

Topology-Driven Synchronization Interval Optimization for Latency-Constrained Geo-Decentralized Federated Learning

QI CHEN¹, WEI YU², XINCHEN LYU¹, ZIMENG JIA², GUOSHUN NAN¹ (Member, IEEE),
AND QIMEI CUI¹ (Senior Member, IEEE)

¹National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing 100876, China

²Business Research Department, China Mobile Research Institute, Beijing 100031, China

CORRESPONDING AUTHOR: X. LYU (e-mail: lxinchen@bupt.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2900302; in part by the National Natural Science Foundation of China under Grant 62371059; in part by the Joint Funds for Regional Innovation and Development of the National Natural Science Foundation of China under Grant U21A20449; in part by the Fundamental Research Funds for the Central Universities under Grant 2242022k60006; and in part by the Beijing University of Posts and Telecommunications–China Mobile Research Institute Joint Innovation Center.

(Qi Chen and Wei Yu contributed equally to this work.)

ABSTRACT Geo-decentralized federated learning (FL) can empower fully distributed model training for future large-scale 6G networks. Without the centralized parameter server, the peer-to-peer model synchronization in geo-decentralized FL would incur excessive communication overhead. Some existing studies optimized synchronization interval for communication efficiency, but may not be applicable to latency-constrained geo-decentralized FL. This paper first proposes the synchronization interval optimization for latency-constrained geo-decentralized FL. The problem is formulated to maximize the model training accuracy within a time window under communication/computation constraints. We mathematically derive the convergence bound by jointly considering data heterogeneity, network topology and communication/computation resources. By minimizing the convergence bound, we optimize the synchronization interval based on the approximated system consistency metric. Extensive experiments on MNIST, Fashion-MNIST and CIFAR10 datasets validate the superiority of the proposed approach by achieving up to 30% higher accuracy than the state-of-the-art benchmarks.

INDEX TERMS Federated learning, edge intelligence, latency-constrained, communication efficiency.

I. INTRODUCTION

BY ENABLING distributed model training across multiple clients, federated learning (FL) is one of the key technologies for 6G communication networks [1], [2], [3], [4], [5], [6], [7]. There are various applications of federated learning for communication networks, including wireless resource optimization, network orchestration, and network security [2], [8], [9], [10], [11]. However, traditional FL typically requires the centralized (or semi-distributed) parameter server for model synchronization [12], [13], which may not be practical geo-decentralized and large-scale 6G networks. Geo-decentralized FL is a promising paradigm to empower the fully-distributed model training without the parameter server [14], [15], [16]. Its key feature is the scalability in general-topology learning networks, where the

edge servers only need to communicate with their one-hop neighbors for global convergence.

As compared to its centralized/semi-distributed counterpart, the model synchronization in geo-decentralized FL may be inefficient (i.e., resulting in increasing communication overhead and complicated convergence guarantee). Recent studies [17], [18], [19] designed the dynamic adjustment of synchronization intervals (i.e., the number of training epochs between two consecutive synchronization operations) for geo-decentralized FL. However, the existing approaches only focus on minimizing the overall communication overhead till model convergence, and may not be applicable to the latency-constrained FL setting [20]. Latency-constrained FL setting holds significant practical relevance for real-time learning applications and resource-limited edge devices [20], [21],

[22], [23], [24], [25], [26]. In this scenario, training must conclude within a designated time limit before recommencing. This setting, emphasizing training time constraints, diverges from the traditional emphasis on accuracy post-convergence, thus requiring different synchronization optimization design.

The objective of latency-constrained geo-decentralized FL (i.e., improving the training effectiveness within a time window) is distinctively different from the traditional training (for better convergence accuracy without deadline requirement). In the latency-constrained setting, there are only some works for the synchronization interval optimization in traditional FL [27], [28], but the design in geo-decentralized FL has yet to be studied in the literature.

This paper first studies the synchronization interval optimization for latency-constrained geo-decentralized FL. The problem is to maximize the model training efficiency (accuracy) of geo-decentralized FL within a time window under the communication/computation constraints. We mathematically derive the convergence bound within a time window by jointly considering data heterogeneity, network topology, and communication/computation resources. The derived convergence result reveals the relationship between the synchronization interval and the achieved loss function at the end of the time window. Then, we reformulate the problem and obtain the optimal synchronization interval by minimizing the upper bound of the loss function. To solve the problem, the system consistency metric (measuring the maximum model difference) is accurately approximated by extending the Dijkstra algorithm for the shortest path in graph theory.

The key contributions can be summarized as follows.

- We mathematically derive the convergence bound of geo-decentralized FL within a time window. The bound reveals the relationship of bounded loss function to data heterogeneity, network topology, and communication/computation consumption.
- We reformulate the problem to minimize the bounded loss function for adaptive synchronization optimization. We propose accurately approximating the system consistency metric (measuring the model difference) by extending the Dijkstra algorithm.
- Extensive experiments on MNIST [29], Fashion MNIST [30] (i.e., FMNIST for brevity) and CIFAR10 [31] datasets validate that the proposed approach can achieve up to 30% higher accuracy than the state-of-the-art benchmarks.

The rest of the article is organized as follows. Section II provides the literature review on recent progress on synchronization interval optimization in FL. Section III presents the system model and formulates the optimization problem for latency-constrained geo-decentralized FL. Section IV derives the convergence bound of geo-decentralized FL and analyzes its relationship to data heterogeneity and network topology. Section V illustrates the proposed synchronization

adjustment algorithm. Section VI analyzes the experimental results, followed by the conclusion in Section VII.

II. RELATED WORK

In its infancy stage, there are limited studies on synchronization interval optimization for geo-decentralized FL. For comprehensiveness, this section provides a literature review on the recent progress in both the topics of traditional FL and geo-decentralized FL.

A. TRADITIONAL FEDERATED LEARNING

Recently, latency-constrained (or resource-constrained) federated learning, in the context of a centralized parameter server, has emerged as a prominent area of interest [23], [24], [25], [26]. Various effective methodologies have been developed to address challenges associated with unreliable and resource-limited wireless networks [23], diverse heterogeneous devices [24], and the implementation of adaptive model pruning or partial aggregation to enhance communication and energy efficiency [25], [26]. In this paper, we consider the latency-constrained geo-decentralized federated learning, where the centralized parameter server is not available in large-scale 6G networks.

Various studies have proposed different approaches to optimize the synchronization interval of federated learning (i.e., the parameter-server architecture). 1) By analyzing the upper bound of convergence of loss function [27], the synchronization interval is dynamically adjusted to make the selected synchronization interval more conducive to convergence. 2) Optimize the synchronization interval based on heterogeneous information such as computing resources and storage capacity of different devices [28], [33]. 3) The synchronization interval is adjusted in lazy aggregation mode, i.e., only performing synchronization when the gradient update exceeds a predefined threshold (reducing unnecessary synchronization overhead) [14], [32]. The threshold can also be dynamically adjusted according to different data distributions at the local devices/workers [34].

Given the distinct learning procedure, the existing approaches designed for traditional FL [14], [27], [28], [32], [33], [34] cannot be applied in geo-decentralized FL.

B. GEO-DECENTRALIZED FEDERATED LEARNING

Decentralized Stochastic Gradient Descent (DSGD) [35] is the typical learning method for geo-decentralized FL. As compared to traditional FL, geo-decentralized FL can operate in a fully distributed manner. Without the centralized parameter server, geo-decentralized FL suffers from increasing communication overhead and complicated convergence guarantee. Communication efficiency via synchronization interval optimization is one of the research hotspots for geo-decentralized FL [17], [18], [19].

The synchronization interval optimization for geo-decentralized FL was initially studied in [17], [18], where the authors highlighted the need for adaptively adjusting the synchronization interval as the training epoch evolves. However,

TABLE 1. Comparison of different methods of adjusting synchronization interval.

	Approach	Article	Latency- constrained	Highlight
Traditional FL	Set threshold condition	[14], [32]	✗	Model synchronization occurs when the threshold is reached
	Optimization	[33]	✗	The computing power and storage capacity of the device are used as a basis for adjusting the frequency of local updates in traditional FL.
	Optimization	[27], [28]	✓	Adjust the synchronization interval in traditional FL by optimizing the convergence upper bound .
Geo-decentralized FL	Heuristic search	[17]	✗	Improve communication efficiency by increasing synchronization intervals and compressing traffic
	Heuristic search	[18]	✗	Point out the changing trend of synchronization interval during training
	Optimization	[19]	✗	Adaptively adjust the synchronization interval and overcome heterogeneity by changing communication topology
	Optimization	Our work	✓	Adaptively adjust the synchronization interval driven by topology and other factors in latency-constrained geo-decentralized FL.

the synchronization intervals were adjusted in a heuristic manner with meticulously-designed control (cannot scale to the general learning settings) [17], [18]. To further design a general method for synchronization interval optimization, the authors in [19] formulated the problem of optimizing synchronization intervals and topology for geo-decentralized FL. In particular, without considering the physical-world link connections, the synchronization/communication topology (i.e., whether performing synchronization between any two edge servers) is dynamically established to reduce data heterogeneity.

However, these approaches in [17], [18], [19] were designed for the learning without deadline, and not applicable for the latency-constrained setting. For the latency-constrained setting, there are only some studies for the synchronization optimization in traditional FL [27], [28]. This paper is the first to study the synchronization optimization for latency-constrained geo-decentralized FL. The related works are summarized in Table 1.

III. SYSTEM MODEL AND PROBLEM STATEMENT

Fig. 1 shows an latency-constrained geo-decentralized FL system, where N edge servers can collaborate to train a global model by only communicating with their one-hop neighbors. In the latency-constrained setting, the servers periodically train the learning models within a time window with duration Z . Let $G = (\mathbf{N}, \mathbf{E})$ denote the topology of the geo-decentralized FL network. Here, \mathbf{N} and \mathbf{E} are the N edge servers and the inter-node links between the servers, respectively.

The key problem is to balance the communication (synchronizing training results) and computation (model training), which can also be called adaptive. Spending

too much time for model synchronization (e.g., at each epoch) would exhaust the model training time and result in the model non-convergence dilemma. As a result, the synchronization interval τ must be meticulously designed to achieve the best learning accuracy, especially in the latency-constrained setting with stringent time window dead line. (as shown in the right part of Fig. 1). Table 2 summarizes the notations used in this paper.

A. GEO-DECENTRALIZED FEDERATED LEARNING MODEL

In machine learning, each sample consists of two parts, i.e., (x, y) , where x and y are the input and the ground-truth label, respectively [36]. Each edge server i has its local model training dataset (e.g., arriving data from the sensory devices within the last time window), denoted by \mathcal{D}_i . Let $|\mathcal{D}_i|$ be the size of dataset \mathcal{D}_i . The local loss function of edge server i based on its local model parameter w_i , i.e., $F_i(w_i)$, is given by

$$F_i(w_i) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y) \in \mathcal{D}_i} h(w_i, x, y) \quad (1)$$

where $h(w_i, x, y)$ is per-sample loss function with respect to (x, y) [37]. The objective of FL is to minimize the global loss function of the system, as given by

$$\min_{w \in \mathbb{R}^d} F(w) := \frac{1}{N} \sum_{i=1}^N F_i(w_i), \quad (2)$$

where $F(w)$ is used to substitute $F(w_1, w_2, \dots, w_N)$ for brevity.

DSGD [35] is typically adopted to solve the FL problem (2) in a fully distributed manner. It operates

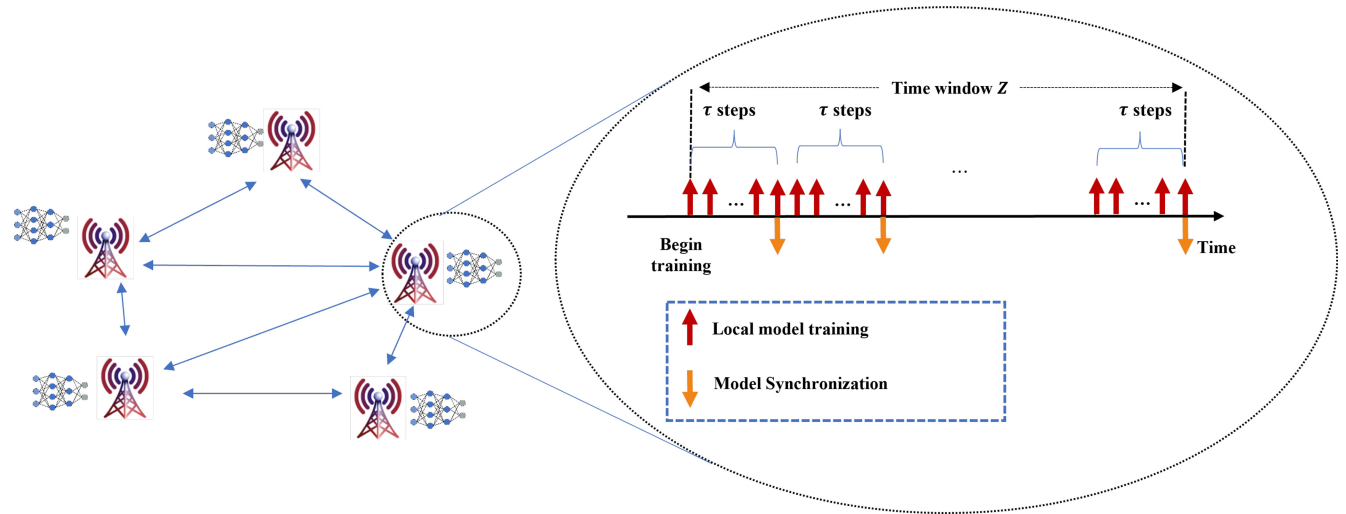


FIGURE 1. An latency-constrained geo-decentralized FL system, where N edge servers can collaborate to train a global model within a predefined time window with duration Z . The synchronization interval must be meticulously optimized based on the data heterogeneity, network topology, as well as the time window duration.

TABLE 2. Summary of main notations.

Notation	Description
t	Local model training iteration index
$w_i(t)$	Local model parameter at edge server i in iteration t
$\bar{w}(t)$	The average of all edge servers' model parameter in iteration t
$F(w)$	Global loss function
$F_i(w)$	Local loss function for edge server i
$s(t)$	Model parameter of centralized learning in iteration t
w^*	True optimal model parameter that minimizes $F(w)$
t^*	The final output iteration of each edge server
A	The synchronization matrix
α	Algebraic connectivity of the graph
γ	Gradient descent step size
τ	Number of local model training steps between two global aggregations (synchronization interval)
T	Total number of local model training steps at each edge server
Z	Time window
L	Time consumption in per-iteration training
Y	Time consumption in one time model synchronization
ρ	Lipschitz parameter
β	Smoothness parameter
φ	Gradient divergence
C	System consistency metric
φ	Gradient Divergence metric
$\bar{\tau}^*$	Optimal τ obtained by minimizing $H(\tau)$
τ^*	The final synchronization interval selected τ
ζ	Degree of non-iid in the Dirichlet distribution

by iteratively performing local model training and model synchronization, as shown in the following.

- 1) *Local model training.* At iteration t , each edge server i needs to randomly choose a local datum ξ_i which is employed to select a mini-batch dataset from \mathcal{D}_i and

uses its current local variable $w_i(t-1)$ to evaluate the stochastic gradient $\nabla f_i(w_i(t-1), \xi_i)$. The local learning model can be updated according to [35]

$$w_i\left(t - \frac{1}{2}\right) = w_i(t-1) - \gamma \nabla f_i(w_i(t-1), \xi_i). \quad (3)$$

- 2) *Model synchronization.* After finishing the local model training, each edge server i exchanges the updated model $w_i(t - \frac{1}{2})$ with its neighboring edge servers. Upon receiving the model weights from one-hop neighbors, the model of edge server i can be updated by

$$w_i(t) = \sum_{j \in \Omega_i} a_{ij} w_j\left(t - \frac{1}{2}\right) \quad (4)$$

where a_{ij} represents the model synchronization weight of edge server j at edge server i (which can be set according to (6) in the following) and Ω_i is the set of one-hop neighbors of edge server i . Let $A = [a_{ij}]$ be the synchronization weight matrix. The model synchronization of all the edge servers can be written as

$$[w(t)] = A \left[w\left(t - \frac{1}{2}\right) \right], \quad (5)$$

where $[w]$ represents the vector of all the local models formed by the vertical arrangement.

Synchronization Matrix Generation. For a connected network, the synchronization matrix A can be set according to Metropolis-Hastings algorithm [38]. Let d_i be the degree (i.e., the number of connected links) of edge server i . The synchronization weight a_{ij} in Eq. (4) can be given by

$$a_{ij} = \begin{cases} \frac{1}{\max\{d_i, d_j\}}, & \text{if } j \in \Omega_i \\ 0, & \text{if } j \notin \Omega_i \\ 1 - \sum_{j \in \Omega_i} a_{ij}. & \text{if } i = j \end{cases} \quad (6)$$

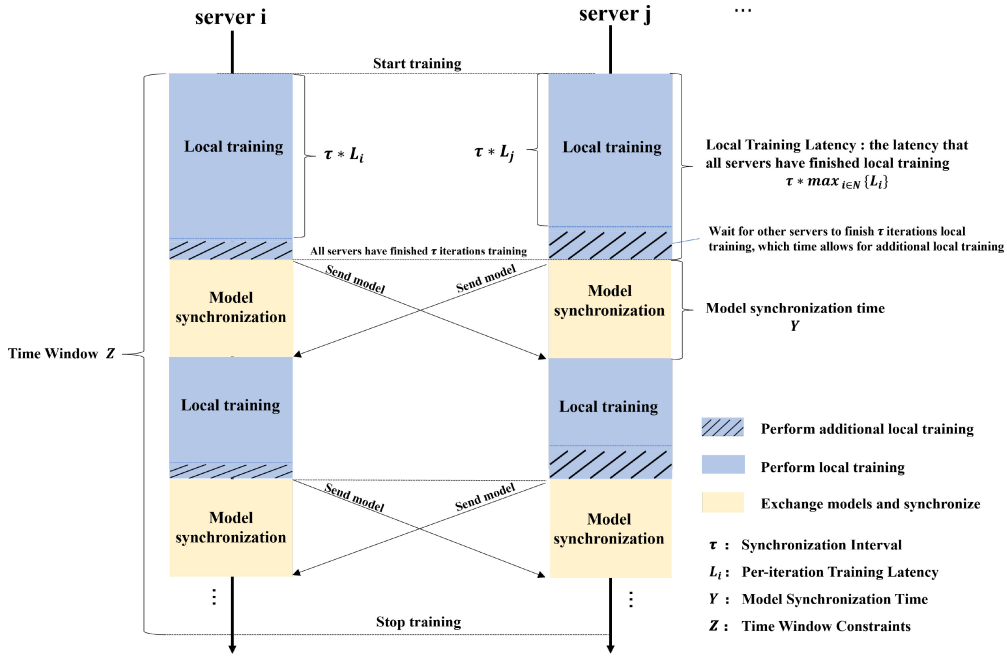


FIGURE 2. An illustrative example of model training and synchronization between any two adjacent servers i and j with $(i, j) \in E$.

Here, Eq. (6) can generate a doubly stochastic synchronization matrix A , and can guarantee the network convergence (as will be shown in Section IV).

Synchronization Interval. As stated above, the model synchronization in Eq. (5) does not need to be conducted after each training iteration t , but at a timescale of a predefined synchronization interval τ . In other words, the model synchronization only performs every τ training iteration. In this paper, we aim to optimize the synchronization interval τ for the latency-constrained geo-decentralized FL system.

B. SYNCHRONIZATION MODEL WITH TIME WINDOW CONSTRAINT

Fig. 2 illustrates the process of local training and model synchronization for two servers. Let T denote the total number of local model training iterations within the predefined time window duration Z . The maximum number of training iterations T depends on 1) the per-iteration training latency, denoted by L_i , 2) model synchronization time, denoted by Y and 3) the synchronization interval τ . The definition and analysis of the aforementioned notations are as follows.

1) *Per-iteration Training Latency.* Consider the heterogeneity of different edge servers. The latency of per-iteration training of edge server i can be denoted by L_i . The training duration within one synchronization interval would be $\max_{i \in N} \{\tau * L_i\} = \tau L$, where $L = \max_{i \in N} L_i$ is the maximum per-iteration training delay of the straggler (depending on the local computation capability). Assume that L_i does not change across different synchronization intervals. Summing up the training delay of different intervals results in the estimation of the overall training delay of T iterations, i.e., TL .

2) *Model Synchronization Time.* This metric represents the time consumed by all adjacent servers to complete the process of model sending and receiving and calculating the weighted average value. Let $B_{i,j}$ denote the capacity of link (i, j) [39], and M be the size of transmitted updated model (depending on the number of neurons of the learning model and the adopted quantization method). Here, the link capacity $B_{i,j}$ indicates the available communication resources of link (i, j) . We ignore the time it takes the server to calculate the mean, so that the model synchronization time Y is also the maximum of parameter transmission time over each link (i, j) , i.e.,

$$Y = \max_{(i,j) \in E} \left\{ \frac{M}{B_{i,j}} \right\}. \quad (7)$$

3) *Time Window Constraints.* Given the training iteration T and synchronization interval τ , the number of model synchronization K can be given by $K = \lfloor \frac{T}{\tau} \rfloor$. In the latency-constrained setting, the maximum training iteration T is confined by the time window duration Z , satisfying

$$KY + TL = \left\lfloor \frac{T}{\tau} \right\rfloor Y + TL \leq Z. \quad (8)$$

In other words, the overall time for local model training and model synchronization cannot exceed the time window duration. The rounding operation in Eq. (8) can be omitted without loss of generality [27]. The maximum training iteration T can be written as

$$T^{\max} = \frac{Z}{L + \frac{Y}{\tau}}. \quad (9)$$

4) *Synchronization Interval*. This metric represents the number of local training performed between each model synchronization.

C. PROBLEM FORMULATION

In the latency-constrained setting, the objective is to minimize the loss function (i.e., the best training accuracy) at the end of the time window, i.e., $F(w(T^{\max}))$. The problem of interest is to optimize the synchronization interval τ to minimize the loss function, as given by

$$\begin{aligned} \min_{\tau} & F(w(T^{\max})) \\ \text{s.t.} & \quad (9) \\ & 0 < \tau \leq T^{\max}, \quad \tau \in \mathbb{N}^+. \end{aligned} \quad (10)$$

The synchronization period in the problem (10) must be a positive integer and cannot exceed the maximum number of iterations. We note that it is challenging to solve problem (10) due to the lack of the relationship between the synchronization interval τ and the final loss function $F(w(T^{\max}))$. Given the complexity of learning models, the relationship is complicated and non-trivial to be obtained. In the following, we start by bounding the loss function $F(w(T^{\max}))$ and establishing its relationship to τ in Section IV. Based on the analyzing results, we can solve problem (10) to minimize the bounded loss function, hence optimizing the synchronization interval, in Section V.

IV. CONVERGENCE ANALYSIS

In this section, we establish a quantitative relationship between the upper bound of the loss function and the synchronization interval. Various factors, including data heterogeneity, communication topology, and computing resources, are revealed in the analysis. The upper bound analysis provides the mathematical foundation for optimizing the synchronization interval in Section V.

A. ASSUMPTIONS AND METRICS

To facilitate the proof, we first introduce the general assumptions of loss functions, and present typical metrics for measuring the connectivity of a specific graph.

1) *Loss Function Assumptions*. We make the following assumptions for local loss functions $\{F_i(w)\}_{i=1}^N$. We use $\|\cdot\|$ to denote the \mathcal{L}_2 norm.

Assumption 1 (Convexity):

$$F_i(w) - F_i(w') \leq \nabla F_i(w)^T (w - w') \text{ for any } w, w'.$$

Assumption 2 (ρ -Lipschitz):

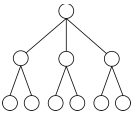
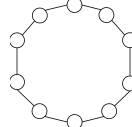
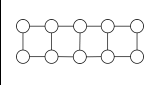
$$\|F_i(w) - F_i(w')\| \leq \rho \|w - w'\| \text{ for any } w, w'.$$

Assumption 3 (β -Smooth):

$$\|\nabla F_i(w) - \nabla F_i(w')\| \leq \beta \|w - w'\| \text{ for any } w, w'.$$

From [40], we know that when local dataset \mathcal{D}_i is uniformly drawn (without replacement) from the overall data, it comes to $E(f_i(w), \xi_i) = F_i(w)$, $E(\nabla f_i(w, \xi_i)) = \nabla F_i(w)$, for any $1 \leq i \leq N$. Also, stochastic gradient descent can be seen as an approximation to gradient descent [27], [40]. We have $f_i(w, \xi_i) = F_i(w)$, $\nabla f_i(w, \xi_i) = \nabla F_i(w)$.

TABLE 3. Three different communication topologies.

Topology			
Name	tree	ring	grid
Algebraic Connectivity	$\alpha = 0.93$	$\alpha = 0.87$	$\alpha = 0.80$

2) *Topology Metrics*. For convergence, the synchronization matrix A in geo-decentralized FL system should be a symmetric and doubly stochastic matrix [41], i.e.,

$$A = A^T, A\mathbf{1}_N = \mathbf{1}_N. \quad (11)$$

The synchronization matrix in (6) meets these conditions. There are N eigenvalues of the matrix A . Let $\lambda_i(A)$ denote the i -th largest eigenvalue of A . According to Perron–Frobenius theory, the absolute eigenvalues cannot exceed 1, i.e., $1 = \lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_N(A) \geq -1$. In particular, the second largest eigenvalue modulus of A can be given by [42]

$$\alpha = \max(|\lambda_2(A)|, |\lambda_N(A)|). \quad (12)$$

According to [42], α satisfies the following properties:

- 1) $0 \leq \alpha < 1$,
- 2) $\alpha = \|A - \mathbf{1}_N \mathbf{1}_N^T\|$,
- 3) $\alpha^x = \|A^x - \mathbf{1}_N \mathbf{1}_N^T\|$.

In graph theory, the second largest eigenvalue modulus of the adjacency matrix, i.e., α is called the algebraic connectivity of the graph. In general, if a graph has a small algebraic connectivity value, the graph has a high connection density. Conversely, if the algebraic connectivity value is large, the graph is sparse.

Table 3 shows the algebraic connectivity of three typical topologies. We can see that the connection densities α of the tree, ring and grid topologies 0.93, 0.87, and 0.80, respectively, i.e., α decreases as the topology becomes denser. We will show in Theorem 2 that the topology metric α also influences the convergence bound.

3) *Systematic Heterogeneity Metrics*. We proceed to introduce the metrics for measuring the differences of local models in the system, which relate to the data heterogeneity at different edge servers.

Definition 1 (System Consistency [43], [44], [45]): The difference between the local model of edge server i and the averaged network model $\bar{w}(t) = \frac{1}{N} \sum_{i=1}^N w_i(t)$, can be given by

$$C_i(t) = \|w_i(t) - \bar{w}(t)\|. \quad (13)$$

The overall system consistency metric is $C(t) = \frac{1}{N} \sum_{i=1}^N C_i(t)$.

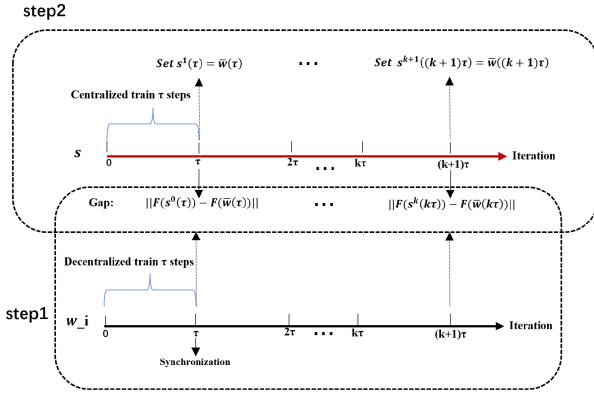


FIGURE 3. Outline of the convergence analysis process.

Definition 2 (Gradient Divergence [46]): For any w , the local gradient divergence of edge server i , denoted by φ_i , can be given by

$$\left\| \nabla F_i(w) - \frac{1}{N} \sum_{j=1}^N \nabla F_j(w) \right\| \leq \varphi_i. \quad (14)$$

The global gradient divergence is the average of local divergence, i.e., $\varphi = \frac{1}{N} \sum_{i=1}^N \varphi_i$.

B. CONVERGENCE ANALYSIS

In this section, we will mathematically prove the upper bound of the loss function $F(w)$. Let $\bar{w}(t) = \frac{1}{N} \sum_{i=1}^N w_i(t)$ denote the averaged model parameter at local model training iteration t . Note that $\bar{w}(t)$ is not available in geo-decentralized FL without centralized synchronization. Nevertheless, $\bar{w}(t)$ can help find the performance gap to the centralized counterpart (whose convergence bound has been widely known [47]), hence establishing the convergence bound.

To capture the centralized counterpart, we introduce an auxiliary variable $s^k(t)$. The auxiliary variable characterizes the model parameter for centralized training between the k -th and $(k+1)$ -th model synchronizations, i.e., $t \in [k\tau, (k+1)\tau]$. The variable is updated via gradient descent, i.e.,

$$s^k(t+1) = s^k(t) - \gamma \nabla F(s^k(t)). \quad (15)$$

Here, we assume the availability of the global loss function in Eq. (15) to approximate the centralized ML. Fig. 3 shows an illustrative example for the evolution of s and the outline of using s to build up the convergence proof. In particular, we set $s^k(k\tau) = \bar{w}(k\tau)$ after the k -th synchronization. Note that $s^k(k\tau)$ and $s^{k-1}(k\tau)$ are not equal, because centralized training converges faster than decentralized training. The value of $\bar{w}(k\tau)$ will not change during model synchronization. As in the notation of auxiliary variable s , $[\cdot]^k$ denote the variable $[\cdot]$ is between the k -th and $(k+1)$ -th model synchronizations. As shown in Fig. 3, there are two steps for the convergence analysis of latency-constrained geo-decentralized FL.

Step 1: Bound the gap to the auxiliary variable s (Theorem 1). We first find the upper bound of the

gap between $F(\bar{w}(k\tau))$ and $F(s^{k-1}(k\tau))$ for any k : $\|F(s^{k-1}(k\tau)) - F(\bar{w}(k\tau))\|$.

Step 2: Integrate the gap to centralized ML. (Theorem 2). This is to establish the final convergence results of the proposed approach.

Theorem 1: The gap (in terms of loss functions) between the averaged network model \bar{w} and the auxiliary one s can be bounded. For any $k > 0$ and $\tau > 0$, we have

$$\|F(\bar{w}(k\tau)) - F(s^{k-1}(k\tau))\| \leq \rho\Theta(\tau) \quad (16)$$

where

$$\Theta(\tau) = \left(C^{k-1}((k-1)\tau) + \frac{\varphi}{\beta} \right) ((\gamma\beta + 1)^\tau - 1) - \varphi\gamma\tau. \quad (17)$$

Proof: Please refer to Appendix-A. ■

Based on Theorem 1, we can proceed to establish the convergence bound of geo-decentralized FL.

Theorem 2: When $\exists \varepsilon > 0$ satisfied $F(s^{k-1}(k\tau)) - F(w^*) \geq \varepsilon$ for all k and $F(\bar{w}(T)) - F(w^*) \geq \varepsilon$, we can choose a suitable learning rate $\gamma \leq \frac{1}{\beta}$ to get the convergence upper bound of Algorithm 2 after T iterations, i.e.,

$$F(\bar{w}(T)) - F(w^*) \leq \frac{1}{T \left(\frac{\gamma}{2\varpi} - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \right)}, \quad (18)$$

where $\varpi = \max_k \|\bar{w}(k\tau) - w^*\|$. When $\frac{\gamma}{2\varpi} - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} > 0$, the value of $F(\bar{w}(T)) - F(w^*)$ diminishes (decreases to 0) as iteration T increases, i.e., the convergence property of geo-decentralized FL. We also have

$$F(w^f) - F(w^*) \leq \frac{\varpi}{\gamma T} + \rho\Theta(\tau) + \sqrt{\frac{\varpi^2}{\gamma^2 T^2} + \frac{2\rho\Theta(\tau)\varpi}{\gamma\tau}}. \quad (19)$$

Proof: Please refer to Appendix-C. ■

C. ANALYSIS OF CONVERGENCE BOUND

Theorem 2 shows the convergence bound of geo-decentralized FL given synchronization interval τ . In the following, we analyze the relationship between the bounded loss function and various factors in the latency-constrained geo-decentralized system.

Lemma 1: For any k , we have:

$$C^k(k\tau) \leq P(\alpha) := \frac{\alpha\gamma\tau N\rho}{1-\alpha} + \alpha\| [w(0)] \| \quad (20)$$

where $[w(0)]$ is the matrix of initial model parameter permutations for all edge servers.

Proof: Please refer to Appendix-B. ■

In Lemma 1 we find the upper bound of $C^k(k\tau)$ for any k . By substituting Eq. (20) into Eq. (19), we find the upper bound of $F(\bar{w}(T)) - F(w^*)$:

$$F(w^f) - F(w^*) \leq \frac{\varpi}{\gamma T} + \rho \left(P(\alpha) + \frac{\varphi}{\beta} \right) M - \varphi \gamma \tau + \sqrt{\frac{\varpi^2}{\gamma^2 T^2} + \frac{2\rho \left(P(\alpha) + \frac{\varphi}{\beta} \right) M - \varphi \gamma \tau \varpi}{\gamma \tau}} \quad (21)$$

$$(M = (\gamma\beta + 1)^\tau - 1)$$

T is related to the communication/computing resources

φ is related to the data heterogeneity

$P(\alpha)$ is related to the communication topology

We can analyze the relationship between the convergence upper bound and various factors including communication/computing resources, data heterogeneity, and communication topology.

1) *Relationship to Topology Metric α* . In Eq. (21), $P(\alpha)$ is the upper bound of system consistency metric $C^k(k\tau)$. In other words, geo-decentralized FL can achieve better convergence as α decreases (see $P(\alpha)$). As stated in Section IV-A, a small value of α indicate a graph with dense connections.

2) *Relationship to Data Heterogeneity (measured by gradient divergence φ)*. φ in Eq. (21) is primarily associated with the heterogeneity of data distribution. Large values of φ indicates increasingly heterogeneous data distribution, also resulting in longer convergence time.

3) *Relationship to Communication and Computing Resources (per-iteration training time L and model synchronization time Y)*. From Eq. (9), the maximum number of training iterations T^{max} is determined by model synchronization time Y and maximum per-iteration training latency L . T^{max} determines the convergence in Eq. (21).

V. DESIGN OF ADAPTIVE SYNCHRONIZATION INTERVAL OPTIMIZATION FOR LATENCY-CONSTRAINED GEO-DECENTRALIZED FL

In this section, we solve problem (10) to optimize the synchronization interval for latency-constrained geo-decentralized FL. Due to the intractability of the accurate loss function, we aim to minimize the convergence bound, i.e., the right-hand-side (RHS) of Eq. (19). By substituting the expression of T^{max} in Eq. (9) into the bound, problem (10) can be reformulated as

$$\min_{\tau} H(\tau) = \frac{\varpi \frac{L\tau+Y}{Z\tau}}{\gamma} + \rho\Theta(\tau)$$

$$+ \sqrt{\left(\frac{\varpi \frac{L\tau+Y}{Z\tau}}{\gamma}\right)^2 + \frac{2\rho\varpi\Theta(\tau)}{\gamma\tau}} \quad (22)$$

$$\text{s.t. } 0 < \tau \leq T^{max}, \quad \tau \in \mathbb{N}^+.$$

Let τ^* be the optimal synchronization interval, as given by

$$\tau^* = \arg \min_{\tau \in \{1,2,3,\dots\}} H(\tau). \quad (23)$$

However, $H(\tau)$ is still hard-to-solve, since some system parameters (e.g., the system consistency $C^{k-1}((k-1)\tau)$) are not available in geo-decentralized FL. In the following, we first design the estimation method of system parameters and then present the proposed adaptive interval control algorithm.

A. ESTIMATION OF SYSTEM CONSISTENCY

To estimate the system consistency metric $C^k(k\tau)$ for any k , we need to calculate the difference between the model parameter of each edge server $w_i^k(k\tau)$ and the average of the model parameter of all edge servers $\bar{w}^k(k\tau)$. However, obtaining the average value of all edge server model parameters is impractical in a decentralized system. Otherwise, the edge server collecting all the model information can directly perform the FedAvg algorithm.

In the following, we refer to the triangular inequality to establish an efficient estimation method based on the shortest path in graph theory. We have reformulate the expression of $C^k(k\tau)$, i.e.,

$$\begin{aligned} C^k(k\tau) &= \frac{1}{N} \sum_{i=1}^N \|w_i^k(k\tau) - \bar{w}^k(k\tau)\| \\ &= \frac{1}{N} \sum_{i=1}^N \|w_i^k(k\tau) - \bar{w}^{k-1}(k\tau)\| \\ &\quad \text{(From Lemma 2 in Appendix-A)} \\ &= \frac{1}{N} \sum_{i=1}^N \left\| w_i^k(k\tau) - \frac{1}{N} \sum_{j=1}^N w_j^{k-1}(k\tau) \right\| \\ &= \frac{1}{N} \sum_{i=1}^N \left\| \frac{1}{N} \sum_{j=1}^N (w_i^k(k\tau) - w_j^{k-1}(k\tau)) \right\| \\ &\leq \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \frac{1}{N} \|w_i^k(k\tau) - w_j^{k-1}(k\tau)\|. \quad (24) \end{aligned}$$

Note that the model gap between two adjacent edge servers i and j , denoted by $c_{(i,j)}$, can be obtained at the servers according to $\|w_i^k(k\tau) - w_j^{k-1}(k\tau)\|$. Consider $c_{(i,j)}$ as the weight over edge (i,j) . The calculation of the weight difference of non-adjacent nodes becomes the shortest-path problem in graph theory. Let $P = (s, t, 1), (s, t, 2), \dots, (s, t, m)$ be the shortest path of two servers s and t of m hops. The estimation can be given by

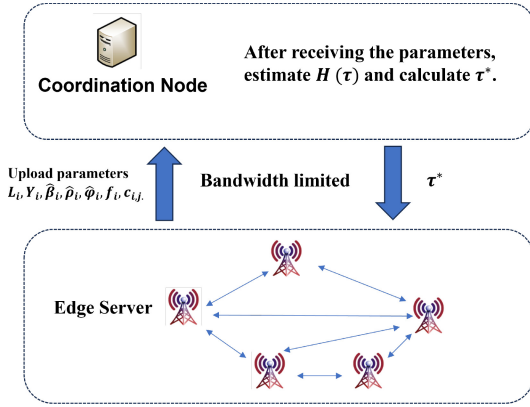


FIGURE 4. Interactions between the coordination node and edge servers.

$$\begin{aligned}
 & \left\| w_s^k(k\tau) - w_t^k(k\tau) \right\| \\
 & \leq \left\| w_s^k(k\tau) - w_{s,t,1}^{k-1}(k\tau) \right. \\
 & \quad \left. + w_{s,t,1}^{k-1}(k\tau), \dots - w_{s,t,m}^{k-1}(k\tau) \right. \\
 & \quad \left. + w_{s,t,m}^{k-1}(k\tau) - w_t^k(k\tau) \right\| \\
 & \leq \left\| w_s^k(k\tau) - w_{s,t,1}^{k-1}(k\tau) \right\| + \dots \\
 & \quad + \left\| w_{s,t,m}^{k-1}(k\tau) - w_t^k(k\tau) \right\|. \quad (25)
 \end{aligned}$$

The estimation of the model difference of any two edge servers in Eq. (25) directly follows the Dijkstra algorithm. Given the model difference, we can find the estimated system consistency $C^k(k\tau)$ based on Eq. (24).

B. ESTIMATION OF OTHER PARAMETERS

The calculation of $H(\tau)$ also needs the estimation of other parameters, including the gradient divergence φ , the smoothness parameter β , and the Lipschitz parameter ρ . As shown in Fig. 4, these parameters can be approximated by locally evaluating the corresponding local parameters at each edge server and being averaged at the coordination node. The local approximation of the parameters at each server i is shown in the following.

1) *Gradient Divergence*. Each edge server uses the local gradient divergence, denoted by $\hat{\varphi}_i$, to approximate the global divergence φ . $\hat{\varphi}_i$ can be locally evaluated at the edge servers without any additional signaling, i.e.,

$$\hat{\varphi}_i = \left\| \nabla f_i(w_i) - \sum_{j \in \Omega_i} a_{ij} \nabla f_j(w_j) \right\|. \quad (26)$$

2) *Smoothness parameter*. The local smoothness parameter, denoted by β_i , can be computed as

$$\hat{\beta}_i = \frac{\|\nabla f(w_i(t_1)) - \nabla f(w_i(t_2))\|}{\|w_i(t_1) - w_i(t_2)\|}, \quad (27)$$

where t_1 and t_2 are two adjacent synchronization time.

3) *Lipschitz parameter*. Similarly, the local Lipschitz parameter, denoted by ρ_i , can be given by

$$\hat{\rho}_i = \frac{\|f(w_i(t_1)) - f(w_i(t_2))\|}{\|w_i(t_1) - w_i(t_2)\|}. \quad (28)$$

C. ADAPTIVE SYNCHRONIZATION INTERVAL CONTROL ALGORITHM

This section details the algorithm of finding the optimal intervals. As specified by (22), the metric of optimal intervals is to minimize the upper bound of the loss function $H(\tau)$ according to the convergence result. As illustrated in Section V-A/V-B, the solver (i.e., coordination node) can approximate the parameters for calculating $H(\tau)$ according to Eqs. (24)–(28). Then, the approximated expression of $H(\tau)$, denoted by $\hat{H}(\tau)$, is given by

$$\begin{aligned}
 \hat{H}(\tau) &= \frac{\varpi \frac{L\tau+Y}{Z\tau}}{\gamma} + \hat{\rho} \hat{\Theta}(\tau) \\
 &+ \sqrt{\left(\frac{\varpi \frac{L\tau+Y}{Z\tau}}{\gamma}\right)^2 + \frac{2\hat{\rho}\varpi \hat{\Theta}(\tau)}{\gamma\tau}} \quad (29a)
 \end{aligned}$$

where

$$\begin{aligned}
 \hat{\Theta}(\tau) &= \left(\hat{C}^{k-1}((k-1)\tau) + \frac{\hat{\varphi}}{\beta} \right) \left((\gamma\hat{\beta} + 1)^\tau - 1 \right) \\
 &- \hat{\varphi}\gamma\tau. \quad (29b)
 \end{aligned}$$

By applying the approximation of (29), problem (22) becomes an integer programming problem, which is hard to find a closed-form expression. To this end, we adopt the Particle Swarm Optimization (PSO) algorithm [48] to find the optimal synchronization interval within the possible values of 1 to τ^{\max} , where $\tau^{\max} = \lfloor \frac{Z-Y}{L} \rfloor$, with the expression of τ^{\max} derived from solving the constraints in problem (22).

The process of PSO to find τ^* operates in an iterative manner. First, we randomly initialize 10 candidate values of τ_i , $i = \{1, 2, \dots, 10\}$, i.e., particles. Let p_{id} be the best location (i.e., synchronization interval) of particle τ_i with the minimum objective $\hat{H}(\tau)$ in Eq. (29), and p_{gd} be the global optimal value for all the particles.

During each iteration, we update the particles according to

$$\begin{aligned}
 v_i^{k+1} &= w * v_i^k + c_1 * rand() * (p_{id} - \tau_i^k) \\
 &+ c_2 * rand() * (p_{gd} - \tau_i^k) \\
 \tau_i^{k+1} &= \tau_i^k + v_i^{k+1} \quad (30)
 \end{aligned}$$

where τ_i^k and v_i^k denote the location and velocity of particle i at the k -th iteration. $rand()$ is the random number generation function. The details can be found in [48]. To account for the integral feature, we also enforce the quantization operation to v_i^k . The iteration process terminates when p_{gd} remains the same for 10 consecutive iterations.

Fig. 4 shows the basic implementation of the proposed approach. In particular, the edge servers would report these locally evaluated local parameters to the coordination node. The coordination node can take the average to approximate $H(\tau)$ and adaptively control the synchronization interval. Note that there are only limited parameters (a few bits to represent the real numbers, e.g., 32-bit floating numbers),

which are required to be transmitted and can be implemented in the bandwidth-critical decentralized setting.

We also extend the algorithm to optimize the synchronization interval in an latency-constrained setting. In particular, instead of using the fixed τ , we readjust τ at each time of model synchronization based on current system parameters and the remaining time window. It is also worth noting that the communication interval needs to be recalculated based on the latest estimated parameters after each model synchronization. This ensures the algorithm's adaptability to various changes in the system. The detailed steps at the coordination node and edge servers are summarized in Algorithms 1 and 2.

VI. EXPERIMENTAL AND RESULT

This section evaluates the performance of our proposed algorithm in different cases against the benchmark algorithms. In the following, we first introduce the experimental settings and then discuss the results.

A. EXPERIMENT SETTINGS

1) Datasets and Models. The experiments were conducted using the open-source datasets MNIST [29], Fashion MNIST (i.e., FMNIST for brevity) [30] and CIFAR10 [31]. We trained the MNIST and FMNIST dataset with a two-layer convolutional layer CNN [29] network and the CIFAR10 dataset with a ResNet [49] network.

2) Parameters and Environment. We configured 10 edge servers in different topologies (including ring, tree, grid, etc.). The learning rate is 0.1. According to [27], the time window for MNIST and FMNIST was 15 seconds and the time window for CIFAR10 was 40 seconds. Communication bandwidth of all links is 1 MB/s. Computation time is determined through actual measurements and communication time is estimated by Eq. (7).

The value of $\varpi := \max_k \|\bar{w}(k\tau) - w^*\|$ cannot be known before training to the convergence, and must be estimated for the synchronization adjustment. We note that the model gap to the optimal values takes the maximum value at slot $t = 0$. We approximate $\varpi := \max_k \|\bar{w}(k\tau) - w^*\|$ as the value of $\|2\bar{w}(0)\|$ is close to ϖ . We set the correction factor λ 1.8 for MNIST, 2 for FMNIST and 9 for CIFAR10, which is designed to compensate the estimation errors. Please refer to Appendix VII-D for the explanations on correction factor selection.

3) Data Distribution. The dataset is distributed across edge servers using the Dirichlet distribution [50] to simulate datasets that are non-independent and non-uniformly distributed (Non-IID). The dispersion values ζ are set to 0.1, 0.2, and 0.3. A higher ζ value implies increasingly independent and uniform distribution.

4) Baseline. For performance evaluation, we also conduct the existing approaches of DSGD [35], DFedAvgM [17], and DFedAvgM-advance to serve as the benchmarks.

- DSGD [35], which is the most widely adopted approach to solve the geo-decentralized FL problem. The edge

Algorithm 1 Procedure at the Coordination Node

```

1: Input: Time window  $Z$ , maximum  $\tau$  value  $\tau_{\max}$ ,
   correction factor  $\lambda$ ;
2: Output:  $w^f$ 
3: Initialize  $\tau^* = 1$ ,  $t^* = 0$ ,  $t = 0$ ,  $T = 0$  and  $w(0)$ ;
4: repeat
5:    $t_0 = t$ ;
6:    $t = t + \tau^*$ ;
7:   if  $t_0 > 0$  then
8:     Receive  $L_i, Y_i, \hat{\varphi}_i, \hat{\rho}_i, \hat{\beta}_i, f_i(w_i(t_0))$  and all  $c_{(i,j)}$ 
       from each edge server  $i$ ;
9:     Compute  $F(w(t_0))$ ;
10:    if  $t == 1$  then
11:      set  $F^* = F(w(t_0))$ ;
12:    end if
13:    if  $F(w(t_0)) < F^*$  then
14:       $t^* = t$ ;
15:    end if
16:    if STOP flag is set then
17:      Send  $t^*$  to all edge servers.
18:      break;
19:    end if
20:    Set  $L = \max\{L_i\}$ ,  $Y = \max\{Y_i\}$ ;
21:    Estimate  $\hat{\rho} = \frac{1}{N} \sum_{i=1}^N \hat{\rho}_i$ ;
22:    Estimate  $\hat{\beta} = \frac{1}{N} \sum_{i=1}^N \hat{\beta}_i$ ;
23:    Estimate  $\hat{\varphi} = \sum_{i=1}^N \hat{\varphi}_i$ ;
24:    Estimate  $\hat{C}$  according to (24) and (25);
25:    Use the PSO algorithm to find the optimal  $\tilde{\tau}^*$ ;
26:    Adjust  $\tau^* = \lambda * \tilde{\tau}^*$ ;
27:     $T = T + L\tau + Y$ ;
28:    if  $T + L(\tau + 1) + 2Y \geq Z$  then
29:      Set STOP flag;
30:    end if
31:  end if
32: until STOP flag is set
33: Send  $\tau^*, t^*$ , to all edge servers.
34: Receive  $f_i(w_i(t_0))$  from each edge server  $i$ ;
35: Compute  $F(w(t_0))$ ;
36: if  $F(w(t_0)) < F^*$  then
37:    $t^* = t$ ;
38: end if
39: Send  $t^*$ , to all edge servers.

```

servers perform the model synchronization after each training round. Please refer to Section III-A for the details.

- DFedAvgM [17], which enables the synchronization interval design in DSGD [35]. The synchronization interval is fixed over the training time, and is set at the interval of each five training rounds according to [17].
- DFedAvgM-enhanced, which is the enhanced version of DFedAvgM [17]. In particular, the synchronization interval is set to the optimum by enumerating all the possible values. Note that the synchronization interval

Algorithm 2 Procedure at Edge Server i

```

1: Output:  $w_i(t^*)$ 
2: Initialize  $t = 0$ ,  $w_i(0)$ ,  $t^* = 0$ ;
3: repeat
4:   Receive new  $\tau^*$ ,  $t^*$  from coordination node;
5:   Save  $w_i(t^*)$ ;
6:    $t_0 = t$ ;
7:   for  $\mu = 1, 2, \dots, \tau^*$  do
8:      $t = t + 1$ ;
9:      $w_i(t) = w_i(t - 1) - \gamma \nabla f_i(w_i(t - 1))$ ;
10:  end for
11:  Estimate  $\hat{\beta}_i = \frac{\|\nabla f(w_i(t_0)) - \nabla f(w_i(t))\|}{\|w_i(t_0) - w_i(t)\|}$ ;
12:  Estimate  $\hat{\rho}_i = \frac{\|f(w_i(t_0)) - f(w_i(t))\|}{\|w_i(t_0) - w_i(t)\|}$ ;
13:  Receive  $w_j(t)$ ,  $\nabla f_j(w_j(t))$  from adjacent edge servers
    and send  $w_i(t)$ ,  $\nabla f_i(w_i(t))$  to adjacent edge servers.
14:  Compute new  $w_i(t) = \sum_j a_{ij} w_j(t)$ ;
15:  Estimate  $\hat{\varphi}_i = \|\nabla f_i(w_i(t)) - \sum_{j \in \Omega_i} a_{ij} \nabla f_j(w_j(t))\|$ .
16:  Compute  $c_{i,j} = \|w_i(t) - w_j(t)\|$  for every  $j \in \Omega_i$ ;
17:  Send  $\hat{\varphi}_i$ ,  $L_i$ ,  $Y_i$ ,  $\hat{\beta}_i$ ,  $\hat{\rho}_i$ ,  $f_i(w_i(t_0))$  and all  $c_{i,j}$  to
    Coordination node;
18: until STOP flag is set
19: Receive  $t^*$  from aggregator;
20: Save  $w_i(t^*)$ ;

```

must be adjusted before/during the training process (not training for multiple times and selecting the optimal value). In other words, this approach is not practical and is only used for comparison purposes.

For brevity, we use ‘‘Proposed’’ to represent our approach.

B. RESULT DISCUSSION

In this section, we first showcase the effectiveness of our algorithm by evaluating its performance across different topologies and data distributions. Next, we examine the adaptability of our algorithm during the training process by presenting its temporal evolution within a time window. Finally, we demonstrate the performance of our algorithm in diverse network scenarios, providing evidence of its ability to adapt to varying network conditions.

1) PERFORMANCE IN DIFFERENT TOPOLOGIES AND DATA DISTRIBUTIONS

Fig. 5 displays the Top-1 test accuracy of our algorithm and the DFedAvgM [17] algorithm in a ring topology, considering different fixed values of τ as the time window cutoff. It is evident that our algorithm consistently outperforms the DFedAvgM [17] algorithm across almost all values of τ . This superiority arises from our algorithm’s adaptive adjustments to the synchronization interval, tailored to the specific training circumstances. We carefully select the optimized synchronization interval at each step.

In Table 4, we present a summary of the results obtained for various topologies and data distributions.

The table includes the difference in Top-1 test accuracy between our algorithm, the DSGD [35] algorithm, and the DFedAvgM [17] ($\tau = 5$) algorithm in each case. Our algorithm exhibits significantly better performance compared to these two algorithms. On the MNIST dataset, our algorithm achieves a 38.65% higher test accuracy than DSGD [35] and a 13.3% higher accuracy than DFedAvgM [17] ($\tau = 5$). In the case of FMNIST dataset, the accuracy achieved by our algorithm is 24.76% and 6.23% higher than DSGD [35] and DFedAvgM [17] ($\tau = 5$), respectively. On the CIFAR10 dataset, our algorithm demonstrates a 26.76% improvement over DSGD [35] and a 16.23% improvement over DFedAvgM [17] ($\tau = 5$).

Furthermore, as the connection density in the communication topology increases, the performance improvement of our algorithm becomes more pronounced. This is because denser communication topologies facilitate increased communication between servers during each synchronization, leading to enhanced system consistency. Consequently, a larger communication interval is required. Our algorithm can adaptively select a larger communication interval to accommodate this need, whereas the DSGD [35] and DFedAvgM [17] algorithms lack this adaptive adjustment, resulting in better performance for our algorithm.

2) PERFORMANCE IN LATENCY-CONSTRAINED SETTING

Fig. 6 depicts the evolution of test accuracy within a limited time window for our algorithm and three other baselines. The test model is obtained by averaging the models from all edge servers. It is important to note that this testing method is not employed during actual training and is solely used for monitoring the algorithm’s execution without impacting the overall training process. In many decentralized federated learning algorithms, such as those proposed in [51], [52], [53], the models from all servers are averaged at the end of the algorithm to produce the final output result. Consequently, during testing, we utilize the test results obtained from the average model as experimental observations, a process that does not affect model training. Additionally, this approach allows us to observe the training effect of the model when the training is stopped at any time.

From the figure, it can be observed that our algorithm closely approximates the results of DFedAvgM-advance and outperforms the other two algorithms. DFedAvgM-advance requires iterating through all τ values within a specified range to select the optimal case based on the results. The effectiveness of our algorithm is evident as it achieves results comparable to or even surpassing DFedAvgM-advance, outperforming the other two algorithms.

In Fig. 7, we present the changes over time in our adaptively adjusted τ value and the system consistency metric $C(t)$ in the ring topology. A correlation can be observed between the changing trend of τ and $C(t)$. Initially, when all edge servers share the same model, the value

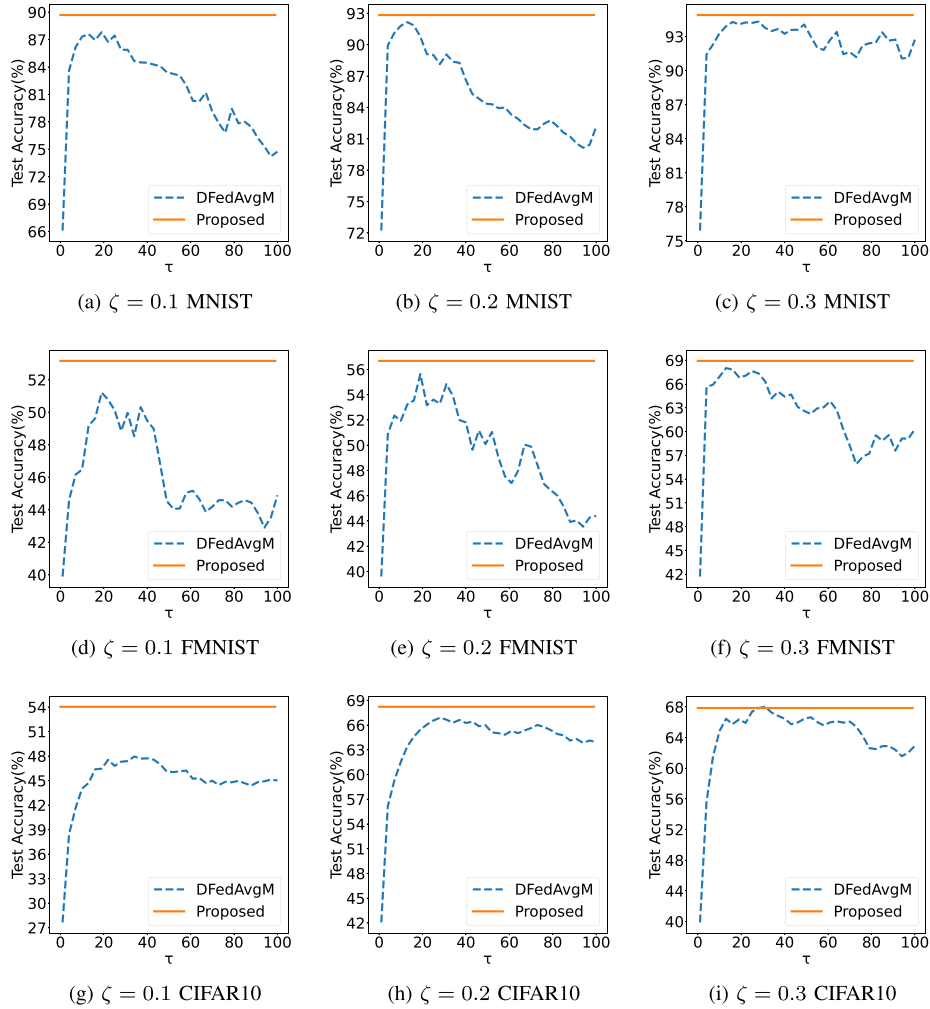


FIGURE 5. Top-1 test accuracy under different values of τ in the ring topology.

TABLE 4. The performance gap between our algorithm, DSGD algorithm, and DFedAvgM ($\tau = 5$) algorithm in various scenarios. In each cell, The number on the left and the number on the right represents the improvement of the Top-1 test accuracy rate of our algorithm compared with the DSGD algorithm and DFedAvgM algorithm, respectively. For example, the number “16.96/8.1” in the first cell indicates that the accuracy of our algorithm is 16.96% higher than that of the DSGD algorithm and 8.1% higher than that of the DFedAvgM algorithm.

	tree $\alpha = 0.93$	ring $\alpha = 0.87$	grid $\alpha = 0.80$	links5 $\alpha = 0.66$	links7 $\alpha = 0.60$
MNIST $\zeta = 0.1$	16.96/8.1	23.59/6.06	33.37/12.13	47.99/20.6	77.12/34.86
MNIST $\zeta = 0.2$	27.09/13.27	20.62/2.91	18.52/10.35	61.18/16.31	58.57/27.52
MNIST $\zeta = 0.3$	22.27/3.21	18.99/3.47	36.66/5.82	55.19/10.91	61.66/23.99
FMNIST $\zeta = 0.1$	13.36/ 7.27	13.27/ 8.57	9.05/ 4.27	22.19/ 8.46	35.2/ 5.43
FMNIST $\zeta = 0.2$	15.03/ 0.93	17.08/ 5.82	19.85/ 1.33	28.94/ 8.99	33.1/ 6.78
FMNIST $\zeta = 0.3$	10.83/ 5.77	27.21/ 4.3	28.82/ 0.92	47.89/ 12.41	47.44/ 17.2
CIFAR10 $\zeta = 0.1$	17.49/6.59	26.38/15.59	25.96/14.24	27.9/24.69	26.87/22.21
CIFAR10 $\zeta = 0.2$	22.54/13.27	26.16/12.09	28.25/15.15	29.27/21.57	29.27/21.57
CIFAR10 $\zeta = 0.3$	32.53/10.63	27.98/12.26	26.56/12.75	29.29/18.43	25.82/23.79

of $C(t)$ is small. This encourages larger adjustments to τ . Conversely, a larger τ implies less communication, resulting in a larger $C(t)$ value, which limits further increases in τ . Simultaneously, as the training progresses, we aim to minimize disparities in the system, leading to a gradual

decrease in τ . Therefore, we conclude that the system consistency metric $C(t)$ plays a crucial role. This observation aligns with the convergence analysis results and demonstrates the adaptability of our algorithm throughout the training process.

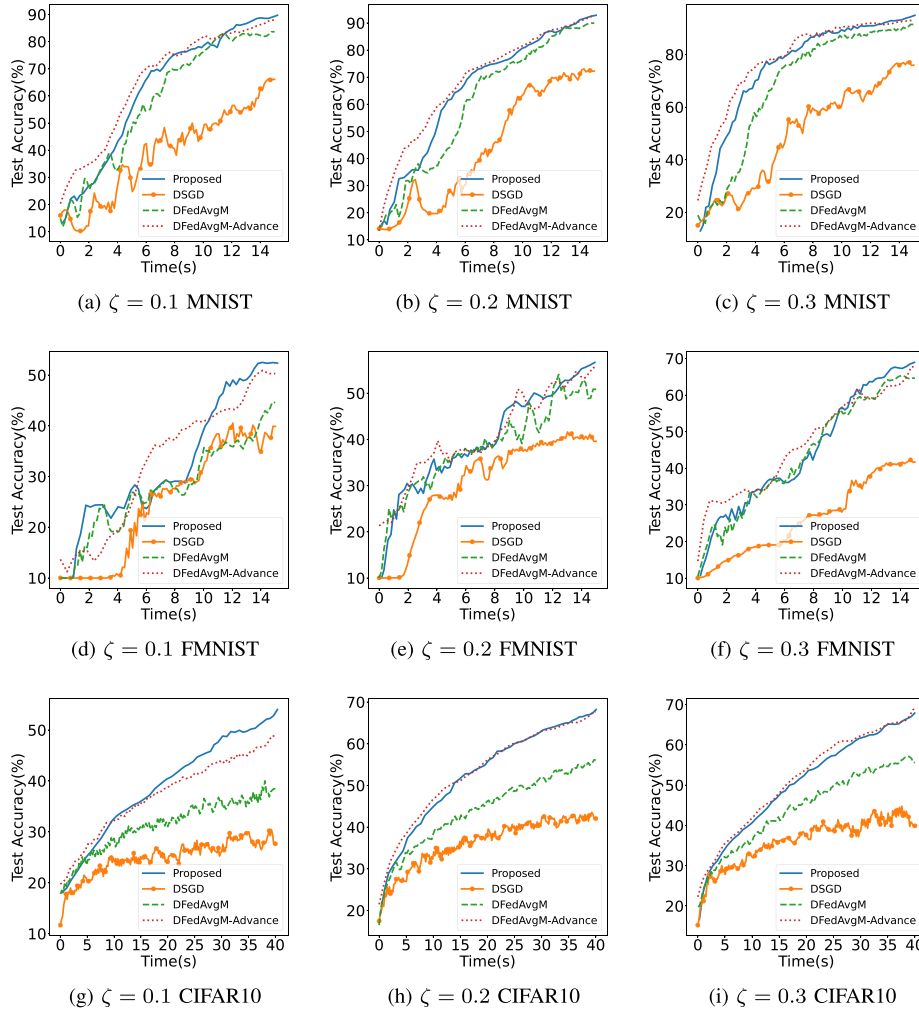


FIGURE 6. The Top-1 test accuracy of our algorithm and the benchmark algorithms changes over time within the time window in the ring topology.

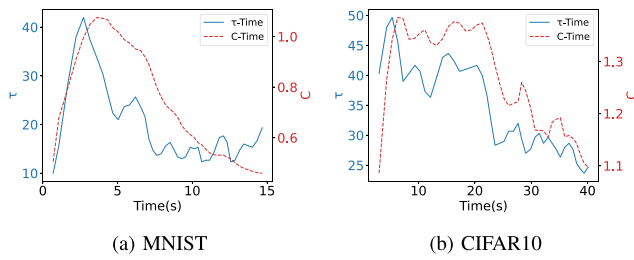


FIGURE 7. The adaptive selection of τ of the proposed algorithm over the training time.

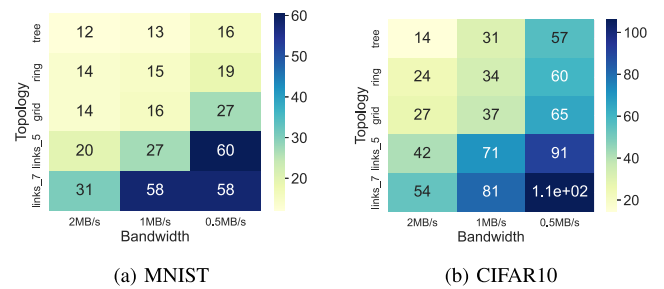


FIGURE 8. Performance in different network conditions. The average synchronization interval under different bandwidths and topologies. The darker the color, the larger the mean synchronization interval.

3) EFFECTIVENESS OF STRIKING THE BALANCE BETWEEN THE COMMUNICATION AND COMPUTATION

The outcomes presented in Table 4 were obtained using a bandwidth of 1MB/s. When the bandwidth is reduced to 0.5MB/s, our algorithm achieves a test accuracy that is 38.22% and 20.04% higher than DSGD [35] and DFedAvgM [17], respectively. Conversely, when the bandwidth is increased to 2MB/s, our algorithm exhibits a test

accuracy improvement of 23.49% and 7.25% compared to DSGD [35] and DFedAvgM [17], respectively. These results clearly demonstrate the superior performance of our algorithm across different topologies and bandwidths, thanks to its adaptive adjustment of the synchronization interval.

To further illustrate the adaptability of our algorithm under various network scenarios (considering bandwidth

and topology), we present the results of the average τ in Fig. 8. As previously mentioned, our algorithm consistently outperforms both DSGD and DFedAvgM algorithms in these scenarios. We modify the communication topology and adjust the bandwidth setting for each link, using $\zeta = 0.1$ as a representative case. We compare the average synchronization intervals throughout the training process across five distinct topologies (tree, ring, grid, links5, and links7) and three bandwidth options (0.5 MB/s, 1 MB/s, and 2 MB/s). The figure clearly demonstrates that as the bandwidth decreases and the connection density in the communication topology increases, the average τ value also increases.

Based on the convergence analysis, a higher connection density in the communication topology indicates stronger system consistency, allowing for longer synchronization intervals. Conversely, as the system bandwidth decreases, communication consumption increases, necessitating an adjustment in the synchronization interval to maintain an appropriate communication frequency. This validates the adaptability of our algorithm, showcasing its ability to strike a balance between communication and computation.

VII. CONCLUSION

This paper first studied the synchronization interval optimization for latency-constrained geo-decentralized FL. The objective is to optimize the model accuracy within a time window. We mathematically derive the convergence bound to reveal its relationship with network topology, data heterogeneity, and communication/computation resources, facilitating the design of the proposed interval optimization approach. Experimental results validate the effectiveness of the proposed approach under different topologies, datasets, data distributions and communication/computation capabilities and demonstrate the adaptability of our algorithm during the training process and across various network scenarios.

APPENDIX

A. PROOF OF THEOREM 1

Before proving the Theorem 1, let's introduce a lemma.

Lemma 2: We use $\bar{w}(k\tau)$ to represent the average of the edge server models before the k th global average and use $\bar{w}(k\tau)^+$ to represent the average of the edge server models before the k th global average. We have:

$$\bar{w}(k\tau) = \bar{w}(k\tau)^+.$$

This lemma tells us that the mean of all edge server models does not change because the models are synchronized.

Proof:

$$\begin{aligned} \bar{w}(k\tau)^+ &= \frac{1}{N} \sum_{i=1}^N w_i(k\tau)^+ \\ &= \frac{1}{N} 1_N^T [w(k\tau)^+] \end{aligned}$$

From equation (5) we know:

$$[w(k\tau)^+] = A[w(k\tau)].$$

So we can change the equation:

$$\begin{aligned} \bar{w}(k\tau)^+ &= \frac{1}{N} 1_N^T A[w(k\tau)] \\ &= \frac{1}{N} 1_N^T [w(k\tau)] \\ &= \bar{w}(k\tau) \end{aligned}$$

A is a doubly stochastic matrix, so we have $1_N^T A = 1_N^T$. ■

Then we can begin the proof of Theorem 1.

$$\begin{aligned} &\|\bar{w}(k\tau) - s(k\tau)\| \\ &= \left\| \frac{1}{N} \sum_{i=1}^N \left(w_i^{k-1}(k\tau) - s^{k-1}(k\tau) \right) \right\| \\ &\leq \frac{1}{N} \sum_{i=1}^N \left\| \left(w_i^{k-1}(k\tau) - s^{k-1}(k\tau) \right) \right\| \end{aligned} \quad (31)$$

For any i , we have:

$$\begin{aligned} &\left\| w_i^{k-1}(k\tau) - s^{k-1}(k\tau) \right\| \\ &= \left\| \left(w_i^{k-1}(k\tau - 1) - \gamma \nabla f_i \left(w_i^{k-1}(k\tau - 1) \right) \right) \right. \\ &\quad \left. - \left(s^{k-1}(k\tau - 1) - \gamma \nabla F \left(s^{k-1}(k\tau - 1) \right) \right) \right\| \\ &= \left\| \left(w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right) - \gamma \left(\nabla f_i \left(w_i^{k-1}(k\tau - 1) \right) \right. \right. \\ &\quad \left. \left. - \nabla f_i \left(s^{k-1}(k\tau - 1) \right) \right) - \gamma \left(\nabla F \left(s^{k-1}(k\tau - 1) \right) \right. \right. \\ &\quad \left. \left. - \nabla f_i \left(s^{k-1}(k\tau - 1) \right) \right) \right\| \\ &\leq \left\| \left(w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right) \right\| \\ &\quad + \left\| \gamma \left(\nabla f_i \left(w_i^{k-1}(k\tau - 1) \right) - \nabla f_i \left(s^{k-1}(k\tau - 1) \right) \right) \right\| \\ &\quad + \left\| \gamma \left(\nabla F \left(s^{k-1}(k\tau - 1) \right) - \nabla f_i \left(s^{k-1}(k\tau - 1) \right) \right) \right\| \\ &\leq \left\| \left(w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right) \right\| \\ &\quad + \gamma \beta_i \left\| \left(w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right) \right\| + \gamma \varphi_i \\ &\leq (\gamma \beta_i + 1) \left\| \left(w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right) \right\| + \gamma \varphi_i \end{aligned}$$

In a similar way, we can get that

$$\begin{aligned} &\left\| w_i^{k-1}(k\tau - 1) - s^{k-1}(k\tau - 1) \right\| \\ &\leq (\gamma \beta_i + 1) \left\| w_i^{k-1}(k\tau - 2) - s^{k-1}(k\tau - 2) \right\| + \gamma \varphi_i. \end{aligned}$$

After τ times iterating we have:

$$\begin{aligned} &\left\| w_i^{k-1}(k\tau) - s^{k-1}(k\tau) \right\| \\ &\leq \left(\left\| w_i^{k-1}((k-1)\tau) - s^{k-1}((k-1)\tau) \right\| + \frac{\varphi_i}{\beta_i} \right) \\ &\quad \left((\gamma \beta_i + 1)^\tau - 1 \right) - \varphi_i \gamma \tau \\ &\leq \left(\left\| w_i^{k-1}((k-1)\tau) - \bar{w}((k-1)\tau) \right\| + \frac{\varphi_i}{\beta_i} \right) \\ &\quad \left((\gamma \beta_i + 1)^\tau - 1 \right) - \varphi_i \gamma \tau \\ &\leq \left(C_i^{k-1}((k-1)\tau) + \frac{\varphi_i}{\beta_i} \right) \left((\gamma \beta_i + 1)^\tau - 1 \right) - \varphi_i \gamma \tau \end{aligned} \quad (32)$$

We can substitute Eq. (32) into Eq. (31) to get the following result:

$$\begin{aligned} & \left\| \bar{w}(k\tau) - s^{k-1}(k\tau) \right\| \\ & \leq \left(C^{k-1}((k-1)\tau) + \frac{\varphi}{\beta} \right) ((\gamma\beta + 1)^\tau - 1) - \varphi\gamma\tau. \quad (33) \end{aligned}$$

B. PROOF OF LEMMA 1

Then we need to bound $C^{k-1}((k-1)\tau)$. We use $w_i^k(t)$ to denote w_i at iteration t between the k -th synchronization and the $(k+1)$ -th synchronization

$$\begin{aligned} C^k(k\tau) &= \frac{1}{N} \sum_{i=1}^N C_i^k(k\tau) \\ &= \frac{1}{N} \sum_{i=1}^N \left\| w_i^k(k\tau) - \bar{w}(k\tau) \right\|. \end{aligned}$$

We first bound $\| [w_i^k(k\tau)] - [\bar{w}(k\tau)] \|$:

$$\begin{aligned} & \left\| [w_i^k(k\tau)] - [\bar{w}(k\tau)] \right\| \\ &= \left\| A[w_i^{k-1}(k\tau)] - \frac{1}{N} 1_N 1_N^T [w_i^{k-1}((k)\tau)] \right\| \\ &= \left\| \left(A - \frac{1}{N} 1_N 1_N^T \right) [w_i^{k-1}(k\tau)] \right\| \\ &= \left\| \left(A - \frac{1}{N} 1_N 1_N^T \right) \left([w_i^{k-1}(k\tau - 1)] \right. \right. \\ & \quad \left. \left. - \gamma [\nabla f_i(w_i^{k-1}(k\tau - 1))] \right) \right\| \\ &= \left\| \left(A - \frac{1}{N} 1_N 1_N^T \right) \left([w_i^{k-1}((k-1)\tau)] \right. \right. \\ & \quad \left. \left. - \sum_{j=0}^{\tau-1} \gamma [\nabla f_i(w_i^{k-1}((k-1)\tau + j))] \right) \right\| \\ &= \left\| \left(A - \frac{1}{N} 1_N 1_N^T \right) \left(A [w_i^{k-2}((k-1)\tau)] \right. \right. \\ & \quad \left. \left. - \sum_{j=0}^{\tau-1} \gamma [\nabla f_i(w_i^{k-2}((k-1)\tau + j))] \right) \right\| \\ &= \left\| \left(A^2 - \frac{1}{N} 1_N 1_N^T \right) [w_i^{k-2}((k-1)\tau)] \right. \\ & \quad \left. - \sum_{j=0}^{\tau-1} \gamma \left(A - \frac{1}{N} 1_N 1_N^T \right) [\nabla f_i(w_i^{k-2}((k-1)\tau + j))] \right\|. \end{aligned}$$

We can use the same method to iterate again:

$$\begin{aligned} & \left\| [w_i^k(k\tau)] - [\bar{w}(k\tau)] \right\| \\ &= \left\| \left(A^3 - \frac{1}{N} 1_N 1_N^T \right) [w_i^{k-3}((k-2)\tau)] \right. \\ & \quad \left. - \sum_{j=0}^{\tau-1} \gamma \left(A^2 - \frac{1}{N} 1_N 1_N^T \right) [\nabla f_i(w_i^{k-3}((k-2)\tau + j))] \right\| \\ & \quad \left. - \sum_{j=0}^{\tau-1} \gamma \left(A - \frac{1}{N} 1_N 1_N^T \right) [\nabla f_i(w_i^{k-2}((k-1)\tau + j))] \right\| \end{aligned}$$

After k times iterations we have:

$$\begin{aligned} & \left\| w_i^k(k\tau)^+ - s^k(k\tau) \right\| \\ &= \left\| \left(A^k - \frac{1}{N} 1_N 1_N^T \right) [w_i^0(\tau)] \right. \\ & \quad \left. - \sum_{p=1}^{k-1} \sum_{j=0}^{\tau-1} \gamma \left(A^p - \frac{1}{N} 1_N 1_N^T \right) [\nabla f_i(w_i^{k-p-1}((k-p)\tau + j))] \right\| \\ &= \left\| \left(A^k - \frac{1}{N} 1_N 1_N^T \right) [w_i(0)] \right. \\ & \quad \left. - \sum_{p=1}^k \sum_{j=0}^{\tau-1} \gamma \left(A^p - \frac{1}{N} 1_N 1_N^T \right) [\nabla f_i(w_i^{k-p-1}((k-p)\tau + j))] \right\| \\ &\leq \left\| \left(A^k - \frac{1}{N} 1_N 1_N^T \right) \| [w_i(0)] \| \right. \\ & \quad \left. + \sum_{p=1}^k \sum_{j=0}^{\tau-1} \gamma \left\| A^p - \frac{1}{N} 1_N 1_N^T \right\| \left\| [\nabla f_i(w_i^{k-p-1}((k-p)\tau + j))] \right\| \right\| \\ &\leq \alpha^k \| [w_i(0)] \| + \sum_{p=1}^{k-1} \sum_{j=0}^{\tau-1} \gamma \alpha^p \left\| [\nabla f_i(w_i^{k-p-1}((k-p)\tau + j))] \right\| \end{aligned}$$

From Assumption IV-A we know that the function $F_i(w)$ is ρ -Lipschitz. So it can be deduced that for any i

$$\| \nabla F_i(w) \| \leq \rho.$$

Then we can bound $\| [\nabla f_i(w_i^{k-p-1}((k-p)\tau + j))] \|$:

$$\| [\nabla F_i(w_i)] \| \leq \sum_{i=1}^N \| \nabla F_i(w_i) \| \leq N\rho$$

Finally, we have

$$\begin{aligned} C^k(k\tau) &= \left\| [w_i^k(k\tau)] - [\bar{w}(k\tau)] \right\| \\ &\leq \alpha^k \| [w_i(0)] \| + \sum_{p=1}^k \tau \gamma \alpha^p N \rho \\ &\leq \alpha \| [w_i(0)] \| + \frac{\alpha \tau N \rho \gamma}{1 - \alpha}. \end{aligned}$$

C. PROOF OF THEOREM 2

This part of the proof we refer to [27], some of the following steps are a repetition of [27].

For the convenience of proof, we make a definition.

$$v^k(t) = F(s^k(t)) - F(w^*).$$

According to [47, Th. 3.14], for any finite t and k we always have

$$v^k(t) > 0. \quad (34)$$

Lemma 3: When $\gamma \leq \frac{1}{\beta}$, for any k , and $t \in [k\tau, (k+1)\tau)$, we have

$$\| s^k(t+1) - w^* \|^2 \leq \| s^k(t) - w^* \|^2$$

Proof: The loss function F is β -smooth and $v^k(t) > 0$, so we have

$$0 < v^k(t) \leq \nabla F(s^k(t))^T (s^k(t) - w^*) - \frac{\| \nabla F(s^k(t)) \|^2}{2\beta}. \quad (35)$$

Therefore,

$$\begin{aligned}
 & \|s^k(t+1) - w^*\|^2 \\
 &= \|s^k(t) - \gamma \nabla F(s^k(t)) - w^*\|^2 \\
 &= \|s^k(t) - w^*\|^2 - 2\gamma \nabla F(s^k(t))^\top (s^k(t) - w^*) \\
 &\quad + \gamma^2 \|\nabla F(s^k(t))\|^2 \\
 &< \|s^k(t) - w^*\|^2 - \gamma \frac{\|\nabla F(s^k(t))\|^2}{\beta} + \gamma^2 \|\nabla F(s^k(t))\|^2 \\
 &\quad \text{(From Eq. (35))} \\
 &= \|s^k(t) - w^*\|^2 - \gamma \left(\frac{1}{\beta} - \gamma \right) \|\nabla F(s^k(t))\|^2.
 \end{aligned}$$

If $\gamma \leq \frac{1}{\beta}$, we can obtain

$$\|s^k(t+1) - w^*\|^2 \leq \|s^k(t) - w^*\|^2. \quad \blacksquare$$

Lemma 4: For any k , when $\gamma \leq \frac{1}{\beta}$ and $t \in [k\tau, (k+1)\tau)$, we have

$$F(s^k(t+1)) - F(s^k(t)) \leq -\gamma \left(1 - \frac{\beta\gamma}{2}\right) \|\nabla F(s^k(t))\|^2 \quad (36)$$

Proof: Because the function $F(\cdot)$ is β -smooth, from [47, Lemma 3.4], we have

$$F(x) \leq F(y) + \nabla F(y)^\top (x - y) + \frac{\beta}{2} \|x - y\|^2$$

for arbitrary x and y . Thus,

$$\begin{aligned}
 & F(s^k(t+1)) - F(s^k(t)) \\
 &\leq \nabla F(s^k(t))^\top (s^k(t+1) - s^k(t)) \\
 &\quad + \frac{\beta}{2} \|s^k(t+1) - s^k(t)\|^2 \\
 &\leq -\gamma \nabla F(s^k(t))^\top \nabla F(s^k(t)) + \frac{\beta\gamma^2}{2} \|\nabla F(s^k(t))\|^2 \\
 &\leq -\gamma \left(1 - \frac{\beta\gamma}{2}\right) \|\nabla F(s^k(t))\|^2.
 \end{aligned} \quad \blacksquare$$

Lemma 5: For any k , when $\gamma \leq \frac{1}{\beta}$ and $t \in [k\tau, (k+1)\tau)$, we have

$$\frac{1}{v^k(t+1)} - \frac{1}{v^k(t)} \geq \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right) \quad (37)$$

where $\frac{1}{\varpi} = \min_k \frac{1}{\|s^k((k-1)\tau) - w^*\|^2}$

Proof: Substituting the definition of v^k into (36) can get

$$v^k(t+1) - v^k(t) \leq -\gamma \left(1 - \frac{\beta\gamma}{2}\right) \|\nabla F(s^k(t))\|^2.$$

Equivalently,

$$v^k(t+1) \leq v^k(t) - \gamma \left(1 - \frac{\beta\gamma}{2}\right) \|\nabla F(s^k(t))\|^2. \quad (38)$$

Then we have

$$\begin{aligned}
 v^k(t) &= F(s^k(t)) - F(w^*) \leq \nabla F(s^k(t))^\top (s^k(t) - w^*) \\
 &\leq \|\nabla F(s^k(t))\| \|s^k(t) - w^*\|.
 \end{aligned}$$

Hence,

$$\frac{v^k(t)}{\|s^k(t) - w^*\|} \leq \|\nabla F(s^k(t))\|. \quad (39)$$

In Lemma 3, we have proven that for any k , $\|s^k(t+1) - w^*\|^2 \leq \|s^k(t) - w^*\|^2$ when $t \in [k\tau, (k+1)\tau)$. Hence, $\|s^k(k\tau) - w^*\| \geq \|s^k(t) - w^*\|$. Then we define $\varpi = \max_k \|s^k(k\tau) - w^*\|^2$, and have $-\frac{1}{\varpi} \geq \frac{-1}{\|s^k(k\tau) - w^*\|^2} \geq \frac{-1}{\|s^k(t) - w^*\|^2}$. Using this inequality relationship and combining (39) and (38), we get

$$\begin{aligned}
 v^k(t+1) &\leq v^k(t) - \frac{\gamma \left(1 - \frac{\beta\gamma}{2}\right) v^k(t)^2}{\|s^k(t) - w^*\|^2} \\
 &\leq v^k(t) - \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right) v^k(t)^2.
 \end{aligned}$$

As $v^k(t+1)v^k(t) > 0$ according to (34), dividing both sides by $v^k(t+1)v^k(t)$, we obtain

$$\frac{1}{v^k(t)} \leq \frac{1}{v^k(t+1)} - \frac{\gamma \left(1 - \frac{\beta\gamma}{2}\right) v^k(t)}{\varpi v^k(t+1)}.$$

From (34) and (38) it can be obtained that $0 < v^k(t+1) \leq v^k(t)$, so we have $\frac{v^k(t)}{v^k(t+1)} \geq 1$. Hence,

$$\frac{1}{v^k(t+1)} - \frac{1}{v^k(t)} \geq \frac{\gamma \left(1 - \frac{\beta\gamma}{2}\right) v^k(t)}{\varpi v^k(t+1)} \geq \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right). \quad \blacksquare$$

From Lemma 5, for any $t \in [k\tau, (k+1)\tau)$ we have

$$\begin{aligned}
 & \frac{1}{v^k((k+1)\tau)} - \frac{1}{v^k(k\tau)} \\
 &= \sum_{t=k\tau}^{(k+1)\tau-1} \left(\frac{1}{v^k(t+1)} - \frac{1}{v^k(t)} \right) \\
 &\geq \tau \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right).
 \end{aligned}$$

We sum up all the values of $k = 0, 1, 2, \dots, K-1$ to get

$$\begin{aligned}
 \sum_{k=0}^{K-1} \left(\frac{1}{v^k((k+1)\tau)} - \frac{1}{v^k(k\tau)} \right) &\geq \sum_{k=0}^{K-1} \tau \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right) \\
 &= K\tau \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right).
 \end{aligned}$$

The left-hand side of this equation can be split as

$$\begin{aligned}
 & \frac{1}{v^{K-1}(T)} - \frac{1}{v^0(0)} - \sum_{k=0}^{K-2} \left(\frac{1}{v^{k+1}((k+1)\tau)} - \frac{1}{v^k((k+1)\tau)} \right) \\
 &\geq T \frac{1}{\varpi} \gamma \left(1 - \frac{\beta\gamma}{2}\right)
 \end{aligned}$$

which is equivalent to

$$\begin{aligned} & \frac{1}{v^{K-1}(T)} - \frac{1}{v^0(0)} \\ & \geq T \frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) + \\ & \sum_{k=0}^{K-2} \left(\frac{1}{v^{k+1}((k+1)\tau)} - \frac{1}{v^k((k+1)\tau)} \right). \end{aligned} \quad (40)$$

For any k , we have

$$\begin{aligned} & \frac{1}{v^{k+1}((k+1)\tau)} - \frac{1}{v^k((k+1)\tau)} \\ & = \frac{v^k((k+1)\tau) - v^{k+1}((k+1)\tau)}{v^k((k+1)\tau)v^{k+1}((k+1)\tau)} \\ & = \frac{F(s^k((k+1)\tau)) - F(s^{k+1}((k+1)\tau))}{v^k((k+1)\tau)v^{k+1}((k+1)\tau)} \\ & = \frac{F(s^k((k+1)\tau)) - F(\bar{w}((k+1)\tau))}{v^k((k+1)\tau)v^{k+1}((k+1)\tau)} \\ & \geq \frac{-\rho\Theta(\tau)}{v^k((k+1)\tau)v^{k+1}((k+1)\tau)}. \end{aligned} \quad (41)$$

From Lemma 4 we know that $F(s^k(t)) \geq F(s^k(t+1))$ for any $t \in [k\tau, (k+1)\tau)$. So we can obtain $v^k(k\tau) = F(s^k(t)) - F(w^*) \geq F(s^k((k+1)\tau)) - F(w^*) \geq \varepsilon$ for all k (we have assumed that $F(s^k((k+1)\tau)) - F(w^*) \geq \varepsilon$). Then we have

$$\begin{aligned} & v^k(k\tau)v^{k+1}(k\tau) \geq \varepsilon^2 \\ & \frac{-1}{v^k(k\tau)v^{k+1}(k\tau)} \geq -\frac{1}{\varepsilon^2}. \end{aligned} \quad (42)$$

Combining (42) with (41), the RHS of (40) have

$$\sum_{k=0}^{K-1} \left(\frac{1}{v^{k+1}(k\tau)} - \frac{1}{v^k(k\tau)} \right) \geq -K \frac{\rho\Theta(\tau)}{\varepsilon^2}. \quad (43)$$

From (43) and (40), we have

$$\frac{1}{v^k(T)} - \frac{1}{v^0(0)} \geq T \frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - K \frac{\rho\Theta(\tau)}{\varepsilon^2}. \quad (44)$$

We also assumed that $F(\bar{w}(T)) - F(w^*) \geq \varepsilon$. So we have

$$\frac{-1}{(F(\bar{w}(T)) - F(w^*))v^k(T)} \geq -\frac{1}{\varepsilon^2}. \quad (45)$$

$$\begin{aligned} & \frac{1}{F(\bar{w}(T)) - F(w^*)} - \frac{1}{v^k(T)} \\ & \geq -\frac{(v^k(T) - F(w^*)) - F(\bar{w}(T))}{\varepsilon^2} \\ & = \frac{F(\bar{w}(T)) - F(s^k(T))}{\varepsilon^2} \\ & \geq -\frac{\rho\Theta(\tau)}{\varepsilon^2}. \end{aligned} \quad (46)$$

From (44) and (40), we can obtain

$$\frac{1}{F(\bar{w}(T)) - F(w^*)} - \frac{1}{v^0(0)}$$

$$\begin{aligned} & \geq T \frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - K \frac{\rho\Theta(\tau)}{\varepsilon^2} \\ & = T \frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - T \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \\ & = T \left(\frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \right). \end{aligned}$$

We note that

$$\begin{aligned} & \frac{1}{F(\bar{w}(T)) - F(w^*)} \\ & \geq \frac{1}{F(\bar{w}(T)) - F(w^*)} - \frac{1}{v^0(0)} \\ & \geq T \left(\frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \right) > 0 \end{aligned}$$

where the last inequality is because we assumed that $\frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} > 0$. Taking the reciprocal of the above inequality yields

$$\begin{aligned} F(\bar{w}(T)) - F(w^*) & \leq \frac{1}{T \left(\frac{1}{\omega} \gamma \left(1 - \frac{\beta\gamma}{2}\right) - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \right)} \\ & \leq \frac{1}{T \left(\frac{\gamma}{2\omega} - \frac{\rho\Theta(\tau)}{\tau\varepsilon^2} \right)} \quad (\gamma\beta \leq 1) \end{aligned}$$

So here we have finished our proof of Eq. (18).

We set

$$\varepsilon_0 = \frac{1}{T \left(\frac{\gamma}{2\omega} - \frac{\rho\Theta(\tau)}{\tau\varepsilon_0^2} \right)}. \quad (47)$$

Solving for ε_0 , we obtain

$$\varepsilon_0 = \sqrt{\frac{1}{4\gamma^2\omega^2T^2} + \frac{\rho h(\tau)}{\gamma\omega\tau}} + \frac{1}{2\gamma\omega T} \quad (48)$$

By setting the value of ε in this way, all the preconditions in Theorem 2 can be satisfied. Now we have

$$F(\bar{w}(T)) - F(w^*) \leq \frac{1}{T \left(\gamma\omega - \frac{\rho h(\tau)}{\tau\varepsilon^2} \right)} < \frac{1}{T \left(\gamma\omega - \frac{\rho h(\tau)}{\tau\varepsilon_0^2} \right)} = \varepsilon_0.$$

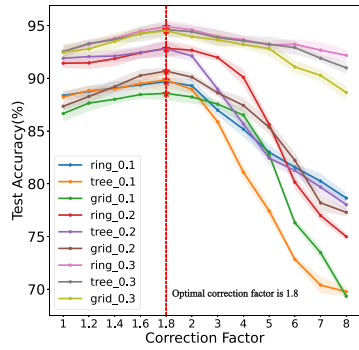
Therefore, there does not exist $\varepsilon > \varepsilon_0$ that satisfy both $F(\bar{w}(T)) - F(w^*) \geq \varepsilon$ and $F(s^{k-1}(k\tau)) - F(w^*) \geq \varepsilon$. This means that either 1) $\exists k$ such that $F(s^{k-1}(k\tau)) - F(w^*) \leq \varepsilon_0$ or 2) $F(\bar{w}(T)) - F(w^*) \leq \varepsilon_0$. It follows that

$$\min \left\{ \min_{k=0,1,\dots,K} F(s^{k-1}(k\tau)); F(\bar{w}(T)) \right\} - F(w^*) \leq \varepsilon_0. \quad (49)$$

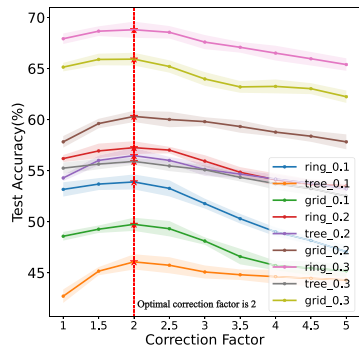
In Theorem 1 we get $F(\bar{w}(k\tau)) \leq F(s^{k-1}(k\tau)) + \rho h(\tau)$ for any k . Combining with (48) and (VII-C), we get

$$\begin{aligned} & \min_{k=1,2,\dots,K} F(\bar{w}(k\tau)) - F(w^*) \\ & \leq \sqrt{\frac{1}{4\gamma^2\omega^2T^2} + \frac{\rho h(\tau)}{\gamma\omega\tau}} + \frac{1}{2\gamma\omega T} + \rho h(\tau). \end{aligned}$$

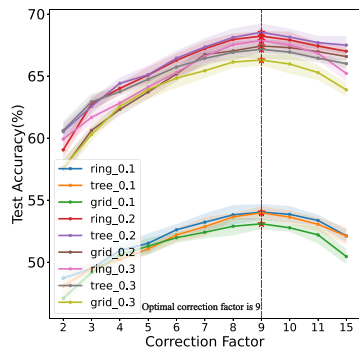
Here we obtain Eq. (19).



(a) MNIST



(b) FMNIST



(c) CIFAR10

FIGURE 9. The top-1 accuracy when selecting different values of correction factors on MNIST, FMNIST, and CIFAR10 datasets.

D. SELECTION OF CORRECTION FACTOR

The performance analysis in Section IV is to find the upper bound of the loss function, where various inequalities are introduced to find and prove the bound. The bound is necessary for the tractability of the formulated problem. However, there is always a gap between the bound and actual values due to the use of inequalities. To achieve better performance, we experimentally check and select the correction factors and find that the optimal correction factors are nearly the same given a specific dataset, making it

practical to implement the correction factor. The experimental results on assessing the correction factor on MNIST, FMNIST, and CIFAR10 datasets are plotted in Fig. 9, where each data point is the average of the results from seven random seeds. We emphasize the highest achieved accuracy in each curve in Fig. 9. Notably, the optimal correction factor remains consistent irrespective of different topologies and data distributions. The optimal correction factors for MNIST, FMNIST and CIFAR10 datasets are $\lambda = 1.8, 2$ and 9 , respectively. The use of correction factor can increase about 5% accuracy of the proposed approach on average.

REFERENCES

- [1] M. Al-Quraan et al., "Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 957–979, Jun. 2023.
- [2] 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, Sep. 2020.
- [3] M. Chen et al., "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.
- [4] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
- [5] X. Lyu, J. Liu, C. Ren, and G. Nan, "Security-computation-computation tradeoff of split decisions for edge intelligence," *IEEE Wireless Commun.*, early access, Jan. 16, 2023, doi: [10.1109/MWC.014.2200438](https://doi.org/10.1109/MWC.014.2200438).
- [6] X. Lyu et al., "Secure and efficient federated learning with provable performance guarantees via stochastic quantization," *IEEE Trans. Inf. Forensics Security*, early access, Mar. 7, 2024, doi: [10.1109/TIFS.2024.3374590](https://doi.org/10.1109/TIFS.2024.3374590).
- [7] J. Liu, X. Lyu, Q. Cui, and X. Tao, "Similarity-based label inference attack against training and inference of split learning," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2881–2895, 2024.
- [8] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.
- [9] D. C. Nguyen, "Federated learning for smart healthcare: A survey," *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–37, 2022.
- [10] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.
- [11] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.
- [12] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 8851–8855.
- [13] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl. Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.
- [14] K. Hsieh, "Gaia: Geo-distributed machine learning approaching LAN speeds," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 629–647.
- [15] Z. Liang, P. Yang, C. Zhang, and X. Lyu, "Secure and efficient hierarchical decentralized learning for Internet of Vehicles," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1417–1429, 2023.
- [16] Y. Li and X. Lyu, "Convergence analysis of sequential federated learning on heterogeneous data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 1–55.
- [17] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4289–4301, Apr. 2023.

- [18] X. Li, W. Yang, S. Wang, and Z. Zhang, "Communication efficient decentralized training with multiple local updates," 2019, *arXiv:1910.09126*.
- [19] Y. Liao, Y. Xu, H. Xu, L. Wang, and C. Qian, "Adaptive configuration for heterogeneous participants in decentralized federated learning," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
- [20] 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.
- [21] G. Lan, X.-Y. Liu, Y. Zhang, and X. Wang, "Communication-efficient federated learning for resource-constrained edge devices," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 210–224, Aug. 2023, doi: [10.1109/TMLCN.2023.3309773](https://doi.org/10.1109/TMLCN.2023.3309773).
- [22] S. S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, "On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2041–2057, Feb. 2022.
- [23] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Robust federated learning for unreliable and resource-limited wireless networks," *IEEE Trans. Wireless Commun.*, early access, Feb. 23, 2024, doi: [10.1109/TWC.2024.3366393](https://doi.org/10.1109/TWC.2024.3366393).
- [24] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Communication-efficient federated learning with heterogeneous devices," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 3602–3607.
- [25] Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Adaptive model pruning for communication and computation efficient wireless federated learning," *IEEE Trans. Wireless Commun.*, early access, Dec. 20, 2023, doi: [10.1109/TWC.2023.3342626](https://doi.org/10.1109/TWC.2023.3342626).
- [26] Z. Chen, W. Yi, A. Nallanathan, and G. Y. Li, "Is partial model aggregation energy-efficient for federated learning enabled wireless networks?" in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 166–171.
- [27] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
- [28] S. Wang et al., "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 63–71.
- [29] 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.
- [30] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
- [31] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
- [32] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [33] Y. Xu, Y. Liao, H. Xu, Z. Ma, L. Wang, and J. Liu, "Adaptive control of local updating and model compression for efficient federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5675–5689, Oct. 2023.
- [34] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and E. Dutkiewicz, "Optimal online data partitioning for geo-distributed machine learning in edge of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2393–2406, Oct. 2019.
- [35] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.
- [36] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ., 2014.
- [37] 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. Stat.*, 2017, pp. 1273–1282.
- [38] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
- [39] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and X. Tao, "Distributed online learning of cooperative caching in edge cloud," *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2550–2562, Aug. 2021.
- [40] T. Tuor, S. Wang, K. K. Leung, and K. Chan, "Distributed machine learning in coalition environments: Overview of techniques," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, 2018, pp. 814–821.
- [41] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016.
- [42] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, 2004.
- [43] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," 2019, *arXiv:1907.09356*.
- [44] L. Kong, T. Lin, A. Koloskova, M. Jaggi, and S. Stich, "Consensus control for decentralized deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5686–5696.
- [45] 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.
- [46] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.
- [47] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3–4, pp. 231–357, 2015.
- [48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [50] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7252–7261.
- [51] X. Lian, C. Zhang, H. Zhang, C. J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
- [52] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Math. Program.*, vol. 180, no. 1–2, pp. 237–284, 2020.
- [53] G. Lan and Y. Zhou, "Asynchronous decentralized accelerated stochastic gradient descent," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 2, pp. 802–811, Jun. 2021.



QI CHEN received the B.E. degree from the Hebei University of Technology in 2022. He is currently pursuing the master's degree with the School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests include the edge intelligence and federated learning.



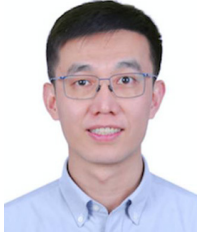
WEI YU received the B.S. degree from the Beijing University of Posts and Telecommunications in 1994, and the M.S. degree from the Beijing University of Posts and Telecommunications in 1997. He is currently the Director of the Business Research Institute, China Mobile Research Institute. His research interests include the circulation of data and big data systems and network intelligence.



XINCHEN LYU received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT) in 2014, and the dual Ph.D. degrees from BUPT and the University of Technology Sydney in 2019. He is currently an Associate Professor with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include the resource management and security of edge intelligence and its applications in future wireless networks.



ZIMENG JIA received the B.S. degree from the University of Electronic Science and Technology of China in 2017, and the M.S. degree from The University of Melbourne in 2020. She is currently a Junior Engineer with the Business Research Institute, China Mobile Research Institute. Her research interests include the circulation of data and big data systems and artificial intelligence.



GUOSHUN NAN (Member, IEEE) is a Tenure-Track Professor with the Beijing University of Posts and Telecommunications. He has broad interest in natural language processing, computer vision, machine learning, and wireless communications, such as information extraction, model robustness, multimodal retrieval, and next generation wireless networks. He is a member of the National Engineering Research Center for Mobile Network Technologies. He has published papers in top-tier conferences and journals including ACL,

CVPR, EMNLP, SIGIR, IJCAI, CKIM, SIGCOMM, IEEE NETWORK, *Computer Networks*, and *Journal of Network and Computer Applications*. He served as a Reviewer for ACL, EMNLP, AACL, IJCAI, *Neurocomputing*, and IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of ACL.



QIMEI CUI (Senior Member, IEEE) received the B.E. and M.S. degrees in electronic engineering from Hunan University, Changsha, China, in 2000 and 2003, respectively, and the Ph.D. degree in information and communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006, where she has been a Full Professor with the School of Information and Communication Engineering since 2014. She was a Visiting Professor with the Department of Electronic

Engineering, University of Notre Dame, Notre Dame, IN, USA, in 2016. Her research interests include B5G/6G wireless communications, mobile computing, and IoT. She won the Best Paper Award at the IEEE ISCT 2012, the IEEE WCNC 2014, the WCSP 2019, and the Honorable Mention Demo Award at the ACM MobiCom 2009, and the Young Scientist Award at the URSI GASS 2014. She serves as the Technical Program Chair of the APCC 2018, the Track Chair of IEEE/CIC ICC 2018, and the Workshop Chair of WPMC 2016. She also serves as a Technical Program Committee Member of several international conferences, such as the IEEE ICC, the IEEE WCNC, the IEEE PIMRC, the IEEE ICC, the WCSP 2013, and the IEEE ISCT 2012. She serves as an Editor for *Science China Information Science*, and a Guest Editor for the *EURASIP Journal on Wireless Communications and Networking*, *International Journal of Distributed Sensor Networks*, and *Journal of Computer Networks and Communications*.