Accelerating Hybrid Federated Learning Convergence Under Partial Participation

Jieming Bian, Lei Wang, Kun Yang, Cong Shen, *Senior Member, IEEE*, and Jie Xu, *Senior Member, IEEE*

***Abstract*—Over the past few years, Federated Learning (FL) has become a popular distributed machine learning paradigm. FL involves a group of clients with decentralized data who collaborate to learn a common model under the coordination of a centralized server, with the goal of protecting clients' privacy by ensuring that local datasets never leave the clients and that the server only performs model aggregation. However, in realistic scenarios, the server may be able to collect a small amount of data that approximately mimics the population distribution and has stronger computational ability to perform the learning process, resulting in the development of a hybrid FL framework. While previous hybrid FL work has shown that the alternative training of clients and server can increase convergence speed, it has focused on the scenario where clients fully participate and ignores the negative effect of partial participation. In this paper, we provide theoretical analysis of hybrid FL under clients' partial participation to validate that partial participation is the key constraint on the convergence speed. We then propose a new algorithm called FedCLG, which investigates the two-fold role of the server in hybrid FL. Firstly, the server needs to process the training steps using its small amount of local datasets. Secondly, the server's calculated gradient needs to guide the participating clients' training and the server's aggregation. We validate our theoretical findings through numerical experiments, which show that FedCLG outperforms state-of-the-art methods.**

***Index Terms*—Federated learning, convergence analysis, server-clients collaboration.**

I. Introduction

RECENT years have seen exponential growth in data collection due to technological advancements, leading to the development of stronger machine learning models [1]. However, traditional centralized machine learning algorithms struggle with handling the distributed nature of this type of data, which is often spread across multiple clients, such as mobile devices [2]. To overcome this problem, Federated Learning (FL) [3] has emerged as an important paradigm in modern machine learning. FL is a distributed machine learning approach where clients with decentralized data collaborate to learn a common model under the coordination of a parameter server. It has several advantages, including enhanced user data privacy [4], [5], scalability to new clients and datasets [6], and faster model convergence rate [7], [8], [9]. Despite these benefits, current FL systems typically assign the server to only simple computations, such as aggregating local models, wasting its powerful computational resources. Moreover, traditional FL assumes that datasets are exclusively available to the clients, either independently and identically distributed (IID) or non-IID. However, this is not always the case in real-world scenarios. In many cases, the entity building the machine learning model operates the server and possesses a small amount of data that approximately mimics the overall population distribution. Although a machine learning model can be trained based solely on the server data, the model performance will be limited by the size of the server dataset. Thus, a hybrid FL approach, which collaboratively utilizes the massive client data and a small amount of server data in a decentralized and privacy-preserving manner is of paramount practical importance to boost model performance.

Compared to traditional FL, which assumes that only clients can access data while the server can only perform model aggregation, the literature on hybrid FL is relatively scarce. Authors in [10] make the assumption that the data collected by the server is complementary to the data held by each client. However, this assumption may only be applicable to specific scenarios, and in most real-world cases, the entity operating the server is likely to have access to a small amount of data that can approximate the population data distribution. To address this issue, this paper adopts a similar setting to [11], which proposes a hybrid model training design called CLG-SGD (short for cascading local-global SGD). In this design, the server performs aggregate-then-advance training. The empirical findings presented in [11] demonstrate that compared to client-only local data training (e.g., Local-SGD), CLG-SGD enhances the convergence speed. However, its theoretical analysis bounds server-side and client-side updates separately, failing to comprehensively represent the theoretical benefits of additional server training in non-convex settings. Moreover, [11] mainly focuses on the IID and fully-participated scenario, which may not be realistic in practical applications. In reality, clients may choose to participate only when they have access to a reliable Wi-Fi connection and a

Manuscript received 11 April 2023; revised 25 November 2023; accepted 16 May 2024. Date of publication 3 June 2024; date of current version 23 July 2024. This work was supported by the NSF under Grant 2033681, Grant 2006630, Grant 2044991, and Grant 2319780. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mingyi Hong. *(Corresponding author: Jie Xu.)*

Jieming Bian, Lei Wang, and Jie Xu are with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146 USA (e-mail: jxb1974@miami.edu; lxw725@miami.edu; jiexu@miami.edu).

Kun Yang and Cong Shen are with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904 USA (e-mail: ky9tc@virginia.edu; cong@virginia.edu).

Digital Object Identifier 10.1109/TSP.2024.3408631

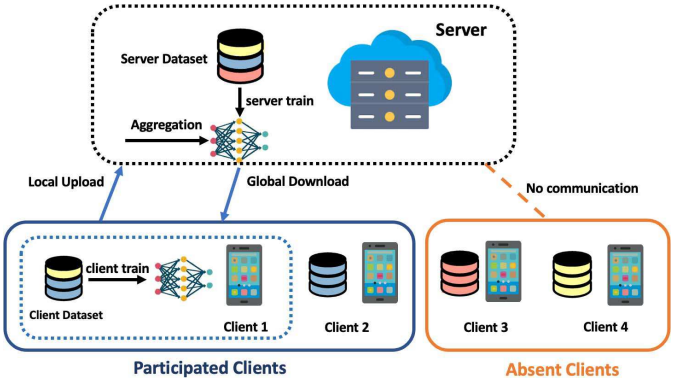


Fig. 1. Illustration of a Communication Round in Hybrid FL with Non-IID Client Data and Partial Participation.

power source [3]. Therefore, only a small percentage of clients may participate in each round. Additionally, data is distributed across multiple clients (e.g., mobile devices), each with its own unique data distribution [12]. Thus, the investigation of hybrid FL under non-IID and partial participation scenarios is crucial.

In this paper, we first revisit CLG-SGD [11] under non-IID and partial participation setting, and introduce a novel convergence analysis. This analysis improves the convergence rate of CLG-SGD, demonstrating how additional server-side training can expedite convergence. However, our findings also indicate that partial participation errors can still impede CLG-SGD's convergence rate, even with augmented server training. To mitigate this issue, we then introduce FedCLG (Federated cascading local-global learning), a new algorithm for hybrid FL that leverages server training to improve model convergence speed and correct partial participation errors in non-IID and partially participated scenarios (See Fig. 1). Specifically, FedCLG has two main responsibilities for server training. First, the server training starts with the latest aggregated global model and advances it using its limited local dataset. This allows the server to contribute to the global model with its advanced computation capabilities. Second, the server conducts an additional one-round training before broadcasting the new global model to each participating client. The gradient computed during this additional server training is utilized to correct partial participation errors. We propose two versions of FedCLG based on where partial participation errors are corrected. FedCLG-S corrects partial participation errors during server model aggregation, while FedCLG-C corrects them at each client's side during local training. Our proposed algorithm aims to maximize the benefits of server training to improve model convergence speed and correct partial participation errors in non-IID and partially participated scenarios. We summarize our main contributions below: 1. We provide a novel theoretical convergence analysis of the state-of-the-art hybrid FL method, CLG-SGD, that validates the benefit of additional server training without requiring the assumption of IID data or full client participation. Our analysis highlights that, despite the additional server training, convergence speed is still limited by partial participation errors. 2. We propose FedCLG, a new algorithm that maximizes the potential benefits of server training in hybrid FL. We introduce two versions of FedCLG, FedCLG-S and FedCLG-C, to account for different communication and computation scenarios. We provide theoretical convergence analysis for both versions. 3. We conduct extensive experiments on three datasets, demonstrating the superior performance of FedCLG over existing state-of-the-art methods.

The remainder of this paper is organized as follows. Related works are surveyed in Section II. The system model and extensive theoretical analysis of the state-of-the-art hybrid FL method are presented in Section III. FedCLG is detailed in Section IV and its two versions FedCLG-S and FedCLG-C are analyzed in Section V. Experiment results are reported in Section VII. Finally, Section VIII concludes the paper.

II. Related Works

A. Federated Learning

With the growing demand for local data storage and on-device model training, Federated Learning [13], [14], [15] has attracted significant interest in recent years. FedAvg first proposed by [3], operates by periodically averaging local Stochastic Gradient Descent (SGD) updates. This work has inspired numerous follow-up studies focusing on FL with IID client datasets and full client participation [16], [17], [18], [19]. Under the assumptions of complete participation and IID client datasets, several theoretical works [16], [20] have emerged, providing a linear speedup convergence guarantee, on par with the rate of parallel SGD [21]. However, real-world scenarios often present challenges in FL due to non-IID data and partial client participation [22]. Recent works [8], [23], [24], [25], [26] have addressed these issues by offering similar convergence rates under non-IID and partial participation settings.

B. Hybrid Federated Learning

The majority of existing FL research focuses on the scenario where data is exclusively stored on the client side, and the server is only responsible for the aggregation step, ensuring clients' privacy requirements are met. However, this approach could potentially underutilize the server's computational capabilities. Compared to the clients, which are typically mobile devices in FL settings, the server generally possesses significantly greater computational power [27]. This has led to the emergence of a new FL configuration, referred to as hybrid FL. Current hybrid FL research can be divided into two categories, based on the source of the server dataset.

The first category of hybrid FL assumes that the server cannot collect data independently, while clients with limited computational resources can upload less privacy-sensitive data samples to the server to aid training [28], [29]. These studies concentrate on optimizing the trade-off between data sample communication costs and the benefits of model training.

The second line of hybrid FL, more closely related to this paper, assumes that the server can collect a small portion of the total data samples [10], [11]. While [10] posits that the server's data complements each client's data, a more realistic assumption is that the server is more likely to gather a small amount

of data that approximates the population data distribution [30]. Our work adopts a similar setting to [11]. However, while [11] focuses on IID data and full client participation, our research investigates the more realistic scenario of non-IID client data and partial client participation.

C. Variance Reduction

Variance reduction has been a widely studied concept across various fields. Monte Carlo sampling methods employ the control variates technique to reduce variance [31], while in stochastic gradient estimates for large-scale machine learning, SVRG [32] and SAG [33] have been introduced to reduce the stochastic sampling-variance. SAG was later simplified, leading to the proposal of SAGA [34]. In Federated Learning (FL), the variance caused by randomly participating clients has a stronger impact than the variance caused by stochastically selected data samples. As a result, variance reduction methods in FL focus more on reducing client-variance. SCAFFOLD [23], an extension of SAGA, was proposed as the first variance reduction method in FL, which inspired subsequent works such as [35], [36], [37], [38] that attempt to reduce client-variance to increase convergence speed. However, none of these methods consider the hybrid FL setting and can result in the misutilization of stale information. In this work, we propose the first approach to reducing client-variance in hybrid FL, which fully exploits the benefits of server-side small datasets. A detailed comparison between our method and existing variance-reduction FL approaches is presented in Section IV.

III. Problem Formulation and CLG-SGD

A. Problem Formulation

In the hybrid federated learning setting, we aim to optimize the model parameters $x \in \mathcal{R}^d$ by minimizing the global objective function $f(x)$, similar to traditional federated learning frameworks. The global objective function is defined as:

$$\min_x f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{1}$$

where $f_i(x) = \frac{1}{m_i}\sum_{z\in\mathcal{D}_i} l(x,z)$ represents the local objective of client i computed on their local dataset $\mathcal{D}_i$ with m_i data points. The loss function is denoted by $l(.,.)$, and z represents a data sample from the local dataset $\mathcal{D}_i$. The total data samples in the FL system are represented as m, such that $\sum_i^N m_i = m$, and the total number of clients is denoted as N. The underlying data distribution of the total m data samples is denoted as $\mathcal{V}$. We assume, without loss of generality, that all N clients' local objectives have equal weight in the global objective function (1). The algorithms and theoretical analysis can be easily extended to cases where client objectives are unequally weighted, such as proportional to the local data size.

In contrast to traditional FL, the hybrid FL framework posits that, in addition to the data available at each client, the server can collect a small dataset $\mathcal{D}_s^t$ with a constant size of m_s, which is data homogeneous with the overall dataset. Although the underlying data distribution of $\mathcal{D}_s^t$ remains consistent and approximates the overall population distribution $\mathcal{V}$, the dataset itself changes with each global round t (while remaining fixed within the global round). Consequently, the server's optimization problem becomes:

$$\min_x f_s(x) = \frac{1}{m_s}\sum_{z\in\mathcal{D}_s^t} l(x,z). \tag{2}$$

However, because the size of $\mathcal{D}_s^t$ is considerably smaller than the overall dataset stored at each client (i.e., $m_s \ll m$), relying solely on $\mathcal{D}_s^t$ for model training could result in suboptimal outcomes. Additionally, the server's limited access to the fixed dataset for local training during specific time periods in each global round may significantly increase the training time.

Determining the best approach to utilize both server and clients' data for training and achieve optimal convergence performance is a challenging problem. To address this, we revisit the CLG-SGD algorithm in the hybrid FL setting. At each round t, the server randomly selects a subset of M clients, denoted as $\mathcal{S}_t$, and sends the global model x_t to these clients. Upon receiving x_t, each selected client i performs K rounds of local updates as follows:

$$\begin{aligned} x_{t,0}^i &= x_t; \\ x_{t,k+1}^i &= x_{t,k}^i - \eta g_{t,k}^i, \quad k = 0,\dots,K-1, \end{aligned} \tag{3}$$

where η is the client-side local learning rate, and $g_{t,k}^i = \nabla f_i(x_{t,k}^i, \zeta_i)$ is the stochastic gradient evaluated on a randomly drawn mini-batch ζ_i at client i ($g_{t,k}^i = \nabla f_i(x_{t,k}^i)$ if a full gradient is used). After K steps of local training, client i sends back its update $\Delta_t^i = x_{t,K}^i - x_t$ to the server, which aggregates the updates to update the global model as follows:

$$x_{t+1}^s = x_t + \eta_g \frac{1}{M}\sum_{i\in\mathcal{S}_t} \Delta_t^i, \tag{4}$$

where η_g is the global (aggregation) learning rate, and x_{t+1}^s represents an intermediate stage between client local training and server local training. In classic FL, the iteration ends at this point. However, in hybrid FL, the server not only aggregates the clients' updates but also utilizes its own dataset $\mathcal{D}_s^t$ for server training. Thus, after aggregating the model x_{t+1}^s, the server also performs E rounds of local updates as follows:

$$\begin{aligned} x_{t+1,0}^s &= x_{t+1}^s; \\ x_{t+1,e+1}^s &= x_{t+1,e}^s - \gamma g_{t+1,e}^s, \quad e = 0,\dots,E-1; \\ x_{t+1} &= x_{t+1,E}^s, \end{aligned} \tag{5}$$

where γ is the server learning rate, and $g_{t,e}^s = \nabla f_s(x_{t,e}^s, \zeta_s)$ is the stochastic gradient evaluated on a randomly drawn mini-batch ζ_s from the server dataset $\mathcal{D}_s^t$ ($g_{t,e}^s = \nabla f_s(x_{t,e}^s)$ if a full gradient is used). After the server-side training, the global model advances from x_{t+1}^s to x_{t+1}. Then the server broadcasts the new global model x_{t+1} for the next round of iteration.

In the previous work [11], the authors focus on IID and fully participating settings and fail to show how additional server training accelerates the convergence speed in non-convex settings. In this paper, we extend the analysis to non-IID and partial participation settings in the following subsection. Our novel theoretical analysis demonstrates that, although additional server

training can improve the convergence rate, convergence speed is still dominated by partial participation error resulting from data heterogeneity and randomly selected clients. Based on this observation, we propose a new algorithm, FedCLG, in Section IV.

B. Novel Convergence Analysis of CLG-SGD [11]

For the theoretical analysis in this paper, we make the following assumptions: in each round, the server selects a subset of clients uniformly without replacement. In addition, our convergence analysis will utilize the following standard technical assumptions.

Assumption 1 (Lipschitz Smoothness): There exists a constant $L > 0$ such that $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^d$ and $\forall i = 1, ..., N$.

Assumption 2 (Bounded Variance): The dataset $\mathcal{D}_s^t$ at the server approximates the overall population distribution $\mathcal{V}$, so the gradient calculated using $\mathcal{D}_s^t$ is an unbiased estimate of the global objective, i.e., $\mathbb{E}_{\mathcal{D}_s^t \sim \mathcal{V}}[\nabla f_s(x)] = \nabla f(x)$. Furthermore, there exists a constant $\sigma > 0$ such that the variance of the gradient estimator is bounded, i.e.,

$$\mathbb{E}_{\mathcal{D}_s^t \sim \mathcal{V}}\left[\|\nabla f_s(x) - \nabla f(x)\|^2\right] \leq \frac{\sigma^2}{m_s}, \forall x, \forall t. \tag{6}$$

where m_s is the size of server dataset $\mathcal{D}_s^t$.

Assumption 3 (Unbiased Gradient Estimate and Bounded Local Variance): The stochastic gradient estimate is unbiased, i.e., $\mathbb{E}_\zeta[F_i(x, \zeta)] = \nabla f_i(x), \forall x$ and $\forall i = 1, \cdots, N$ and its variance is bounded $\mathbb{E}[\|\nabla F_i(x, \zeta_i) - \nabla f_i(x)\|^2] \leq \sigma_l^2, \ \forall x \in \mathbb{R}^d$ and $\forall i = 1, ..., N$.

Assumption 4 (Bounded Global Variance): There exists a constant number $\sigma_g > 0$ such that the variance between the local gradient of client i and the global gradient is bounded:

$$\|\nabla f_i(x) - \nabla f(x)\|^2 \leq \sigma_g^2, \ \forall i \in [N], \forall x. \tag{7}$$

Assumptions 1, 3 and 4 are commonly adopted in the convergence analysis of FL under non-IID settings [7], [8], [9], [20]. Assumption 2 provides a bound on the variance introduced by server local training, which is dependent on the size of $\mathcal{D}_s^t$ [39]. We here consider the size of $\mathcal{D}_s^t$ as a hyper-parameter.

Theorem 1: Suppose that client local learning rate η, global learning rate η_g, and server local learning rate γ are chosen such that $\eta \leq \frac{1}{3KL}, \eta\eta_g \leq \frac{1}{27KL}$, and $\gamma \leq \frac{1}{6EL}$. Under Assumptions 1, 2, 3, 4, suppose that in each round t the server uniformly selects M out of N clients without replacement, the sequence of model vectors x_t satisfies:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{(f_0 - f_*)}{T(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^3 \sigma_g^2}{\gamma E + \eta\eta_g K}\right) + \mathcal{O}\left(\frac{\gamma^2 E L \sigma^2}{m_s(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{(N-M)K^2\eta^2\eta_g^2 L\sigma_g^2}{M(N-1)(\gamma E + \eta\eta_g K)}\right) + \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^2 \sigma_l^2}{\gamma E + \eta\eta_g K}\right) \\ &+ \mathcal{O}\left(\frac{\eta^2\eta_g^2 LK\sigma_l^2}{M(\gamma E + \eta\eta_g K)}\right),\end{aligned} \tag{8}$$

Proof: The proof is shown in Appendix [A]. □

Remark 1: The convergence bound presented above consists of six terms, with the second term accounting for the effect of client local training, the third term representing the impact of limited data availability at the server, the fourth term reflecting the influence of partial participation of clients, and the last two terms representing the error caused stochastic client updates.

Remark 2: The third term is influenced by the number of data points available at the server, denoted by m_s. As m_s increases, the convergence bound tightens, which aligns with the expectation that having more training data stored at the server should result in better convergence performance.

Corollary 1: Let $\eta = \Theta(\frac{1}{K\sqrt{T}})$, $\eta_g = \Theta(\sqrt{MK})$ and $\gamma = \Theta(\frac{1}{\sqrt{ET}})$, the convergence rate of CLG-SGD becomes:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{\sqrt{MK}}{(\sqrt{MK} + \sqrt{E})T}\right) \\ &+ \mathcal{O}\left(\frac{K}{(\sqrt{MK} + \sqrt{E})\sqrt{T}}\right)\end{aligned} \tag{9}$$

Remark 3: Our convergence analysis of CLG-SGD is more general than that of [11], as we consider non-IID data and partial client participation. Additionally, our analysis demonstrates the theoretical benefits of additional server training even in the case of IID data and full participation, where [11] fails to do so. Notably, in the case of IID data ($\sigma_g = 0$) or full participation ($M = N$), our convergence rate is $\mathcal{O}\left(\frac{1}{(\sqrt{MK}+\sqrt{E})\sqrt{T}}\right) + \mathcal{O}\left(\frac{\sqrt{MK}}{(\sqrt{MK}+\sqrt{E})T}\right)$, which converges faster than the rate of $\mathcal{O}\left(\frac{1}{\sqrt{MKT}}\right) + \mathcal{O}\left(\frac{1}{T}\right)$ found in [11].

Remark 4: If we consider full client participation and set the server's local training epoch $E = 0$, the hybrid FL approach becomes equivalent to classic FL, and the convergence speed degenerates to $\mathcal{O}\left(\frac{1}{\sqrt{MKT}}\right) + \mathcal{O}\left(\frac{1}{T}\right)$. This rate is the same as the state-of-the-art rate found in classical FL [8], [23].

Remark 5: The corollary reveals that the dominating factor in the convergence bound is $\mathcal{O}\left(\frac{K}{(\sqrt{MK}+\sqrt{E})\sqrt{T}}\right)$, which is closely related to the global variance σ_g^2. This suggests that the global variance has a more significant effect on convergence behavior in cases with partial participation, particularly in highly non-IID scenarios where σ_g is substantial. Therefore, developing a new hybrid FL approach to mitigate the negative effects of partial participation is a challenging task.

Our novel theoretical analysis of CLG-SGD suggests that hybrid FL can achieve faster convergence by incorporating additional server local training after the aggregation step. However, like classic FL, hybrid FL is still constrained by the convergence limitations caused by partial client participation in non-IID settings. Prior research, such as [23], [35], has used variance reduction techniques in classic FL to mitigate the adverse effects of partial participation. Although these methods can be adapted to hybrid FL, they do not fully leverage the potential benefits of the small amount of server data. In the next sections,

we introduce FedCLG, a novel algorithm that fully exploits the server data.

IV. FedCLG

In hybrid FL, a significant difference from the classical FL setting is the server's possession of its local training dataset $\mathcal{D}_s^t$, which is a small subset approximating the overall population dataset. While using only the server dataset $\mathcal{D}_s^t$ to train a model has drawbacks, such as slower training speed and higher risk of reaching sub-optimal points, it can provide a more accurate direction of the global objective compared to using the large non-IID dataset stored at each client. The key innovation of FedCLG is the utilization of the server gradient to produce variance correction either at the server aggregation step (FedCLG-S) or the client local training step (FedCLG-C). This correction helps address the issue of non-IID data distribution across clients and ultimately improves the FL model's accuracy. In this section, we provide further elaboration on the specific details of each version of FedCLG.

A. FedCLG-C

In FedCLG-C, the server randomly selects a subset of M clients out of N total clients, denoted as $\mathcal{S}_t$, at each round t. Prior to broadcasting the global model x_t to the selected clients, the server conducts an additional local training step using its own local dataset $\mathcal{D}_s^t$ based on the global model x_t, producing a gradient denoted as g_t^s, where $g_t^s = \nabla f_s(x_t)$ represents the full batch gradient or $g_t^s = \nabla f_s(x_t, \zeta_s)$ represents the stochastic gradient evaluated on a randomly drawn mini-batch ζ_s from the server dataset $\mathcal{D}_s^t$. FedCLG-C requires the server to broadcast both the global model x_t and the gradient g_t^s to each selected client $i \in \mathcal{S}_t$. Upon receiving the gradient g_t^s and the model x_t, each client i performs K rounds of local epoch with the correction term c_i as follows:

$$\begin{aligned} c_i &= g_t^s - g_t^i \\ x_{t,0}^i &= x_t; \\ x_{t,k+1}^i &= x_{t,k}^i - \eta(g_{t,k}^i + c_i), \ k = 0, \ldots, K-1, \end{aligned} \tag{10}$$

Here, $g_t^i = \nabla f_i(x_t, \zeta_i)$ is the stochastic gradient evaluated on a randomly drawn mini-batch ζ_i at client i. ($g_t^i = \nabla f_i(x_t)$ if a full gradient is used). After completing K rounds of training, each client i sends its update to the server. The server then carries out the same aggregation and server's local training steps as described in Eqs. 4 and 5.

B. FedCLG-S

In FedCLG-S, similar to FedCLG-C, the server selects a random subset of M clients at each round t and performs an extra training step based on x_t using its own local dataset $\mathcal{D}_s^t$. The resulting training gradient g_t^s is held by the server, which subsequently broadcasts the global model x_t to the selected clients. Upon receipt of the global model, each selected client $i \in \mathcal{S}_t$ conducts K steps of local training as specified by Eq. 3 and calculates the client gradient g_t^i based on the received global model. Each client sends back its cumulative local updates and local gradient g_t^i to the server, which then aggregates the updates using the following formula:

$$x_{t+1}^s = x_t + \eta_g \frac{1}{M} \sum_{i \in \mathcal{S}_t} (\Delta_t^i - K\eta(g_t^s - g_t^i)), \tag{11}$$

where η_g is the learning rate for the server's local training. After this aggregation, the server performs E epochs of local training as described in Eq. 5. The steps involved in FedCLG-C and FedCLG-S are summarized in Algorithm 1.

The primary difference between FedCLG-C and FedCLG-S, which is shown in Fig. 2, lies in their methods for addressing the variance issue. FedCLG-C addresses the problem of partial participation during local training on the client side, while FedCLG-S corrects the partial participation error at the server side during the aggregation step. While FedCLG-C requires that

Algorithm 1 FedCLG

Initial model x_0, client local learning rate η, global learning rate η_g, server local learning rate γ, number of client local epoch K, number of server local epoch E, number of global iterations T
for $t = 0, 1, \ldots, T-1$ **do**
  Uniformly sample $\mathcal{S}_t$ clients without replacement
  Compute a server gradient $g_t^s = \nabla f_s(x_t, \zeta_s)$
  Client Side:
  for each client $i \in \mathcal{S}_t$ in parallel **do**
    if FedCLG-C **then**
      Perform client local training as Eq. 10
    else if FedCLG-S **then**
      Perform client local training as Eq. 3
    end if
  end for
  Server Side:
  if FedCLG-C **then**
    Aggregate the model x_{t+1}^s as Eq. 4
  else if FedCLG-S **then**
    Aggregate the model x_{t+1}^s as Eq. 11
  end if
  Perform server local training as Eq. 5
end for

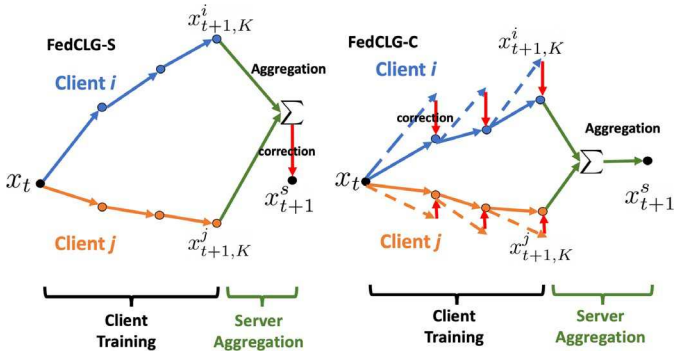


Fig. 2. The key distinction between FedCLG-S and FedCLG-C lies in the timing and location of the correction step. In FedCLG-S, the corrections occur during the server aggregation step, whereas in FedCLG-C, they take place during each client's local training step.

the server broadcast an additional gradient g_t^s during communication, FedCLG-S requires that each client upload an additional gradient g_t^i to the server per round. The choice between using FedCLG-S or FedCLG-C should be based on the available bandwidth for uploading and downloading. Specifically, if the download bandwidth is restricted, FedCLG-S should be utilized. Conversely, if the upload bandwidth is restricted, FedCLG-C should be used. To further address the communication efficiency concerns, our method can be integrated with quantization or compression techniques, which have been extensively studied in the FL setting (e.g. [40], [41], [42], [43]). These methods are capable of significantly reducing the communication costs associated with additional transmissions.

C. Comparison With FL Variance Reduction Methods

Existing variance reduction methods in FL do not consider the potential benefits of using the server dataset to reduce variance. SCAFFOLD, proposed in [23], is the first work to identify client drift error and utilize control variates to correct it. However, SCAFFOLD requires additional gradient communication during both the upload and download processes. Alternatively, FedCLG-C or FedCLG-S can be chosen based on different upload/download communication scenarios, reducing the overall communication workload. Other variance reduction methods, such as [35] and [36], require the server to maintain $\mathcal{O}(Nd)$ memory, where N is the number of total clients and d is the model size, which can be very expensive and unrealistic in cross-device settings of FL. Others [37], [38] require additional client computations. Moreover, all of the above methods use stale information to build the correction term c_i, which can negatively affect performance. Furthermore, none of these methods provide convergence guarantees under the hybrid FL setting. In the experimental section, we demonstrate the superiority of FedCLG.

V. Convergence Analysis of FedCLG

In this section, we will provide a convergence analysis of both versions of FedCLG in a non-convex setting. We will adopt the same assumptions as in the previous section, which were used for the convergence analysis of CLG-SGD.

Theorem 2: Suppose that client local learning rate η, global learning rate η_g and server local learning rate γ are chosen such that $\eta \leq \frac{1}{8KL}$, $\eta\eta_g \leq \frac{1}{36KL}$ and $\gamma \leq \frac{1}{6EL}$. Under Assumptions 1, 2, 3, suppose in each round t the server uniformly selects M out of N clients without replacement, the sequence of FedCLG-C model vectors x_t satisfies:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{(f_0 - f_*)}{T(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^3 \sigma^2}{m_s(\gamma E + \eta\eta_g K)}\right) + \mathcal{O}\left(\frac{\gamma^2 LE\sigma^2}{m_s(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{\eta^2\eta_g^2 KL\sigma^2}{Mm_s(\gamma E + \eta\eta_g K)}\right) + \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^2\sigma_l^2}{\gamma E + \eta\eta_g K}\right) \\ &+ \mathcal{O}\left(\frac{\eta^2\eta_g^2 LK\sigma_l^2}{M(\gamma E + \eta\eta_g K)}\right), \end{aligned} \tag{12}$$

where $f_0 = f(x_0)$, $f_* = f(x_*)$.

Proof: The proof is shown in the supplementary material. □

Corollary 2: Let $\eta = \Theta(\frac{1}{K\sqrt{T}})$, $\eta_g = \Theta(\sqrt{MK})$ and $\gamma = \Theta(\frac{1}{\sqrt{ET}})$, the convergence rate of FedCLG-C becomes:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{1}{(\sqrt{MK} + \sqrt{E})\sqrt{T}}\right) \\ &+ \mathcal{O}\left(\frac{\sqrt{MK}}{(\sqrt{MK} + \sqrt{E})T}\right) \end{aligned} \tag{13}$$

Theorem 3: Suppose that client local learning rate η, global learning rate η_g and server local learning rate γ are chosen such that $\eta \leq \frac{1}{3KL}$, $\eta\eta_g \leq \frac{1}{27KL}$ and $\gamma \leq \frac{1}{6EL}$. Under Assumptions 1, 2, 3, 4, suppose that in each round t the sever uniformly selects M out of N clients without replacement, the sequence of FedCLG-S model vectors x_t satisfies:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{(f_0 - f_*)}{T(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^3 \sigma_g^2}{\gamma E + \eta\eta_g K}\right) + \mathcal{O}\left(\frac{\gamma^2 LE\sigma^2}{m_s(\gamma E + \eta\eta_g K)}\right) \\ &+ \mathcal{O}\left(\frac{\eta^2\eta_g^2 KL\sigma^2}{Mm_s(\gamma E + \eta\eta_g K)}\right) + \mathcal{O}\left(\frac{\eta^3\eta_g L^2 K^2\sigma_l^2}{\gamma E + \eta\eta_g K}\right) \\ &+ \mathcal{O}\left(\frac{\eta^2\eta_g^2 LK\sigma_l^2}{M(\gamma E + \eta\eta_g K)}\right), \end{aligned} \tag{14}$$

where $f_0 = f(x_0)$ and $f_* = f(x_*)$.

Proof: The proof is shown in the supplementary material. □

Corollary 3: Let $\eta = \Theta(\frac{1}{K\sqrt{T}})$, $\eta_g = \Theta(\sqrt{MK})$ and $\gamma = \Theta(\frac{1}{\sqrt{ET}})$, the convergence rate of FedCLG-S becomes:

$$\begin{aligned}\min_{t\in[T]} \mathbb{E}\|\nabla f(x_t)\|_2^2 &= \mathcal{O}\left(\frac{1}{(\sqrt{MK} + \sqrt{E})\sqrt{T}}\right) \\ &+ \mathcal{O}\left(\frac{\sqrt{MK}}{(\sqrt{MK} + \sqrt{E})T}\right) \end{aligned} \tag{15}$$

Remark 6: Both FedCLG-C and FedCLG-S's convergence results contain six terms where the second term is client local drift error, the third term is server training update error, the fourth term captures the stochastic error due to the limited size of the server's subset data compared to the overall population and the last two terms are stochastic client update error. Notably, both FedCLG-S and FedCLG-C eliminate the partial participation error. Additionally, the bound becomes increasingly dependent on the size of the dataset stored at the server, which is expected since we use the server gradient to guide client updates. A larger server dataset can result in a more accurate server gradient direction, leading to a tighter overall bound. While a smaller m_s may produce less accurate corrections than a larger m_s, our experiments demonstrate that even with a small m_s, our proposed method can significantly enhance convergence speed under extremely non-IID settings.

Remark 7: The primary distinction between FedCLG-C and FedCLG-S lies in the second term, which addresses client local drift error. This is justifiable as FedCLG-C incorporates a

correction term at each client's local training step, making it less dependent on global objective variance and more reliant on the quality of the correction step. Conversely, FedCLG-S applies the correction step during server aggregation, allowing client local training to be influenced by the variance between local and global objectives. This indicates that FedCLG-C may yield marginally better results when server datasets are reliable (i.e., low $\frac{\sigma^2}{m_s}$) and client data heterogeneity is high. However, FedCLG-S remains a robust choice, outperforming baseline methods (as shown in the experiment section). Under less extreme non-IID conditions, the choice between FedCLG-S and FedCLG-C should be influenced by bandwidth considerations.

Remark 8: Under partial client participation setting, assume the server's local training epoch $E = 0$, which reduces hybrid FL setting to classic FL setting, the convergence rates of FedCLG-C and FedCLG-S reduce to $\mathcal{O}\left(\frac{1}{\sqrt{MKT}}\right) + \mathcal{O}\left(\frac{1}{T}\right)$ which match the convergence rate achieved by the SOTA variance reduction methods [23], [35] used in the classic FL setting with partial client participation.

Remark 9: To prevent client local drift error from dominating the convergence process, aiming for a convergence rate of $\mathcal{O}\left(\frac{1}{(\sqrt{MK}+\sqrt{E})\sqrt{T}}\right)$, both FedCLG-S and FedCLG-C need that the local epoch K should not surpass T/M.

Remark 10: Both FedCLG-C and FedCLG-S exhibit convergence rates of $\mathcal{O}\left(\frac{1}{(\sqrt{MK}+\sqrt{E})\sqrt{T}}\right)$ under non-IID and partial participation settings, provided that there are enough training rounds T (i.e. $T \geq KM$). This rate is faster than that of CLG-SGD, which converges with a rate dominated by $\mathcal{O}\left(\frac{K}{(\sqrt{MK}+\sqrt{E})\sqrt{T}}\right)$. Interestingly, larger client-side local training epochs K can actually hurt the convergence rate for CLG-SGD due to the negative effects of partial participation. However, after eliminating these negative effects in both FedCLG-S and FedCLG-C, the new convergence rates show that larger client-side local training epochs K can actually increase the convergence rate.

VI. Experiments

A. Setup

We conducted all experiments using Federated Learning (FL) simulation on the PyTorch framework and trained the models on Geforce RTX 3080 GPUs. We performed five random repeats and reported the averaged results. The detailed experimental settings are presented below.

1) Dataset and Backbone Model: To verify our theoretical findings, we evaluate the proposed methods on three datasets:

MNIST [44]: We utilize LeNet-5 [45] as the backbone model. The default training hyperparameters are as follows: server local learning rate tuning from $\gamma = \{0.01, 0.05, 0.25\}$, client local learning rate tuning from $\eta = \{0.01, 0.05, 0.25\}$, local learning rates' decay factor equals to 0.99 until learning rate reaches 0.001, global learning rate $\eta_g = 1$, and training batch sizes at both the server and clients set to 64.

CIFAR-10 [46]: We also utilize LeNet-5 as the backbone model. The default training hyperparameters are server local learning rate tuning from $\gamma = \{0.01, 0.05, 0.25\}$, client local learning rate tuning from $\eta = \{0.02, 0.08, 0.32\}$, local learning rates' decay factor equals to 0.99 until learning rate reaches 0.001, global learning rate $\eta_g = 1$, and training batch sizes at both the server and clients set to 128.

CIFAR-100 [46]: For CIFAR-100 datasets, we use 20 superclasses to reclassify the data samples. We utilize MobileNetV2 [47] as the backbone model. The default training hyperparameters are server local learning rate tuning from $\gamma = \{0.005, 0.05, 0.5\}$, client local learning rate tuning from $\eta = \{0.01, 0.1, 1\}$, local learning rates' decay factor equals to 0.99 until learning rate reaches 0.001, global learning rate $\eta_g = 1$, and training batch sizes at both the server and clients set to 128.

2) Client Setting: Our experimental evaluations cover both IID and non-IID client datasets. Without loss of generality, we assume that each client has an equal number of data samples. Specifically, for MNIST, we simulate 200 clients with 150 data samples each, for CIFAR-10 and CIFAR-100 we simulate 200 clients with 200 data samples each.

IID and Non-IID scenario. For IID datasets, we randomly assign an equal-size local dataset from the total training set to each client. For non-IID datasets, we apply the Dirichlet method (α) to create the data distribution for each client. We use various α values to demonstrate the convergence performance under different degrees of non-IID.

Client participation. The primary objective of this paper is to investigate how the server dataset can be leveraged to eliminate the partial participation error in Federated Learning. Therefore, in the experiment section, we will focus on the scenario where clients participate partially in the training process. Specifically, in the main experiments, we uniformly sample $M = 4$ clients without replacement in every round for MNIST dataset, and $M = 10$ clients for CIFAR-10 and CIFAR-100 dataset.

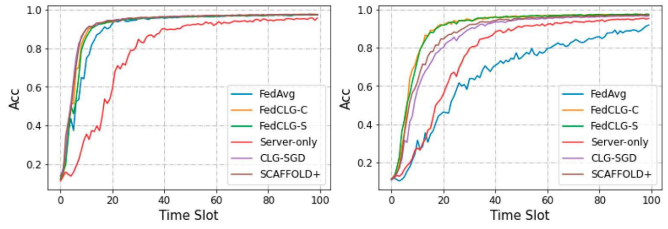
Number of Local Epoch K. To investigate the impact of the client's local training epoch, we vary the value of K to be 1, 3, or 5.

3) Server Setting: Compared to classical FL, Hybrid FL introduces the novel setting where the server itself contains a subset of the population dataset.

Size of server dataset m_s. In the MNIST experiment, we consider the server dataset to contain 1% of the total training data samples, while in the CIFAR-10 and CIFAR-100 experiments, we consider it to contain 5% of the total training data samples. In the ablation studies, we change the size of the server dataset to examine its impact on the hybrid FL approach.

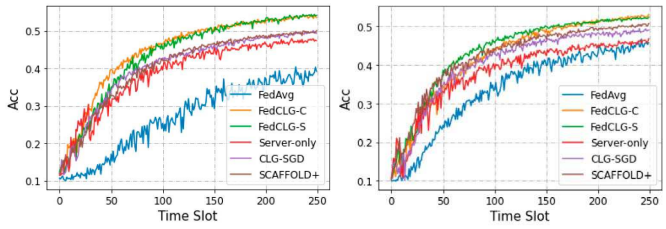
Number of Local Epoch E. To investigate the impact of the server's local training epoch, we vary the value of E to be 1, 3, or 5.

4) Baselines: In our study, we evaluate the performance of our approach against the following established methods: (1) **Server-only**: This method involves training a model solely on the server's dataset, denoted as $\mathcal{D}_s^t$, without using any client data. (2) **FedAvg**: As a key baseline in traditional Federated Learning (FL), FedAvg employs client-side data for distributed learning. It is worth noting that the server's dataset is not used in this method. (3) **CLG-SGD**: This state-of-the-art hybrid FL



(a) MNIST IID (b) MNIST Non-IID ($\alpha = 0.2$)

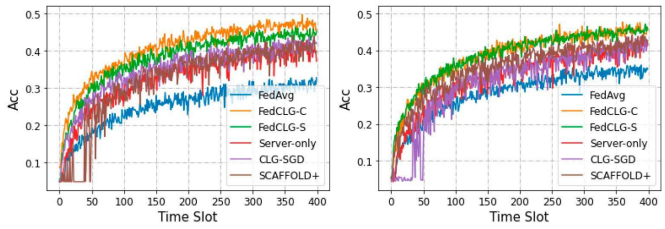
Fig. 3. Convergence performances on MNIST.



(a) CIFAR-10 Non-IID ($\alpha = 0.3$) (b) CIFAR-10 Non-IID ($\alpha = 0.8$)

Fig. 4. Convergence performances on CIFAR-10.

technique, introduced in [11], alternates between server local training and client local training. (4) **SCAFFOLD+**: SCAFFOLD [23] is a leading FL variance reduction method. To enable a fair comparison that highlights the advantages of our proposed approach, we have incorporated the alternating training concept from CLG-SGD into SCAFFOLD. Consequently, SCAFFOLD+ is an improved version of SCAFFOLD that utilizes the server's local dataset for additional training.



(a) CIFAR-100 Non-IID ($\alpha = 0.5$) (b) CIFAR-100 Non-IID ($\alpha = 0.8$)

Fig. 5. Convergence performances on CIFAR-100.

B. Experiments Results

The primary objective of the experiments is to showcase the differences in the number of global training rounds required by different methods to achieve a specific test accuracy, thereby highlighting the differences in their convergence speeds.

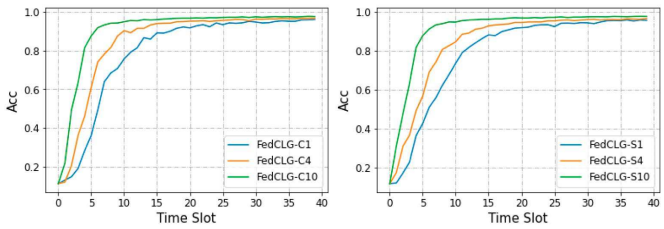
Performances Comparison. We first compare the convergence performance of our proposed methods and baselines on the MNIST dataset. We consider both IID and non-IID settings. As shown in Fig. 3(a), for the IID setting, FedCLG-S and FedCLG-C outperform FedAvg and Server-Only, but only achieve comparable performance with CLG-SGD (FedCLG-S even performs slightly worse). This is expected, as there is no variance reduction needed in the IID setting, i.e., $\sigma_g = 0$. Introducing the correction step can bring additional errors caused by the variance of server gradient, resulting in a marginal benefit (or slight weakness) in the IID setting. However, in most realistic scenarios, non-IID data distribution is more common. As seen in Fig. 3(b), under the non-IID setting ($\alpha = 0.2$), even for the MNIST dataset, both FedCLG-S and FedCLG-C outperform the other baselines. Moreover, we observe that in the IID setting, FedAvg converges faster than Server-Only, but in the non-IID setting, the convergence speed of FedAvg decreases significantly. In contrast, for the methods applying additional server training, the convergence speed does not decrease significantly even under the non-IID case. This further validates the necessity of additional server local training, which is consistent with findings in [11].

We evaluate the proposed methods on CIFAR-10 and CIFAR-100 under non-IID settings. As shown in Figs. 4 and 5, FedCLG-S and FedCLG-C outperform CLG-SGD by a significant margin. Specifically, in Fig. 4(a), setting the target test accuracy to 0.5, FedCLG-C requires 134 global rounds, and FedCLG-S requires 144 global rounds. In contrast, CLG-SGD requires 246 global rounds, which is 1.83 (and 1.70) times more than FedCLG-C and FedCLG-S. Furthermore, under high non-IID conditions, FedCLG-C slightly outperforms FedCLG-S, aligning with our theoretical predictions. Despite this, FedCLG-S still significantly surpasses the existing baseline. In scenarios with lower non-IID conditions, both versions of FedCLG exhibit comparable performance. Consequently, the choice between FedCLG-C and FedCLG-S should be based on the conditions of upload/download communication bandwidth. Moreover, although SCAFFOLD+ also applies variance reduction methods, it can only achieve comparable performance with CLG-SGD, indicating that such variance reduction fails to work. We attribute this failure to the use of stale estimated global and local gradients as the guideline to correct the variance, which introduces additional error. The inability to directly apply SCAFFOLD-related methods, which use stale information, further underscores the importance of reasonable exploitation of the server's local data and the necessity of our proposed method, FedCLG.

Impact of number of participated clients M. In this series of experiments, we investigate the impact of the number of participated clients M on the convergence speed of our proposed methods and the baseline method, CLG-SGD. We test the experiments on the MNIST dataset while holding all other parameters constant and only varying the number of participated clients M. The third column of Table I reports the number of global rounds required by each experiment, with the target test accuracy set to 97%. The results show that, for both the baseline and our proposed methods, increasing the number of participated clients leads to a decrease in the required number of global rounds and a higher convergence speed. Moreover, under different numbers of participated clients (i.e., 4, 6, 24), the convergence speeds of FedCLG-S and FedCLG-C outperform CLG-SGD. However, with a larger

TABLE I
IMPACT OF PARTICIPATED CLIENTS M

METHODS	PARTICIPATED CLIENTS	NUMBERS OF ROUND
CLG-SGD	4	68 (1.0×)
	6	59 (1.0×)
	24	39 (1.0×)
FedCLG-S	4	42 (1.61×)
	6	35 (1.68×)
	24	28 (1.39×)
FedCLG-C	4	39 (1.74×)
	6	32 (1.84×)
	24	25 (1.56×)



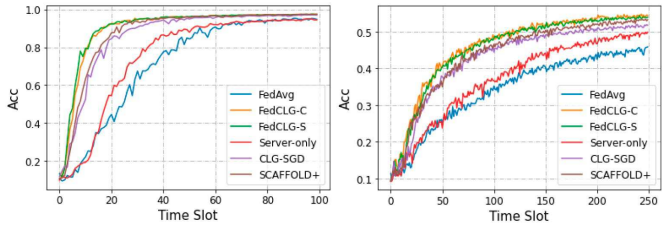
(a) FedCLG-C with different m_s (b) FedCLG-S with different m_s

Fig. 6. Impact of server dataset size m_s.

increase in the number of participated clients, the benefit of our proposed methods slightly decreases. For example, with $M = 4$, FedCLG-C achieves a $1.74\times$ speed-up compared to CLG-SGD, while with a larger number of clients ($M = 24$), the speed-up decreases to $1.56\times$. This observation is consistent with our theoretical analysis, which suggests that increasing the number of participated clients in CLG-SGD reduces the error caused by partial participation, thus leading to a smaller benefit from the correction step.

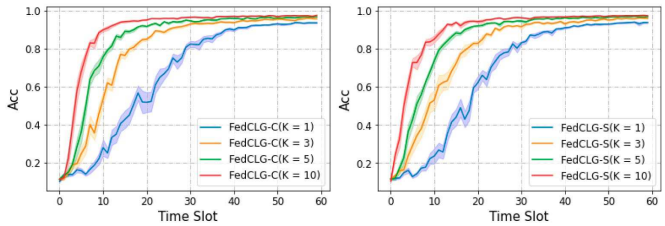
Impact of server dataset size. To investigate the impact of the server dataset, we first conduct an experiment on MNIST with different levels of m_s. In our initial setting, we assume that the server contained 1% of the total training samples each global round. We then extend this value to be 4% and 10%. The results, shown in Fig. 6, indicate that increasing the size of the server dataset can improve the overall convergence speeds of both FedCLG-C and FedCLG-S, as a larger server dataset provides a two-fold improvement. Firstly, more data can be acquired each round, leading to improved server local training. Secondly, the larger server dataset helps us to achieve a more reliable correction step with g_s approaching closer to the actual global optimal direction.

We then include experiments when the server dataset has a slight distribution shift. This shift is quantified using cosine similarity between the server and overall distributions, specifically targeting scenarios where the shift is small (cosine similarity is approximately 0.95). The results, illustrated in Fig. 7 for both MNIST and CIFAR-10 datasets, reveal that even with this slight distribution shift, our proposed method consistently outperforms baseline approaches. It is important to note that



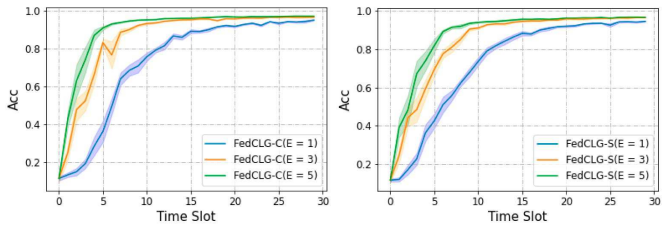
(a) MNIST (Similarity = 0.946) (b) CIFAR-10 (Similarity = 0.952)

Fig. 7. m_s Distribution Shift.



(a) FedCLG-C with different K (b) FedCLG-S with different K

Fig. 8. Impact of client epochs K.



(a) FedCLG-C with different E (b) FedCLG-S with different E

Fig. 9. Impact of server epochs E.

we do not explore scenarios involving significant distribution shifts, as such conditions would effectively reduce the server to another client role, a situation outside the scope of our study in hybrid federated learning.

Impact of client epochs K and server epochs E. In this subsection, we investigate the impact of the number of client epochs K and server epochs E. We keep other parameters fixed and only vary the number of client local epochs in Fig. 8. The convergence results shown in Fig. 8 are consistent with our theoretical analysis, as a larger number of client local epochs K results in increased convergence speed for both FedCLG-C and FedCLG-S. Similarly, in Fig. 9, we fix all parameters and only change the number of server local epochs. It can be observed that, for both FedCLG-C and FedCLG-S, the largest local epoch ($E = 5$) achieves the best convergence performance in both Fig. 9(a) and 9(b).

VII. CONCLUSION

In this paper, we address the hybrid Federated Learning (FL) setting, where the server has access to a small portion of the total training samples. We consider a more realistic scenario

where clients with non-IID data can only partially participate in each server aggregation. Firstly, we provide a novel theoretical analysis for CLG-SGD, the state-of-the-art hybrid FL method. Our analysis reveals the drawbacks of the current method due to clients' partial participation. Motivated by these observations, we propose a novel method called FedCLG, which fully exploits the benefits of a small server dataset. We further study two versions of FedCLG based on different server-client communication scenarios. We provide thorough theoretical analysis and experimental comparisons to validate the proposed methods. Future research will focus on developing a theoretically guaranteed method under an unbounded client-server communication pattern.

APPENDIX A
PROOF OF THEOREM 1

Note that in the following proof, we utilize $g_{t,k}^i$ to represent $\nabla F_i(x_{t,k}^i, \zeta_i)$. For each global round t, we have the intermediate model after client local training as:

$$x_{t+1}^s = x_t + \eta_g \frac{1}{M} \sum_{i \in \mathcal{S}_t} \Delta_t^i, \tag{16}$$

where $\Delta_t^i = -\sum_{k=0}^{K-1} \eta g_{t,k}^i$. The final model after server local training is as:

$$\begin{aligned} x_{t+1} &= x_{t+1}^s - \sum_{e=0}^{E-1} \gamma g_{t+1,e}^s \\ &= x_t - \eta\eta_g \frac{1}{M} \sum_{i \in \mathcal{S}_t} \sum_{k=0}^{K-1} g_{t,k}^i - \gamma \sum_{e=0}^{E-1} \nabla f_s(x_{t+1,e}^s). \end{aligned} \tag{17}$$

Due to the smoothness in Assumption 1, taking expectation of $f(x_{t+1})$ over the randomness at communication round t, we have:

$$\begin{aligned} \mathbb{E}_t[f(x_{t+1})] \leq f(x_t) &+ \underbrace{\left\langle \nabla f(x_t), \mathbb{E}_t[x_{t+1} - x_t] \right\rangle}_{T_1} \\ &+ \underbrace{\frac{L}{2} \mathbb{E}_t[\|x_{t+1} - x_t\|^2]}_{T_2}. \end{aligned} \tag{18}$$

We first bound T_1 as follows:

$$\begin{aligned} T_1 &= \left\langle \nabla f(x_t), \mathbb{E}_t[x_{t+1} - x_t] \right\rangle \\ &= -\eta\eta_g \left\langle \nabla f(x_t), \mathbb{E}_t \left[\frac{1}{M} \sum_{i \in \mathcal{S}_t} \sum_{k=0}^{K-1} g_{t,k}^i \right] \right\rangle \\ &\quad - \gamma \left\langle \nabla f(x_t), \mathbb{E}_t \left[\sum_{e=0}^{E-1} \nabla f_s(x_{t+1,e}^s) \right] \right\rangle \\ &= \underbrace{-\eta\eta_g \left\langle \nabla f(x_t), \mathbb{E}_t \left[\frac{1}{M} \sum_{i \in \mathcal{S}_t} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) \right] \right\rangle}_{T_3} \\ &\quad \underbrace{- \gamma \left\langle \nabla f(x_t), \mathbb{E}_t \left[\sum_{e=0}^{E-1} \nabla f_s(x_{t+1,e}^s) \right] \right\rangle}_{T_4}, \end{aligned} \tag{19}$$

There the second equality is due to the assumption of an unbiased local gradient estimate. The term T_3 can be bounded as follows:

$$\begin{aligned} T_3 &= -\eta\eta_g \left\langle \nabla f(x_t), \mathbb{E}_t \left[\frac{1}{M} \sum_{i \in \mathcal{S}_t} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) \right] \right\rangle \\ &= -\frac{\eta\eta_g}{K} \left\langle K \nabla f(x_t), \mathbb{E}_t \left[\frac{1}{N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) \right] \right\rangle \\ &= \underbrace{\frac{\eta\eta_g}{2K} \mathbb{E}_t \left[\| \frac{1}{N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) - K \nabla f(x_t) \|^2 \right]}_{T_5} \\ &\quad - \frac{\eta\eta_g K}{2} \|\nabla f(x_t)\|^2 - \frac{\eta\eta_g}{2K} \mathbb{E}_t \left[\| \frac{1}{N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) \|^2 \right]. \end{aligned} \tag{20}$$

The last equality is due to the fact that $< x, y >= \frac{1}{2}[\|x\|^2 + \|y\|^2 - \|x - y\|^2]$. Then we can bound T_5 as:

$$\begin{aligned} T_5 &= \frac{\eta\eta_g}{2K} \mathbb{E}_t \left[\| \frac{1}{N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \nabla f_i(x_{t,k}^i) - K \nabla f(x_t) \|^2 \right] \\ &= \frac{\eta\eta_g}{2K} \mathbb{E}_t \left[\| \frac{1}{N} \sum_{i \in [N]} \sum_{k=0}^{K-1} [\nabla f_i(x_{t,k}^i) - \nabla f_i(x_t)] \|^2 \right] \\ &\leq \frac{\eta\eta_g}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t [\| [\nabla f_i(x_{t,k}^i) - \nabla f_i(x_t)] \|^2] \\ &\leq \underbrace{\frac{\eta\eta_g L^2}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t [\|x_{t,k}^i - x_t\|^2]}_{T_6} \end{aligned} \tag{21}$$

The first inequality is based on Cauchy-Schwarz inequality. Then we have T_6 be bounded as:

$$\begin{aligned} T_6 &= \frac{\eta\eta_g L^2}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t \left[\| \eta \sum_{\tau=0}^{k} g_{t,\tau}^i \|^2 \right] \\ &= \frac{\eta\eta_g L^2}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t \left[\| \eta \sum_{\tau=0}^{k} (g_{t,\tau}^i - \nabla f_i(x_{t,k}^i)) \|^2 \right] \\ &\quad + \frac{\eta\eta_g L^2}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t \left[\| \eta \sum_{\tau=0}^{k} \nabla f_i(x_{t,k}^i) \|^2 \right] \\ &= \frac{\eta\eta_g L^2}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t \left[\| \eta \sum_{\tau=0}^{k} (g_{t,\tau}^i - \nabla f_i(x_{t,k}^i)) \|^2 \right] \\ &\quad + \frac{\eta\eta_g L^2 K}{2N} \sum_{i \in [N]} \sum_{k=0}^{K-1} \mathbb{E}_t \left[\sum_{\tau=0}^{k} \| \eta \nabla f_i(x_{t,k}^i) \|^2 \right] \end{aligned}$$

$$\leq \frac{\eta^3\eta_g L^2K^2}{2}\sum_{k=0}^{K-1}\underbrace{\frac{1}{N}\sum_{i\in[N]}\mathbb{E}_t[\|\nabla f_i(x_{t,k}^i)\|^2]}_{T_7} + \frac{\eta^3\eta_g L^2K^2\sigma_l^2}{2}, \tag{22}$$

where the second equality is based on the assumption 4.

To further bound T_7, we have:

$$\begin{aligned} T_7 &\leq \frac{3}{N}\sum_{i\in[N]}\mathbb{E}_t[\|\nabla f_i(x_{t,k}^i)-\nabla f_i(x_t)\|^2] \\ &\quad + \frac{3}{N}\sum_{i\in[N]}\mathbb{E}_t[\|\nabla f_i(x_t)-\nabla f(x_t)\|^2] \\ &\quad + \frac{3}{N}\sum_{i\in[N]}\mathbb{E}_t[\|\nabla f(x_t)\|^2] \\ &\leq \frac{3L^2}{N}\sum_{i\in[N]}\mathbb{E}_t[\|x_t-x_{t,k}^i\|^2] + 3\sigma_g^2 + 3\mathbb{E}_t[\|\nabla f(x_t)\|^2], \end{aligned} \tag{23}$$

where the last inequality is due to the assumptions 1, 4. Substituting T_7 to (22), we have:

$$\begin{aligned} T_6 &\leq \frac{\eta^3\eta_g L^2K^2}{2N}\sum_{i\in[N]}\sum_{k=0}^{K-1}(3L^2\mathbb{E}_t[\|x_t-x_{t,k}^i\|^2] \\ &\quad + 3\sigma_g^2 + 3\mathbb{E}_t[\|\nabla f(x_t)\|^2]) + \frac{\eta^3\eta_g L^2K^2\sigma_l^2}{2} \\ &\leq \frac{3\eta^3\eta_g L^2K^3}{2(1-\mathcal{B})}\sigma_g^2 + \frac{\eta^3\eta_g L^2K^2\sigma_l^2}{2(1-\mathcal{B})} \\ &\quad + \frac{3\eta^3\eta_g L^2K^3}{2(1-\mathcal{B})}\mathbb{E}_t[\|\nabla f(x_t)\|^2] \\ &\leq \frac{3\eta^3\eta_g L^2K^2(\sigma_l^2+3K\sigma_g^2)}{4} + \frac{\eta\eta_g K}{4}\mathbb{E}_t[\|\nabla f(x_t)\|^2] \end{aligned} \tag{24}$$

where $\mathcal{B}=3\eta^2L^2K^2$ and let $\eta\leq\frac{1}{3LK}$ such that $\mathcal{B}\leq\frac{1}{3}$, $\frac{1}{1-\mathcal{B}}\leq\frac{3}{2}$ and $\frac{\mathcal{B}}{1-\mathcal{B}}\leq\frac{1}{2}$. Then we substitute it to (20), we can get:

$$\begin{aligned} T_3 &\leq \frac{3\eta^3\eta_g L^2K^2(\sigma_l^2+3K\sigma_g^2)}{4} - \frac{\eta\eta_g K}{4}\|\nabla f(x_t)\|^2 \\ &\quad - \frac{\eta\eta_g}{2K}\mathbb{E}_t\Big[\|\frac{1}{N}\sum_{i\in[N]}\sum_{k=0}^{K-1}\nabla f_i(x_{t,k}^i)\|^2\Big]. \end{aligned} \tag{25}$$

Next, the term T_4 can be bounded as

$$\begin{aligned} T_4 &= -\frac{\gamma}{E}\Big\langle E\nabla f(x_t), \mathbb{E}_t\Big[\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\Big]\Big\rangle \\ &= \underbrace{\frac{\gamma}{2E}\mathbb{E}_t\Big[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s) - E\nabla f(x_t)\|^2\Big]}_{T_8} \\ &\quad - \frac{\gamma E}{2}\|\nabla f(x_t)\|^2 - \frac{\gamma}{2E}\mathbb{E}_t\Big[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\Big]. \end{aligned} \tag{26}$$

The last equality is due to the fact that $<x,y>=\frac{1}{2}[\|x\|^2+\|y\|^2-\|x-y\|^2]$. Then the term T_8 can be bounded as follows:

$$\begin{aligned} T_8 &= \frac{\gamma}{2E}\mathbb{E}_t\Big[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s) - E\nabla f(x_t)\|^2\Big] \\ &\leq \frac{\gamma}{2}\sum_{e=0}^{E-1}\mathbb{E}_t[\|\nabla f(x_{t+1,e}^s)-\nabla f(x_t)\|^2] \\ &\leq \underbrace{\frac{\gamma L^2}{2}\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s - x_t\|^2]}_{T_9} \\ &\leq \gamma L^2\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s - x_{t+1}^s\|^2] + \gamma L^2\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1}^s - x_t\|^2] \\ &\leq \underbrace{\gamma L^2\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s - x_{t+1}^s\|^2]}_{T_{10}} + \frac{\eta^2\eta_g^2\gamma EL^2K\sigma_l^2}{M} \\ &\quad + \eta^2\eta_g^2\gamma EL^2\underbrace{\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}\nabla f_i(x_{t,k}^i)\|^2\Big]}_{T_{11}}. \end{aligned} \tag{27}$$

Then bounding T_{10}, we have:

$$\begin{aligned} T_{10} &= \gamma L^2\sum_{e=0}^{E-1}\mathbb{E}_t\Big[\|\sum_{\tau_e=0}^{e-1}\gamma\nabla f_s(x_{t+1,\tau_e}^s)\|^2\Big] \\ &\leq \gamma L^2\sum_{e=0}^{E-1}\gamma^2\mathbb{E}_t\Big[\|\sum_{\tau_e=0}^{e-1}\nabla f_s(x_{t+1,e}^s)\|^2\Big] \\ &\leq 3\gamma^3L^2\sum_{e=0}^{E-1}\mathbb{E}_t\Big[\|\sum_{\tau_e=0}^{e-1}(\nabla f_s(x_{t+1,e}^s)-\nabla f(x_{t+1,e}^s))\|^2\Big] \\ &\quad + 3\gamma^3L^2\sum_{e=0}^{E-1}\mathbb{E}_t\Big[\|\sum_{\tau_e=0}^{e-1}(\nabla f(x_{t+1,e}^s)-\nabla f(x_t))\|^2\Big] \\ &\quad + 3\gamma^3L^2\sum_{e=0}^{E-1}\mathbb{E}_t\Big[\|\sum_{\tau_e=0}^{e-1}\nabla f(x_t)\|^2\Big] \\ &\leq \frac{3\gamma^3E^2L^2\sigma^2}{m_s} + 3\gamma^3E^2L^4\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s - x_t\|^2] \\ &\quad + 3\gamma^3E^2L^2\sum_{e=0}^{E-1}\mathbb{E}_t[\|\nabla f(x_t)\|^2], \end{aligned} \tag{28}$$

where the first term in the third inequality is due to the fact that $\mathbb{E}[\|x_1+\cdots+x_n\|^2]=\mathbb{E}[\|x_1\|^2+\cdots+\|x_n\|^2]$ if x_i is independent with zero mean and assumption 4,

Then to bound the term T_{11}, we have:

$$\begin{aligned} T_{11} &\leq 3\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}[\nabla f_i(x_{t,k}^i)-\nabla f_i(x_t)]\|^2\Big] \\ &\quad + 3\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}[\nabla f_i(x_t)-\nabla f(x_t)]\|^2\Big] \end{aligned}$$

$$
\begin{aligned}
&+3\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}\nabla f(x_t)\|^2\Big]\\
&\le 3\mathbb{E}_t\Big[\frac{1}{M}\sum_{i\in\mathcal{S}_t}\|\sum_{k=0}^{K-1}[\nabla f_i(x_{t,k}^i)-\nabla f_i(x_t)]\|^2\Big]\\
&\quad+3\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}[\nabla f_i(x_t)-\nabla f(x_t)]\|^2\Big]\\
&\quad+3K^2\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\le \frac{3}{N}\sum_{i\in[N]}\mathbb{E}_t\Big[\|\sum_{k=0}^{K-1}[\nabla f_i(x_{t,k}^i)-\nabla f_i(x_t)]\|^2\Big]\\
&\quad+\underbrace{3\,\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}[\nabla f_i(x_t)-\nabla f(x_t)]\|^2\Big]}_{T_{12}}\\
&\quad+3K^2\mathbb{E}_t[\|\nabla f(x_t)\|^2], \qquad (29)
\end{aligned}
$$

where the last inequality is due to the server's uniformly selection without replacement. Next we need to bound T_{12}. For convenience, we utilize $\delta_t^i=\sum_{k=0}^{K-1}\nabla f_i(x_t)$ and $\delta_t=\sum_{k=0}^{K-1}\nabla f(x_t)$ in the following step.

$$
\begin{aligned}
T_{12}&=\mathbb{E}_t\Big[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\delta_t^i-\delta_t\|^2\Big]\\
&=\frac{1}{M^2}\mathbb{E}_t\Big[\|\sum_{i\in[N]}(\mathcal{I}(i\in\mathcal{S}_t))(\delta_t^i-\delta_t)\|^2\Big]\\
&=\frac{1}{M^2}\mathbb{E}_t\Big[\sum_{i\in[N]}(\mathcal{I}(i\in\mathcal{S}_t))^2\|\delta_t^i-\delta_t\|^2\Big]\\
&\quad+\frac{1}{M^2}\mathbb{E}_t\Big[\sum_{i\in[N]}\sum_{j\ne i\in[N]}\mathcal{I}(i\in\mathcal{S}_t)\mathcal{I}(j\in\mathcal{S}_t)\Big\langle\delta_t^i-\delta_t,\delta_t^j-\delta_t\Big\rangle\Big]\\
&=\frac{1}{M^2}\frac{M}{N}\mathbb{E}_t\Big[\sum_{i\in[N]}\|\delta_t^i-\delta_t\|^2\Big]\\
&\quad+\frac{M(M-1)}{N(N-1)}\frac{1}{M^2}\mathbb{E}_t\Big[\|\sum_{i\in[N]}(\delta_t^i-\delta_t)\|^2\Big]\\
&\quad-\frac{M(M-1)}{N(N-1)}\frac{1}{M^2}\mathbb{E}_t\Big[\sum_{i\in[N]}\|\delta_t^i-\delta_t\|^2\Big]\\
&=\frac{N-M}{MN(N-1)}\sum_{i\in[N]}\mathbb{E}_t[\|\delta_t^i-\delta_t\|^2]\\
&\le\frac{(N-M)K}{MN(N-1)}\sum_{i\in[N]}\sum_{k=0}^{K-1}\mathbb{E}_t[\|\nabla f_i(x_t)-\nabla f(x_t)\|^2]\\
&\le\frac{(N-M)K^2}{M(N-1)}\sigma_g^2, \qquad (30)
\end{aligned}
$$

where the second equality is due to the server's uniform selection without replacement and the third equality is due to $\sum_{i\in[N]}(\delta_t^i-\delta_t)=0$. Then substituting the result to (29):

$$
\begin{aligned}
T_{11}&\le\frac{3}{N}\sum_{i\in[N]}\mathbb{E}_t\Big[\|\sum_{k=0}^{K-1}[\nabla f_i(x_{t,k}^i)-\nabla f_i(x_t)]\|^2\Big]\\
&\quad+3\frac{(N-M)K^2}{M(N-1)}\sigma_g^2+3K^2\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\le\frac{9\eta^2L^2K^3(\sigma_l^2+3K\sigma_g^2)}{2}+\frac{3K^2}{2}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+3\frac{(N-M)K^2}{M(N-1)}\sigma_g^2+3K^2\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&=\frac{9\eta^2L^2K^3(\sigma_l^2+3K\sigma_g^2)}{2}\\
&\quad+\frac{9K^2}{2}\mathbb{E}_t[\|\nabla f(x_t)\|^2]+3\frac{(N-M)K^2}{M(N-1)}\sigma_g^2. \qquad (31)
\end{aligned}
$$

Substituting the results of T_{10} and T_{11}, we have:

$$
\begin{aligned}
T_9&\le\frac{3\gamma^3E^2L^2\sigma^2}{m_s}+3\gamma^3E^2L^4\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s-x_t\|^2]\\
&\quad+3\gamma^3E^3L^2\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+\eta^2\eta_g^2\gamma EL^2\frac{9\eta^2L^2K^3(\sigma_l^2+3K\sigma_g^2)}{2}\\
&\quad+(\eta^2\eta_g^2\gamma EL^2)\frac{9K^2}{2}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+3(\eta^2\eta_g^2\gamma EL^2)\frac{(N-M)K^2}{M(N-1)}\sigma_g^2\\
&\quad+\frac{\eta^2\eta_g^2\gamma EL^2K\sigma_l^2}{M}\\
&\le\frac{3\gamma^3E^2L^2\sigma^2}{m_s}+3\gamma^3E^2L^4\sum_{e=0}^{E-1}\mathbb{E}_t[\|x_{t+1,e}^s-x_t\|^2]\\
&\quad+\frac{\eta^2\eta_g^2\gamma EL^2K\sigma_l^2}{M}\\
&\quad+3\gamma^3E^3L^2\mathbb{E}_t[\|\nabla f(x_t)\|^2]+\frac{3\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{4}\\
&\quad+\frac{3\eta^2\eta_g^2LK^2}{4}\mathbb{E}_t[\|\nabla f(x_t)\|^2]+\frac{\eta^2\eta_g^2L(N-M)K^2}{2M(N-1)}\sigma_g^2\\
&\le\frac{3\gamma^3E^2L^2\sigma^2}{(1-\mathcal{A})m_s}+\frac{\mathcal{A}\gamma E}{2(1-\mathcal{A})}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+\frac{3\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{4(1-\mathcal{A})}+\frac{\eta^2\eta_g^2\gamma EL^2K\sigma_l^2}{M(1-\mathcal{A})}\\
&\quad+\frac{3\eta^2\eta_g^2LK^2}{4(1-\mathcal{A})}\mathbb{E}_t[\|\nabla f(x_t)\|^2]+\frac{\eta^2\eta_g^2L(N-M)K^2}{2M(N-1)(1-\mathcal{A})}\sigma_g^2\\
&\le\frac{18\gamma^3E^2L^2\sigma^2}{5m_s}+\frac{\gamma E}{10}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+\frac{9\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{10}+\frac{9\eta^2\eta_g^2LK^2}{10}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+\frac{3\eta^2\eta_g^2L(N-M)K^2}{5M(N-1)}\sigma_g^2+\frac{\eta^2\eta_g^2LK\sigma_l^2}{5M}, \qquad (32)
\end{aligned}
$$

where $\mathcal{A}=6\gamma^2L^2E^2$ and let $\gamma\le\frac{1}{6LE}$ such that $\mathcal{A}\le\frac{1}{6}$, $\frac{1}{1-\mathcal{A}}\le\frac{6}{5}$ and $\frac{\mathcal{A}}{1-\mathcal{A}}\le\frac{1}{5}$. Substituting the above result to (26), we have:

$$
\begin{aligned}
T_4&\le\frac{18\gamma^3E^2L^2\sigma^2}{5m_s}-\frac{2\gamma E}{5}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\
&\quad+\frac{9\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{10}+\frac{9\eta^2\eta_g^2LK^2}{10}\mathbb{E}_t[\|\nabla f(x_t)\|^2]
\end{aligned}
$$

$$+\frac{3\eta^2\eta_g^2L(N-M)K^2}{5M(N-1)}\sigma_g^2-\frac{\gamma}{2E}\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\right]+\frac{\eta^2\eta_g^2LK\sigma_l^2}{5M}. \tag{33}$$

Combing the inequalities of T_3 and T_4, we can bound T_1:

$$\begin{aligned}T_1\le&\frac{3\eta^3\eta_gL^2K^2(\sigma_l^2+3K\sigma_g^2)}{4}-\left(\frac{2\gamma E}{5}+\frac{\eta\eta_gK}{4}\right)\|\nabla f(x_t)\|^2\\&-\frac{\eta\eta_g}{2K}\mathbb{E}_t\left[\|\frac{1}{N}\sum_{i\in[N]}\sum_{k=0}^{K-1}\nabla f_i(x_{t,k}^i)\|^2\right]+\frac{18\gamma^3E^2L^2\sigma^2}{5m_s}\\&+\frac{9\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{10}+\frac{9\eta^2\eta_g^2LK^2}{10}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\&+\frac{3\eta^2\eta_g^2L(N-M)K^2}{5M(N-1)}\sigma_g^2-\frac{\gamma}{2E}\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\right]\\&+\frac{\eta^2\eta_g^2LK\sigma_l^2}{5M}.\end{aligned} \tag{34}$$

The rest term T_2 can be bounded as:

$$T_2\le\eta^2\eta_g^2L\underbrace{\mathbb{E}_t\left[\|\frac{1}{M}\sum_{i\in\mathcal{S}_t}\sum_{k=0}^{K-1}\nabla f_i(x_{t,k}^i)\|^2\right]}_{T_{11}}+\frac{\eta^2\eta_g^2LK\sigma_l^2}{M}+\gamma^2L\underbrace{\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f_s(x_{t+1,e}^s)\|^2\right]}_{T_{13}}. \tag{35}$$

The only rest term is T_{13}, which can be bounded as:

$$\begin{aligned}T_{13}=&\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}(\nabla f_s(x_{t+1,e}^s)-\nabla f(x_{t+1,e}^s)+\nabla f(x_{t+1,e}^s))\|^2\right]\\=&\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}(\nabla f_s(x_{t+1,e}^s)-\nabla f(x_{t+1,e}^s))\|^2\right]\\&+\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\right]\le\frac{E\sigma^2}{m_s}\\&+\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\right],\end{aligned} \tag{36}$$

where the third equality is due to the fact that $\mathbb{E}[\|x\|^2]=\mathbb{E}[\|x-\mathbb{E}[x]\|^2]+\|\mathbb{E}[x]\|^2$ and the last inequality is due to assumption 2 and $\mathbb{E}[\|x_1+\cdots+x_n\|^2]\le n\mathbb{E}[\|x_1\|^2+\cdots+\|x_n\|^2]$. Substituting the result of T_{11} and T_{13}, we can finally bound T_2 as:

$$\begin{aligned}T_2\le&\frac{9\eta^4\eta_g^2L^3K^3(\sigma_l^2+3K\sigma_g^2)}{2}+\frac{9K^2\eta^2\eta_g^2L}{2}\mathbb{E}_t[\|\nabla f(x_t)\|^2]\\&+3\frac{(N-M)K^2\eta^2\eta_g^2L}{M(N-1)}\sigma_g^2+\frac{\gamma^2LE\sigma^2}{m_s}\\&+\gamma^2L\mathbb{E}_t\left[\|\sum_{e=0}^{E-1}\nabla f(x_{t+1,e}^s)\|^2\right]+\frac{\eta^2\eta_g^2LK\sigma_l^2}{M}.\end{aligned} \tag{37}$$

With both T_1 and T_2 bounded, we finally have:

$$\begin{aligned}\mathbb{E}_t[f(x_{t+1})]\le&f(x_t)-\left(\frac{2\gamma E}{5}+\frac{\eta\eta_gK}{20}\right)\|\nabla f(x_t)\|^2\\&+\frac{8\gamma^2EL\sigma^2}{5m_s}\\&+\frac{57\eta^3\eta_gL^2K^3\sigma_g^2}{20}+\frac{18\eta^2\eta_g^2L(N-M)K^2}{5M(N-1)}\sigma_g^2\\&+\frac{19\eta^3\eta_gL^2K^2\sigma_l^2}{20}+\frac{6\eta^2\eta_g^2LK\sigma_l^2}{5M},\end{aligned} \tag{38}$$

where $\gamma\le\frac{1}{6EL}$, $\eta\le\frac{1}{3KL}$ and $\eta\eta_g\le\frac{1}{27KL}$. Rearranging and summing from $t=0,\ldots,T-1$, we have the convergence as:

$$\begin{aligned}&\min_{t\in[T]}\mathbb{E}\|\nabla f(x_t)\|_2^2\\&=\mathcal{O}\left(\frac{(f_0-f_*)}{T(\gamma E+\eta\eta_gK)}\right)+\mathcal{O}\left(\frac{\eta^3\eta_gL^2K^3\sigma_g^2}{\gamma E+\eta\eta_gK}\right)\\&\quad+\mathcal{O}\left(\frac{\gamma^2EL\sigma^2}{m_s(\gamma E+\eta\eta_gK)}\right)+\mathcal{O}\left(\frac{(N-M)K^2\eta^2\eta_g^2L\sigma_g^2}{M(N-1)(\gamma E+\eta\eta_gK)}\right)\\&\quad+\underbrace{\mathcal{O}\left(\frac{\eta^3\eta_gL^2K^2\sigma_l^2}{\gamma E+\eta\eta_gK}\right)+\mathcal{O}\left(\frac{\eta^2\eta_g^2LK\sigma_l^2}{M(\gamma E+\eta\eta_gK)}\right)}_{\text{stochastic gradient error}},\end{aligned} \tag{39}$$

where $f_0=f(x_0)$, $f_*=f(x_*)$.

Now we finish the proof of theorem 1.

REFERENCES

[1] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, 2021, Art. no. 160, 2021.

[2] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–33, 2020.

[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. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[4] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, 2019, pp. 13–23.

[5] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[6] K. Bonawitz et al., "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, vol. 1, Apr. 2019, pp. 374–388.

[7] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7184–7193.

[8] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," 2021, *arXiv:2101.11203*.

[9] J. Bian and J. Xu, "Mobility improves the convergence of asynchronous federated learning," 2022, *arXiv:2206.04742*.

[10] S. Augenstein et al., "Mixed federated learning: Joint decentralized and centralized learning," 2022, *arXiv:2205.13655*.

[11] K. Yang, S. Chen, and C. Shen, "On the convergence of hybrid server-clients collaborative training," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 3, pp. 802–819, Mar. 2023.

[12] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

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

[14] Y. Liu et al., "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

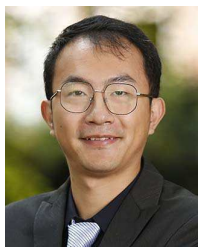
[15] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, 2020.

[16] S. U. Stich, "Local SGD converges fast and communicates little," 2018, *arXiv:1805.09767*.

[17] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication," 2019, *arXiv:1909.05350*.
[18] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 9709–9758, 2021.
[19] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," 2020, *arXiv:2012.13995*.
[20] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5693–5700.
[21] J. Zhang, C. De Sa, I. Mitliagkas, and C. Ré, "Parallel SGD: When does averaging help?" 2016, *arXiv:1606.07365*.
[22] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.
[23] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Int. Conf. Mach. Learn.,* PMLR, 2020, pp. 5132–5143.
[24] S. Wang and M. Ji, "A unified analysis of federated learning with arbitrary client participation," 2022, *arXiv:2205.13648*.
[25] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE),* Piscataway, NJ, USA: IEEE Press, 2022, pp. 965–978.
[26] Y. Li, H. Chen, W. Bao, Z. Xu, and D. Yuan, "Honest score client selection scheme: Preventing federated learning label flipping attacks in non-IID scenarios," 2023, *arXiv:2311.05826*.
[27] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Serv.*, 2012, pp. 21–28.
[28] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1529–1542, Sep. 2022.
[29] N. Huang, M. Dai, Y. Wu, T. Q. Quek, and X. Shen, "Wireless federated learning with hybrid local and centralized training: A latency minimization design," *IEEE J. Sel. Topics in Signal Process.*, vol. 17, no. 1, pp. 248–263, Jan. 2023.
[30] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," 2020, *arXiv:2006.12097*.
[31] P. Glasserman, *Monte Carlo Methods in Financial Engineering,* vol. 53. New York, USA: Springer, 2004.
[32] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.
[33] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *Proc. Int. Conf. Mach. Learn.,* PMLR, 2016, pp. 314–323.
[34] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
[35] D. Jhunjhunwala, P. Sharma, A. Nagarkatti, and G. Joshi, "Fedvarp: Tackling the variance due to partial client participation in federated learning," in *Proc. Uncertainty Artif. Intell.,* PMLR, 2022, pp. 906–916.
[36] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12052–12064.
[37] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," 2021, *arXiv:2111.04263*.
[38] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A federated learning framework with adaptivity to non-IID data," *IEEE Trans. Signal Process.*, vol. 69, pp. 6055–6070, 2021.
[39] P. Prakash, J. Ding, M. Wu, M. Shu, R. Yu, and M. Pan, "To talk or to work: Delay efficient federated learning over mobile edge devices," in *Proc. IEEE Global Commun. Conf. (GLOBECOM),* Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–6.
[40] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Int. Conf. Artif. Intell. Statist.,* PMLR, 2020, pp. 2021–2031.
[41] 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.
[42] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, "Adaptive quantization of model updates for communication-efficient federated learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP),* Piscataway, NJ, USA: IEEE Press, 2021, pp. 3110–3114.
[43] Y. Mao et al., "Communication-efficient federated learning with adaptive quantization," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 13, no. 4, pp. 1–26, 2022.
[44] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[45] 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.
[46] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," May 2012.
[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

Jieming Bian received the B.A. degree in economics from the University of Colorado Denver, in 2019, and the M.S. degree in operations research from Columbia University, in 2021. He is currently working toward the Ph.D. degree in electrical and computer engineering department with the University of Miami. His research interests include communication efficiency and client scheduling federated learning problems.

Lei Wang received the B.A. degree in electronic information engineering from the University of Electronic Science and Technology of China, in 2020, and the M.S. degree in electrical and computer engineering from the University of California, Los Angeles, in 2022. He is currently working toward the Ph.D. degree in electrical and computer engineering department, University of Miami. His research interests include heterogeneous data federated learning problems and quantum networks.



Kun Yang received the B.E. degree in electronic information science and technology department from Tsinghua University, in 2017, and the M.S. degree in electrical engineering department from Texas A&M University, in 2019. He is currently working toward the Ph.D. degree with the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia. His research interests include on reinforcement learning for wireless communication, federated learning, and prompt engineering.

Cong Shen (Senior Member, IEEE) received the B.E. and M.E. degrees from the Department of Electronic Engineering, Tsinghua University, China, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles. He is currently an Assistant Professor with Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia. His research interests include area of communications, wireless networks, and machine learning. He received the National Science Foundation CAREER Award in 2022, and the Best Paper Award in 2021 IEEE International Conference on Communications (ICC).

Jie Xu (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from UCLA, in 2015. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Miami. His research interests include mobile edge computing/intelligence, machine learning for networks, and network security. He received the NSF CAREER Award in 2021.