. Shi, and X. Chu, "Communication-efficient decentralized learning with sparsification and adaptive peer selection," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 1207–1208.
[44] S. Shi, K. Zhao, Q. Wang, Z. Tang, and X. Chu, "A convergence analysis of distributed SGD with communication-efficient gradient sparsification," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3411–3417.
[45] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2530–2541.
[46] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
[47] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.
[48] Y. Zhan and J. Zhang, "An incentive mechanism design for efficient edge learning by deep reinforcement learning approach," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2489–2498.
[49] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2020.
[50] L. Wang, Y. Xu, H. Xu, M. Chen, and L. Huang, "Accelerating decentralized federated learning in heterogeneous edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5001–5016, Sep. 2023.
[51] Z. Ma, Y. Xu, H. Xu, Z. Meng, L. Huang, and Y. Xue, "Adaptive batch size for federated learning in resource-constrained edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 37–53, Jan. 2023.
[52] J. Liu et al., "Adaptive asynchronous federated learning in resource-constrained edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 674–690, Feb. 2023.
[53] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 6950–6960.
[54] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Proc. Optim. Techn.*, Springer, 1978, pp. 234–243.
[55] G. Cornuejols, M. Fisher, and G. L. Nemhauser, "On the uncapacitated location problem," *Ann. Discrete Math.*, vol. 1, pp. 163–177, 1977.
[56] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
[57] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1812–1818.
[58] B. Mirzasoleiman, M. Zadimoghaddam, and A. Karbasi, "Fast distributed submodular cover: Public-private data summarization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3601–3609.
[59] S. Mitchell, M. OSullivan, and I. Dunning, "Pulp: A linear programming toolkit for Python," *Univ. Auckland, Auckland, New Zealand*, vol. 65, 2011.
[60] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," in *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 17, 2021, Art. no. e2024789118.
[61] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[62] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 2921–2926.
[63] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.
[64] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 3, 2015.
[65] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, 2020.
[66] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2020.

Yang Xu (Member, IEEE) received the BS degree from the Wuhan University of Technology, in 2014, and the PhD degree in computer science and technology from the University of Science and Technology of China, in 2019. He is currently an associate researcher with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include ubiquitous computing, deep learning, and mobile edge computing.

Zhida Jiang received the BS degree from the Hefei University of Technology, in 2019. He is currently working toward the PhD degree with the School of Computer Science and Technology, University of Science and Technology of China (USTC). His research interests include mobile edge computing, federated learning, and distributed machine learning.



Hongli Xu (Member, IEEE) received the BS degree in computer science from the University of Science and Technology of China, China, in 2002, and the PhD degree in computer software and theory from the University of Science and Technology of China, China, in 2007. He is a professor with the School of Computer Science and Technology, University of Science and Technology of China (USTC), China. He was awarded the Outstanding Youth Science Foundation of NSFC, in 2018. He has won the best paper award or the best paper candidate in several famous conferences. He has published more than 100 papers in famous journals and conferences, including *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*, Infocom and ICNP, etc. He has also held more than 30 patents. His main research interest is software defined networks, edge computing, and Internet of Thing.

Zhiyuan Wang received the BS degree from Jilin University, in 2019. He is currently working toward the PhD degree with the School of Computer Science and Technology, University of Science and Technology of China (USTC). His main research interests are edge computing, federated learning, and distributed machine learning.

Chen Qian (Senior Member, IEEE) received the BS degree in computer science from Nanjing University, in 2006, the MPhil degree in computer science from The Hong Kong University of Science and Technology, in 2008, and the PhD degree in computer science from The University of Texas at Austin, in 2013. He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of California at Santa Cruz. He has published more than 60 research papers in highly competitive conferences and journals. His research interests include computer networking, network security, and Internet of Thing.

Chunming Qiao (Fellow, IEEE) is a SUNY distinguished professor and also the current chair of the Computer Science and Engineering Department, University at Buffalo, Buffalo, New York. He was elected to IEEE fellow for his contributions to optical and wireless network architectures and protocols. His current focus is on connected and autonomous vehicles. He has published extensively with an h-index of more than 69. Two of his papers have received the best paper awards from IEEE and Joint ACM/IEEE venues. He also has seven US patents and served as a consultant for several IT and Telecommunications companies since 2000. His research has been funded by a dozen major IT and telecommunications companies including Cisco and Google, and more than a dozen NSF grants.