POLITECNICO DI TORINO
Repository ISTITUZIONALE

Adaptive Quantization and Device Scheduling for Asynchronous Federated Learning over Resource-constrained Wireless Networks

Original
Adaptive Quantization and Device Scheduling for Asynchronous Federated Learning over Resource-constrained Wireless Networks / Li, X., Gao, Y., Liang, F., Chiasserini, C.F.. - In: IEEE TRANSACTIONS ON MOBILE COMPUTING. - ISSN 1536-1233. - (2026).

Availability:
This version is available at: 11583/3017948 since: 2026-11-15T14:39:58Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

Adaptive Quantization and Device Scheduling for Asynchronous Federated Learning over Resource-constrained Wireless Networks

Xiaodong Li, Yulong Gao, *Senior Member, IEEE*, Fengyu Liang, and Carla Fabiana Chiasserini, *Fellow, IEEE*

***Abstract*—Federated learning (FL) is a promising distributed learning paradigm that enables multiple devices to collaboratively train a global model without sharing local private data. Considering the device heterogeneity in computation and communication capabilities, asynchronous aggregation is introduced in FL to improve the learning efficiency. However, asynchronous FL suffers from communication bottlenecks due to limited wireless resources. To address this issue, we propose AQAFL, an adaptive quantization method for asynchronous FL, while also integrating device scheduling and wireless resource allocation. We first present a convergence analysis of asynchronous FL with quantized transmission, and then formulate an optimization problem to minimize the convergence upper bound under long-term latency and bandwidth constraints by jointly optimizing the quantization bit width, device scheduling, and wireless bandwidth allocation. Following that, we analytically characterize the optimal bandwidth allocation and prove that it can be obtained via bisection search, and derive the closed-form expression for the quantization bit width. Subsequently, an efficient device scheduling strategy is proposed to determine the number of participating devices in each round. Experiments show that compared with other methods, AQAFL reduces the total communication overhead by up to 69% and the overhead required to achieve the same target test accuracy by 32.6%–88.6%, while improving test accuracy by 4.1%–6.5%.**

***Index Terms*—Federated learning, asynchronous aggregation, quantization, communication efficiency.**

I. Introduction

With the advancement of the Internet of Things (IoT) and mobile communication technologies, an increasing number of edge devices (e.g., intelligent mobiles, micro servers, and small cells) are being connected to wireless networks and continuously generating massive data at the network edge [1]. Traditional machine learning (ML) follows a centralized training paradigm, where devices need to upload local data to the cloud center. However, collecting all data to the cloud is often impractical due to data privacy concerns and high communication costs. To address these issues, federated learning (FL) [2] has emerged as a distributed learning paradigm that enables multiple devices to collaboratively train ML models in a privacy-preserving manner. Instead of sharing local data, FL exchanges model parameters or updates between the server and devices for model updating, thereby offering improved privacy protection and reduced communication load. Nowadays, FL has shown promising results in various applications, such as edge computing [3], intelligent transportation [4], and wireless networks [5].

Conventional FL adopts synchronous aggregation [2], where the server aggregates local updates from all participating devices in each round. Due to device heterogeneity in computation and communication capabilities, this synchronous manner incurs the straggler issue, i.e., the server has to wait for local updates of the slow devices. This will slow down model convergence, damage training efficiency, and lead to resource under-utilization of non-stragglers. To address the straggler issue, some works investigate partial model updates [6], device selection [7], [8], or resource allocation [9], [10] in synchronous FL. Nevertheless, such strategies still require the server to wait for the selected devices to complete their local updates before performing aggregation in each round, thereby limiting training efficiency [11]. To overcome these limitations, asynchronous FL is proposed to enable asynchronous model aggregation without waiting for stragglers to complete their local computation [12]–[14]. Stragglers can continue their training to prepare for later aggregation. This way effectively reduces the single-round wall-clock time and improves the resource utilization of devices [15]. However, when deploying asynchronous FL over wireless networks, the limited wireless resources make the transmission of model updates costly and can lead to communication bottlenecks between devices and the server. Moreover, dynamic wireless conditions further affect the transmission efficiency. Therefore, it is essential to develop efficient asynchronous FL schemes for resource-constrained wireless networks.

A. Related Works

A straightforward approach to asynchronous FL is to update the global model once the server receives one local update [12], which is called fully asynchronous FL. Nevertheless, this method is often not scalable and may degrade model performance due to unstable updates induced by severe asynchrony [13]. To address this issue, a commonly adopted strategy is to buffer and aggregate a certain number of local updates in each

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62171163.

Xiaodong Li, Yulong Gao, and Fengyu Liang are with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: lixd@stu.hit.edu.cn; ylgao@hit.edu.cn; lfy1508880114@gmail.com).

Carla Fabiana Chiasserini is with the Politecnico di Torino and the CNR-IEIIT, Torino, Italy, with the CNIT, Parma, Italy, and with the Chalmers University of Technology, Sweden (e-mail: carla.chiasserini@polito.it).

0000–0000/00$00.00 © 2021 IEEE

round [14]. Under this framework, fully asynchronous FL can be regarded as a special case with a buffer size of one.

To improve the communication efficiency of asynchronous FL under limited wireless resources, many methods have been proposed for device scheduling and wireless resource allocation [16]–[18]. For instance, the authors of [16] minimized training time under a given communication budget by optimizing the number of participating devices in each round. The work in [17] developed an online device selection strategy for asynchronous FL to minimize training latency under limited resources. Moreover, [18] minimized the convergence bound of asynchronous FL under long-term latency, thereby optimizing the number of participating devices and bandwidth allocation in each round. However, they do not fundamentally reduce the payload size of transmitted model updates, and thus communication efficiency remains limited.

To further reduce communication overhead, some studies apply federated dropout or model sparsification techniques to decrease the number of transmitted parameters [19]–[21]. The authors of [19] presented a FedDrop scheme, where the server generates subnets from the global model for devices using heterogeneous dropout rates according to their channel conditions. In [20], the authors proposed FedLoDrop, which jointly optimizes the dropout rate and resource allocation under latency and per-device energy consumption constraints to minimize the upper bound of the generalization error. [21] employed Top-K sparsification to reduce communication cost and investigated the relationship between sparsity ratio and the efficiency of asynchronous FL.

Nevertheless, in practical systems, model parameters or updates need to be quantized before transmission. All the above methods overlook the optimization of quantization schemes and adopt a fixed quantization bit width for all devices, which limits communication efficiency under resource constraints and fails to adapt to varying wireless conditions. Note that the quantization here is different from that in ML model compression, which focuses on reducing model storage and simplifying inference computation after the training is completed [22]. In contrast, the former aims to improve communication efficiency and has to be carried out in every round. Recently, several works have explored quantization schemes for asynchronous FL. [23] and [24] adjusted quantization strategies to reduce communication overhead and improve convergence behavior. Similarly, [25] introduced a hidden-state quantization scheme in asynchronous FL to mitigate quantization errors and reduce communication cost. However, none of these works considers the optimization of the quantization bit width in asynchronous FL. Instead, they mainly focus on the application of quantization schemes and adjust the bit width empirically. Moreover, they do not incorporate device scheduling or wireless resource allocation to further enhance asynchronous FL efficiency.

In contrast to the limited exploration of quantization optimization in asynchronous FL, some studies have optimized the quantization bit width for synchronous FL [26]–[28]. However, their analyses and optimization designs rely on the round-synchronous aggregation mechanism and are thus unsuitable for direct extension to asynchronous FL. By contrast, in asynchronous FL, stale local updates and cross-round propagation of quantization errors make the analysis and optimization substantially more complex. Therefore, the quantization optimization in asynchronous FL remains insufficiently explored.

Unlike prior asynchronous FL methods that incorporate quantization schemes [23]–[25], we consider the optimization of quantization bit width and jointly optimize device scheduling and wireless resource allocation from a long-term perspective to improve the efficiency of asynchronous FL. Moreover, federated dropout or model sparsification techniques are complementary to our method, since they reduce the number of model parameters, whereas our method focuses on optimizing the quantization bit width for transmission.

B. Motivations and Contributions

From the above analysis, the optimization of quantization bit width in asynchronous FL remains insufficient. It is still unclear how the quantization bit width in each round affects model performance. Therefore, theoretical analysis is needed to analyze its impact on convergence and guide the design of optimization strategies. Moreover, device scheduling and wireless resource allocation should also be jointly considered to further improve the efficiency of asynchronous FL. Motivated by the above, we first perform the convergence analysis of asynchronous FL with quantized model updates. Then, we propose AQAFL, an adaptive joint optimization method for asynchronous FL that determines quantization bit width, number of participating devices, and wireless bandwidth allocation in each round under long-term latency and bandwidth constraints. Our main contributions are summarized as follows.

- To improve the efficiency of asynchronous FL under limited wireless resources, we propose AQAFL with adaptive quantization. To the best of our knowledge, this is the first work to optimize the quantization bit width in asynchronous FL.
- We conduct the convergence analysis of asynchronous FL with adaptive quantization, and theoretically demonstrate the impact of the quantization errors induced in each round. Then, the quantization bit width, device scheduling, and bandwidth allocation are jointly optimized to minimize the convergence upper bound under long-term latency and bandwidth constraints.
- To solve the long-term optimization problem, we reformulate it into an online form leveraging the Lyapunov optimization framework. Then, by introducing a latency control variable, we analytically characterize the optimal bandwidth allocation and prove that it can be obtained via bisection search, and derive the closed-form expression for the quantization bit width. Subsequently, we propose an efficient device scheduling strategy to determine the number of participating devices in each round.
- Extensive experiments on two benchmark datasets show that compared with other state-of-the-art methods, our AQAFL reduces the communication overhead by up to 69% under the same training time, and decreases the overhead required to achieve the same target test accuracy by 32.6%–88.6%. Meanwhile, AQAFL achieves a 4.1%–6.5% improvement in test accuracy.

TABLE I
MAIN NOTATIONS

Notation	Definition
K	Total number of devices
T	Number of communication rounds
$\boldsymbol{w}_t$	Global model in round t
$\boldsymbol{w}_{t,r}^k$	Local model of device k in r-th local iteration
$F_k(\boldsymbol{w}_{t,r}^k)$	Local objective function of device k
Δ_t^k	Local updates of device k in round t
s_t^k	Staleness of device k in round t
q_t^k	Quantization bit width of device k in round t
l_t^k	Wireless bandwidth allocation of device k in round t
K_t	Number of participating devices in round t
$\mathcal{K}_t$	Set of K_t devices the server collects updates from
J	Number of model weights
B	Total available wireless bandwidth in the FL system
$T^{\max}$	Long-term latency constraint

The rest of this paper is organized as follows. Section II introduces the system model. Section III presents the convergence analysis and problem formulation. Section IV proposes our AQAFL. Section V shows the experimental results, and Section VI presents the conclusion.

II. SYSTEM MODEL

Consider a wireless FL system consisting of a server and K devices. Each device $k \in \{1, \ldots, K\}$ holds a local dataset $\mathcal{D}_k = \{(\boldsymbol{x}_i^k, y_i^k)\}_{i=1}^{|\mathcal{D}_k|}$, where $\boldsymbol{x}_i^k$ denotes the sample features and y_i^k represents the corresponding label. The goal of FL is to collaboratively learn a global model $\boldsymbol{w}$ through iterative communication between the server and the devices, so as to minimize the following objective function:

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\boldsymbol{w}), \tag{1}$$

where $F_k(\boldsymbol{w})$ and $F(\boldsymbol{w})$ denote the local and global objectives, respectively. For ease of reference, the main notations in this paper are listed in Table I.

A. Asynchronous Federated Learning

In conventional asynchronous FL [18], [25], each device performs local training and participates in global update asynchronously. Devices that are not involved in the current round will continue their local training to prepare for later aggregation. Let T denote the total number of communication rounds. At the beginning of FL, the server initializes the global model and broadcasts it to all devices. Then, the detailed processes in each round t are described as follows.

1) Device Side: After receiving the global model $\boldsymbol{w}_t$, each device k initializes local model $\boldsymbol{w}_{t,0}^k$ with it and performs τ steps of stochastic gradient descent (SGD) as follows:

$$\boldsymbol{w}_{t,r+1}^k = \boldsymbol{w}_{t,r}^k - \eta \nabla F_k(\boldsymbol{w}_{t,r}^k, \xi_{t,r}^k), \tag{2}$$

where $\boldsymbol{w}_{t,r}^k$ denotes the local model in r-th iteration, η is the learning rate, and $\nabla F_k(\boldsymbol{w}_{t,r}^k, \xi_{t,r}^k)$ represents the local gradient on a random mini-batch $\xi_{t,r}^k$ of local data. Subsequently, the local update Δ_t^k of device k can be written as

$$\Delta_t^k = \boldsymbol{w}_{t,\tau}^k - \boldsymbol{w}_{t,0}^k. \tag{3}$$

Then, each device k reports its state to the server and requests wireless resource allocation to upload its local update Δ_t^k.

2) Server Side: The server determines the number of devices K_t to participate in round t, and waits to collect updates from the first K_t devices that complete local computation and become ready for uploading [18]. Denote by $\mathcal{K}_t$ the set of these devices, where $|\mathcal{K}_t| = K_t$. Since devices participate in global updates asynchronously, the local update of a device is not necessarily derived from the most recent global model. Let s_t^k denote the staleness of device k in round t, defined as the interval between the current round t and the round when device k received the last global model. Thus, the local update Δ_t^k of device k is actually computed based on an older version of the global model $\boldsymbol{w}_{t-s_t^k}$, i.e.,

$$\Delta_t^k = -\eta \sum_{r=0}^{\tau-1} \nabla F_k(\boldsymbol{w}_{t-s_t^k,r}^k, \xi_{t,r}^k), \tag{4}$$

Upon receiving K_t local updates from $\mathcal{K}_t$, the server performs model aggregation as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \lambda \frac{1}{|\mathcal{K}_t|} \sum_{k \in \mathcal{K}_t} \Delta_t^k, \tag{5}$$

where $\lambda \in (0, 1]$ is the global step size. Following [15], [29], we set $\lambda = 1$ by default. Then, $\boldsymbol{w}_{t+1}$ is broadcast to the devices in $\mathcal{K}_t$ for local training. We adopt the average scheme in (5), which is commonly used in asynchronous FL [25], [30]. The subsequent analysis and algorithms can be easily extended to scenarios with weighted aggregation.

B. Model Quantization

We mainly focus on the optimization of uplink quantization bit width, since the uplink transmission from devices dominates the communication bottleneck in asynchronous FL [25]. As shown in [26], transmitting the model update Δ_t^k instead of the model itself is more bit-efficient, which is also employed in this paper. Thus, each device k quantizes its local update Δ_t^k before uploading and transmits the quantized version to the server. We adopt the stochastic quantization method in [22], which is designed for practical FL systems and has been widely used and proven effective [27]. Following that, the local update of each model weight j, denoted by $\Delta_t^{k,j}$, satisfies $|\Delta_t^{k,j}| \in [\underline{\Delta}_{t,k}, \overline{\Delta}_{t,k}]$, where $\underline{\Delta}_{t,k}$ and $\overline{\Delta}_{t,k}$ are the lower and upper bounds of the quantization range, respectively. Let q_t^k denote the quantization bit width of device k in round t. Then, the interval $[\underline{\Delta}_{t,k}, \overline{\Delta}_{t,k}]$ of device k can be divided into ζ sub-intervals $[c_0, c_1], [c_1, c_2], \ldots, [c_{\zeta-1}, c_\zeta]$, where $\zeta = 2^{q_t^k} - 1$ and

$$c_i = \underline{\Delta}_{t,k} + i \frac{\overline{\Delta}_{t,k} - \underline{\Delta}_{t,k}}{2^{q_t^k} - 1}, \quad i = 0, \ldots, \zeta. \tag{6}$$

If the local update $\Delta_t^{k,j}$ of weight j falls in $[c_i, c_{i+1}]$, it will be quantized as

$$\mathcal{Q}(\Delta_t^{k,j}) = \begin{cases} \text{sign}(\Delta_t^{k,j}) \cdot c_i, & \text{w.p. } \dfrac{c_{i+1} - |\Delta_t^{k,j}|}{c_{i+1} - c_i}, \\ \text{sign}(\Delta_t^{k,j}) \cdot c_{i+1}, & \text{w.p. } \dfrac{|\Delta_t^{k,j}| - c_i}{c_{i+1} - c_i}, \end{cases} \tag{7}$$

where 'w.p.' stands for 'with probability'. Then, the quantized local update is denoted as $\mathcal{Q}(\Delta_t^k) = [\mathcal{Q}(\Delta_t^{k,1}), \ldots, \mathcal{Q}(\Delta_t^{k,J})]$, where J is the total number of model weights. Meanwhile, the bit size for each device k to upload $\mathcal{Q}(\Delta_t^k)$ in round t is

$$u_t^k = J(q_t^k + 1) + z, \tag{8}$$

where the $+1$ accounts for the sign bit, and z represents the bits for the values of $\underline{\Delta}_{t,k}$ and $\overline{\Delta}_{t,k}$. At this point, the model aggregation (5) at the server can be rewritten as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \lambda \frac{1}{|\mathcal{K}_t|} \sum_{k \in \mathcal{K}_t} \mathcal{Q}(\Delta_t^k). \tag{9}$$

C. Computation and Communication Latency

As mentioned earlier, we focus on mitigating the communication bottleneck caused by the uplink transmission from the device side. Following [26], [31], we assume that the downlink latency is approximately a small constant and can be neglected compared to that of the uplink, enabled by the ability to vary the data rate of the server. Moreover, since the server is typically more computationally powerful than devices, its computation latency is ignored [27]. Let f_k denote the CPU frequency allocated for local training on device k, b_k represent the local mini-batch size, and c_k be the number of CPU cycles required to process one sample. Then, the local computation time of device k can be expressed as

$$T_{t,k}^{\text{cmp}} = \frac{\tau c_k b_k}{f_k}. \tag{10}$$

We consider the orthogonal frequency-division multiple access (OFDMA) scheme for uplink transmission [32], and the achievable transmission rate of device k is

$$R_t^k = l_t^k B \log_2(1 + \frac{h_t^k p_t^k}{l_t^k B N_0}), \tag{11}$$

where B is the total bandwidth, l_t^k denotes the bandwidth fraction allocated to device k, h_t^k is the uplink channel gain, p_t^k is the transmit power, and N_0 is the noise power spectral density. The control message latency for state reporting and resource requests is ignored [18], [33], since it is lightweight and negligible compared with that of model updates. Then, the communication time of device k can be expressed as

$$T_{t,k}^{\text{com}} = \frac{u_t^k}{R_t^k} = \frac{J(q_t^k + 1) + z}{R_t^k}. \tag{12}$$

Based on the above models, we characterize the per-round latency for each general round t. Due to the asynchrony of FL, devices may have completed part of their local training at the start of each round t. Thus, the computation time of device k usually differs from $T_{t,k}^{\text{cmp}}$. Let $T_{t,k}^{\text{rem}}$ denote the remaining computation time of device k in round t. Then, the total latency of device $k \in \mathcal{K}_t$ is

$$T_{t,k}^{\text{total}} = T_{t,k}^{\text{rem}} + T_{t,k}^{\text{com}}. \tag{13}$$

Therefore, the overall latency of FL in round t is determined by the slowest device, which can be expressed as

$$T_t = \max_{k \in \mathcal{K}_t} \{T_{t,k}^{\text{total}}\}. \tag{14}$$

III. Convergence Analysis and Problem Formulation

In this section, we first conduct the convergence analysis of asynchronous FL with quantized transmission, and then formulate an optimization problem to minimize the convergence bound under long-term latency and bandwidth constraints.

A. Convergence Analysis

Before the analysis, we make the following assumptions that are commonly used in the analysis of FL [34].

Assumption 1 (L-smoothness)**.** *The objective function $F(\boldsymbol{w})$ is L-smooth: there exists a constant $L > 0$ such that $\|\nabla F(\boldsymbol{w}_1) - \nabla F(\boldsymbol{w}_2)\| \leq L \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|$ for any $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$.*

Assumption 2 (Unbiased Gradient and Bounded Variance)**.** *Let $\xi_{t,r}^k$ be a random mini-batch data sampled from $\mathcal{D}_k$. For each device k, the stochastic gradient is unbiased, i.e., $\mathbb{E}\big[\nabla F_k(\hat{\boldsymbol{w}}_{t,r}^k, \xi_{t,r}^k)\big] = \nabla F_k(\hat{\boldsymbol{w}}_{t,j}^k)$, and has bounded variance: $\mathbb{E}\big[\|\nabla F_k(\hat{\boldsymbol{w}}_{t,r}^k, \xi_{t,r}^k) - \nabla F_k(\hat{\boldsymbol{w}}_{t,r}^k)\|^2\big] \leq \sigma^2$ for $\forall t$ and $\forall r$.*

Assumption 3 (Bounded Gradient)**.** *The expected squared norm of stochastic gradient is bounded by a constant G^2, i.e., $\mathbb{E}_{\xi_{t,r}^k \sim \mathcal{D}_k}\big[\|\nabla F_k(\hat{\boldsymbol{w}}_{t,j}^k, \xi_{t,r}^k)\|^2\big] \leq G^2$.*

In asynchronous FL, the local updates of different devices are typically computed based on different versions of the global model, resulting in varying staleness among devices. To facilitate the analysis, the following bounded staleness assumption similar to that in [14], [21], [25] is also introduced.

Assumption 4 (Bounded Staleness)**.** *There exists a maximum staleness $s_{\max}$ such that $s_t^k \leq s_{\max}$ for $\forall t$ and $\forall k$.*

This is reasonable, since the staleness is often limited by system-level constraints such as stale-update discarding mechanisms and finite aggregation windows [16], [35]. Considering that the loss function may be non-convex, we do not impose convexity assumptions and characterize the convergence performance using the average squared gradient norm, which is a standard stationarity measure for convex and non-convex objectives [36], [37]. The convergence bound of asynchronous FL with quantized transmission is presented as follows.

Theorem 1. *Under the above assumptions, let the learning rate $\eta \leq \frac{1}{2\tau L}$, then the convergence bound is*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\boldsymbol{w}_t)\|^2 \leq{}& \frac{2(F_0 - F_*)}{T\eta\tau\lambda} + 2\eta\tau L\sigma^2 \\ &+ 2\eta^2\tau^2 L^2(\sigma^2 + G^2)(2s_{\max}^2 + 1) \\ &+ \frac{JL}{4T\eta\tau}\sum_{t=1}^{T}(1 + \beta m_t)C_t,\end{aligned} \tag{15}$$

where

$$C_t = \frac{1}{|\mathcal{K}_t|}\sum_{k \in \mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k} - 1)^2}, \tag{16}$$

$\beta = 2L\eta\tau s_{\max}$, $m_t = \min(s_{\max}, T - t)$, $\delta_{t,k} = \overline{\Delta}_{t,k} - \underline{\Delta}_{t,k}$, $F_0 = F(\boldsymbol{w}_0)$, and $F_* = F(\boldsymbol{w}_*)$ is the theoretical optimum.

Proof: See Appendix A in the supplementary material.

In Theorem 1, C_t denotes the quantization error induced in round t, and m_t represents the maximum number of subsequent rounds that C_t can affect, with the influence strength controlled by β. This indicates that in asynchronous scenarios, the quantization error C_t not only affects the current round but also propagates its impact to subsequent rounds. Moreover, Theorem 1 shows that quantization introduces an error term of $\frac{JL}{4T\eta\tau}\sum_{t=1}^{T}(1+\beta m_t)C_t$ into the convergence bound, thereby affecting the attainable stationarity level.

B. Problem Formulation

Our objective is to obtain the optimal number of participating devices K_t, quantization bit width q_t^k, and bandwidth allocation l_t^k for each round t and device k to improve the efficiency of asynchronous FL while preserving model performance. This requires minimizing the objective function in (1) under latency and bandwidth constraints. However, directly analyzing its minimization before the training process is difficult. Theoretically, a tighter upper bound corresponds to a better model performance, which has been proven and widely used in ML optimization [8], [38], [39]. Therefore, we turn to minimizing the convergence bound in Theorem 1. Moreover, as discussed earlier, decisions across rounds in asynchronous FL are coupled, which jointly influence the convergence performance. Thus, the optimization should be conducted from a long-term perspective. According to Theorem 1, $m_t \le s_{\max}$ and takes the value of $s_{\max}$ for most rounds except for the last $T-t$ rounds. This implies that the quantization error incurred in each round affects up to $s_{\max}$ subsequent rounds in most cases. In practical asynchronous FL, the total number of training rounds T is often unspecified in advance, and instead, training proceeds until a given time budget is exhausted [17], [40]. In this case, m_t is not available due to the unspecified T. Therefore, we substitute $m_t \le s_{\max}$ into (15) to yield a tractable surrogate of the convergence bound while retaining the essential effect of quantization errors in asynchronous FL. Then, the last term on the right-hand side of (15) becomes $\frac{JL(1+2L\eta\tau s_{\max}^2)}{4T\eta\tau}\sum_{t=1}^{T}\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2}$, where the coefficient characterizes the per-round impact of quantization errors. Accordingly, the following long-term optimization problem is formulated to minimize the surrogate convergence bound.

$$\mathbf{P1}: \min_{K_t,q_t^k,l_t^k} \sum_{t=1}^{T}\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2}, \tag{17}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} T_t \le T^{\max}, \tag{17a}$$

$$\sum_{k\in\mathcal{K}_t} l_t^k \le 1, \quad \forall t, \tag{17b}$$

$$0 \le l_t^k \le 1, \quad \forall k,\forall t, \tag{17c}$$

$$0 \le K_t \le K, \quad K_t \in \mathbb{Z}^+, \forall t, \tag{17d}$$

$$q_t^k \in \mathbb{Z}^+, \ q_t^k \le q_{t,k}^{\max}, \quad \forall k,\forall t, \tag{17e}$$

where $T^{\max}$ in (17a) denotes the long-term latency constraint, (17b) and (17c) are the wireless resource constraints, (17d) and (17e) specify the limits on K_t and q_t^k, and $q_{t,k}^{\max}$ in (17e) is the maximum allowed quantization bit width for device k.

IV. Adaptive Device Scheduling, Quantization Bit Width, and Bandwidth Allocation

In this section, we first reformulate $\mathbf{P1}$ into an online problem, and then decompose it into two sub-problems: one for the quantization bit width and bandwidth allocation, and the other for the number of participating devices in each round.

A. Problem Analysis

Solving $\mathbf{P1}$ offline requires full knowledge of the wireless conditions across all rounds, which is infeasible in practice due to the dynamic nature of the wireless environments. Therefore, we adopt the drift-plus-penalty framework from Lyapunov optimization [41] to reformulate $\mathbf{P1}$ into an online problem, enabling decisions based on the current system states of each round. To deal with the long-term latency constraint, we introduce a virtual latency queue that evolves as

$$d_{t+1} = \max\{d_t + T_t - \overline{T}, 0\} \tag{18}$$

with an initial value $d_1 = 0$, where $\overline{T} = \frac{T^{\max}}{T}$ is the long-term average latency budget per round. Then, leveraging this queue structure and (14), the original problem $\mathbf{P1}$ is reformulated into the following online form:

$$\mathbf{P2}: \min_{K_t,q_t^k,l_t^k} V\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2} + d_t\max_{k\in\mathcal{K}_t}\{T_{t,k}^{rem} + \frac{J(q_t^k+1)+z}{R_t^k}\}, \tag{19}$$

$$\text{s.t.} \quad \text{(17b)}, \text{(17c)}, \text{(17d)}, \text{(17e)}, \tag{19a}$$

where V is the Lyapunov weight that controls the trade-off between quantization error and latency performance.

We denote $\Phi(t) = \frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2}$, so that the objective (17) of $\mathbf{P1}$ can be written as minimizing $\sum_{t=1}^{T}\Phi(t)$. Let the superscripts $\dagger$ and $*$ denote the optimal solutions of the original problem $\mathbf{P1}$ and the online problem $\mathbf{P2}$, respectively. Then, their performance gap is presented as follows.

Lemma 1. *With the virtual latency queue initialized to zero, the gap between the optimal value of* $\mathbf{P1}$ *and that of the optimal online solution to* $\mathbf{P2}$ *is bounded by*

$$\sum_{t=1}^{T}\left[\Phi^*(t) - \Phi^\dagger(t)\right] \le \frac{\Theta T^2}{V}, \tag{20}$$

and the cumulative latency is bounded by

$$\sum_{t=1}^{T} T_t \le T^{\max} + \sqrt{2\Theta T^2 + 2V\sum_{t=1}^{T}\Phi^\dagger(t)}, \tag{21}$$

where $\Theta \triangleq \frac{1}{2}\theta^2$ *and* $\theta \triangleq \max_t\{|T_t - \overline{T}|\}$.

Proof: See Appendix B in the supplementary material.

Lemma 1 indicates that the performance gap between the optimal solutions of $\mathbf{P1}$ and $\mathbf{P2}$ is no more than $\frac{\Theta T^2}{V}$, and total latency remains acceptable and controlled. Therefore, we can alternatively solve $\mathbf{P2}$ instead of directly tackling $\mathbf{P1}$. Moreover, a larger V can reduce the optimality gap and bring the solution of $\mathbf{P2}$ closer to that of $\mathbf{P1}$, but it also impairs the

stability of the latency queue. Thus, V controls the trade-off between convergence performance and latency stability.

However, the second term in the objective of $\mathbf{P2}$ involves a max operator, making its optimization still challenging. To address this issue, we apply the epigraph reformulation method [42] by introducing an auxiliary latency control variable γ_t to upper-bound the latency of each round t, thereby converting the non-smooth max operator into epigraph constraints. Thus, $\mathbf{P2}$ can be equivalently rewritten as

$$\mathbf{P3}: \min_{K_t, q_t^k, l_t^k, \gamma_t} V\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2} + d_t\gamma_t, \tag{22}$$

$$\text{s.t.} \quad T_{t,k}^{rem} + \frac{J(q_t^k+1)+z}{R_t^k} \le \gamma_t, \ \forall k \in \mathcal{K}_t, \tag{22a}$$

$$(17b), (17c), (17d), (17e). \tag{22b}$$

Note that the optimal value γ_t^* of γ_t satisfies $\gamma_t^* = T_t$. At this point, we need to optimize K_t, q_t^k, and l_t^k with latency control γ_t. This reformulation eliminates the non-smoothness caused by the max operator.

Problem $\mathbf{P3}$ is a mixed-integer nonlinear programming (MINLP) problem, which is non-convex and NP-hard [43], making it difficult to solve directly. Therefore, we decouple it into two sub-problems: one for the quantization bit width and bandwidth allocation with latency control, and the other for the number of participating devices.

B. Optimal Quantization Bit Width and Bandwidth Allocation with Latency Control

As described in Section II, given the number of participating devices K_t in round t, the selected device set $\mathcal{K}_t$ can be determined, which is the first K_t devices that complete their local training. However, this sub-problem is still a MINLP problem. Since q_t^k is an integer variable and difficult to optimize directly, we first relax it to a continuous variable satisfying $1 \le q_t^k \le q_{t,k}^{\max}$. Then, we solve this sub-problem through alternating optimization: optimize l_t^k and γ_t under given q_t^k, and optimize q_t^k and γ_t under fixed l_t^k. It is worth noting that γ_t is optimized in each sub-step, which is necessary for two main reasons. First, q_t^k and γ_t must be optimized simultaneously. This is because q_t^k and γ_t affect two individual terms in (22), respectively, and alternately fixing them would disregard the influence of the coefficients $\frac{V}{|\mathcal{K}_t|}$ and d_t on the solution. Second, l_t^k cannot be optimized independently, since if both q_t^k and γ_t are fixed, the objective function in (22) no longer depends on l_t^k, making the optimization meaningless.

Before deriving the solution, we first present the following proposition, which characterizes the necessary conditions satisfied by the optimal q_t^k, l_t^k, and γ_t.

Proposition 1. *Given $\mathcal{K}_t$, the optimal q_t^k, l_t^k, and γ_t to $\mathbf{P3}$ must make the constraints* (22a) *and* (17b) *be tight, i.e.,*

$$T_{t,k}^{rem} + \frac{J(q_t^k+1)+z}{R_t^k} = \gamma_t, \ \forall k \in \mathcal{K}_t, \tag{23}$$

$$\sum_{k\in\mathcal{K}_t} l_t^k = 1. \tag{24}$$

Proof: The proposition is proved by contradiction. Suppose that there exists an optimal solution such that (22a) *is not tight, we can then: 1) keep the quantization bit width unchanged and reallocate the bandwidth of devices that complete uploading ahead of others to the slower devices, thereby decreasing the latency term γ_t in* (22)*; or 2) maintain γ_t fixed and increase quantization bit width of those devices for which the constraint is slack, which reduces the first term in* (22)*. Both can yield better solutions, which contradicts the optimality assumption. For constraint* (17b)*, if an optimal solution makes it slack, we can redistribute the remaining bandwidth to further reduce γ_t, which can result in a better solution. Proposition 1 is proved.*

Therefore, the optimal solutions of the two alternating sub-steps must also satisfy this proposition. In fact, for each sub-step, the same tightness property can be proved by a similar contradiction argument.

1) Optimal Bandwidth Allocation with Latency Control: When optimizing l_t^k and γ_t under given q_t^k, problem $\mathbf{P3}$ can be transformed into

$$\mathbf{P4}: \min_{l_t^k, \gamma_t} \ \gamma_t, \tag{25}$$

$$\text{s.t.} \quad (22a), (17b), (17c). \tag{25a}$$

We first prove that the problem $\mathbf{P4}$ is convex through the following Lemma.

Lemma 2. *Given q_t^k of each device k, the problem $\mathbf{P4}$ with respect to l_t^k and γ_t is convex.*

Proof: See Appendix C in the supplementary material.

Subsequently, based on Proposition 1, the optimal bandwidth allocation is given by the following theorem.

Theorem 2. *Given the quantization bit width q_t^k, the optimal bandwidth allocation $l_{t,k}^*$ for each device $k \in \mathcal{K}_t$ is*

$$l_{t,k}^* = \frac{-u_t^k \ln 2}{B\left[\Psi_t^k + W_{-1}\left(-\Psi_t^k e^{-\Psi_t^k}\right)\right](\gamma_t^* - T_{t,k}^{rem})}, \tag{26}$$

where

$$\Psi_t^k = \frac{u_t^k N_0 \ln 2}{h_t^k p_t^k (\gamma_t^* - T_{t,k}^{rem})}, \tag{27}$$

$W_{-1}(\cdot)$ is the W_{-1} branch of the Lambert-W function [44], and satisfy $\sum_{k\in\mathcal{K}_t} l_{t,k}^* = 1$.

Proof: See Appendix D in the supplementary material.

According to Theorem 2, the bandwidth allocation depends on the latency control variable γ_t. To determine optimal l_t^k and γ_t, we further provide the following Lemma.

Lemma 3. *The optimal bandwidth allocation $l_{t,k}^*$ is a monotonically decreasing function of γ_t^*.*

Proof: See Appendix E in the supplementary material.

Therefore, bisection search can be employed to find the optimal l_t^k and γ_t under the given quantization bit width. To facilitate this, we first derive the feasible lower and upper bounds of γ_t, denoted by $\underline{\gamma}_t^{(P4)}$ and $\overline{\gamma}_t^{(P4)}$, respectively. Obviously, the lower bound $\underline{\gamma}_t^{(P4)}$ corresponds to the minimum achievable latency, i.e., the maximum latency among devices when each

device occupies the entire bandwidth B individually, which is expressed as

$$\underline{\gamma}_t^{(P4)} = \max_{k\in\mathcal{K}_t}\left\{T_{t,k}^{rem} + \frac{J(q_t^k+1)+z}{B\log_2(1+\frac{p_t^k h_t^k}{BN_0})}\right\}. \tag{28}$$

The upper bound $\overline{\gamma}_t^{(P4)}$ is obtained by uniformly allocating bandwidth $l_t^k = \frac{1}{|\mathcal{K}_t|}$ to each device $k \in \mathcal{K}_t$, which yields

$$\overline{\gamma}_t^{(P4)} = \max_{k\in\mathcal{K}_t}\left\{T_{t,k}^{rem} + \frac{J(q_t^k+1)+z}{\frac{B}{|\mathcal{K}_t|}\log_2(1+\frac{|\mathcal{K}_t|p_t^k h_t^k}{BN_0})}\right\}. \tag{29}$$

Since $\overline{\gamma}_t^{(P4)}$ is within the feasible region of $\mathbf{P4}$, it naturally follows that $\gamma_t^* \le \overline{\gamma}_t^{(P4)}$. Based on $\underline{\gamma}_t^{(P4)}$ and $\overline{\gamma}_t^{(P4)}$, bisection search can be applied to find the optimal γ_t and determine the corresponding bandwidth allocation l_t^k, which iteratively halves the search region until the tolerance ϵ_0 is met.

2) Optimal Quantization Bit Width with Latency Control: Given l_t^k for each device k, problem $\mathbf{P3}$ becomes

$$\mathbf{P5}: \min_{q_t^k,\gamma_t} V\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_t^k}-1)^2} + d_t\gamma_t, \tag{30}$$

$$\text{s.t. (22a), (17e).} \tag{30a}$$

Based on (23) in Proposition 1, q_t^k can be derived as

$$q_t^k = \frac{\left(\gamma_t - T_{t,k}^{\text{rem}}\right)R_t^k - z}{J} - 1 \triangleq q_k(\gamma_t). \tag{31}$$

Then, by substituting (31), problem $\mathbf{P5}$ is transformed into

$$\mathbf{P6}: \min_{\gamma_t} V\frac{1}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2}{(2^{q_k(\gamma_t)}-1)^2} + d_t\gamma_t. \tag{32}$$

Meanwhile, according to (23), γ_t is monotonically increasing with q_t^k. Thus, by substituting the boundary points 1 and $q_{t,k}^{\max}$ of q_t^k into (23), the minimum and maximum feasible values of γ_t in problem $\mathbf{P6}$ are given respectively by

$$\underline{\gamma}_t^{(P6)} = \max_{k\in\mathcal{K}_t}\left\{T_{t,k}^{rem} + \frac{2J+z}{R_t^k}\right\}, \tag{33}$$

$$\overline{\gamma}_t^{(P6)} = \max_{k\in\mathcal{K}_t}\left\{T_{t,k}^{rem} + \frac{J(q_{t,k}^{\max}+1)+z}{R_t^k}\right\}. \tag{34}$$

It is worth noting that $\underline{\gamma}_t^{(P6)}$ and $\overline{\gamma}_t^{(P6)}$ specify the feasible interval of γ_t for $\mathbf{P6}$, which are distinct from $\underline{\gamma}_t^{(P4)}$ and $\overline{\gamma}_t^{(P4)}$ that serve as the search bounds for $\mathbf{P4}$. At this point, $\mathbf{P6}$ becomes an unconstrained problem over $\gamma_t \in [\underline{\gamma}_t^{(P6)}, \overline{\gamma}_t^{(P6)}]$.

Lemma 4. *The problem $\mathbf{P6}$ with respect to γ_t is convex.*

Proof: See Appendix F in the supplementary material.

Thus, the optimal γ_t lies either at one of the boundary points $\underline{\gamma}_t^{(P6)}$ and $\overline{\gamma}_t^{(P6)}$, or at a stationary point within the feasible interval. We denote by $g(\gamma_t)$ the first-order derivative of (32) with respect to γ_t, which can be expressed as

$$g(\gamma_t) = d_t - \frac{2V\ln 2}{J|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\frac{\delta_{t,k}^2 2^{q_k(\gamma_t)}R_t^k}{\left(2^{q_k(\gamma_t)}-1\right)^3}. \tag{35}$$

Since $q_k(\gamma_t)$ increases with γ_t and the second term in $g(\gamma_t)$ decreases with $q_k(\gamma_t)$, $g(\gamma_t)$ monotonically increases with γ_t. Therefore, according to the monotonicity, the optimal γ_t for problem $\mathbf{P6}$ is given by

$$\gamma_t^* = \begin{cases}\underline{\gamma}_t^{(P6)}, & g(\underline{\gamma}_t^{(P6)}) \ge 0,\\ \overline{\gamma}_t^{(P6)}, & g(\overline{\gamma}_t^{(P6)}) \le 0,\\ \text{the unique root of } g(\gamma_t)=0, & \text{otherwise.}\end{cases} \tag{36}$$

For the third case in (36), the optimal γ_t can be efficiently obtained using the Brent–Dekker method [45]. Subsequently, the optimal q_t^k can be computed by substituting γ_t into (31).

From the above analysis, the optimized $l_{t,k}^*$, $q_{t,k}^*$, and γ_t^* can be obtained after relaxing q_t^k as a continuous variable. Since q_t^k should finally take integer values, we need to round $q_{t,k}^*$ back to its nearest integers. According to Theorem 1 and (8), a larger q_t^k leads to a smaller quantization error C_t but a larger transmitted bit size. Therefore, rounding $q_{t,k}^*$ down will increase the quantization error while reducing the communication load, which helps decrease the communication latency. In contrast, rounding it up will reduce the quantization error but may increase the transmitted bit size and communication latency, thereby increasing the risk of violating the latency constraint. Consequently, the two rounding directions reflect a trade-off between quantization error and communication overhead. To further analyze the impact of rounding direction, let the subscript 'floor' denote the solution obtained by rounding $q_{t,k}^*$ down and re-optimizing the corresponding l_t^k and γ_t. Then, the following theorem is provided based on Lemma 1.

Theorem 3. *With the virtual latency queue initialized to zero, by rounding $q_{t,k}^*$ down and re-optimizing l_t^k and γ_t, the performance gap between the online solution to $\mathbf{P2}$ and the optimal solution to $\mathbf{P1}$ remains bounded and controllable:*

$$\sum_{t=1}^{T}\left[\Phi_{\textit{floor}}^*(t) - \Phi^\dagger(t)\right] \le \frac{\Theta T^2}{V} + \sum_{t=1}^{T}U(t), \tag{37}$$

and the cumulative latency is bounded by

$$\sum_{t=1}^{T}T_t \le T^{\max} + \sqrt{2\Theta T^2 + 2V\sum_{t=1}^{T}\Phi^*(t) + 2V\sum_{t=1}^{T}U(t)}, \tag{38}$$

where

$$U(t) = \frac{4\ln 2}{|\mathcal{K}_t|}\sum_{k\in\mathcal{K}_t}\delta_{t,k}^2. \tag{39}$$

In contrast, if upward rounding is allowed, no uniform bound on the performance gap that is independent of d_t can be guaranteed.

Proof: See Appendix G in the supplementary material.

Based on Theorem 3 and the above analysis, to keep the performance gap and long-term latency controllable, we round $q_{t,k}^*$ down as $\tilde{q}_{t,k}^* = \lfloor q_{t,k}^* \rfloor$. Although upward rounding can reduce the quantization error, it may incur a d_t-dependent latency penalty. Subsequently, with $q_t^k = \tilde{q}_{t,k}^*$ fixed, we re-optimize l_t^k and γ_t for problem $\mathbf{P4}$ using Theorem 2 and bisection search, which yields the final optimized $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, and γ_t^*. The optimization algorithm for q_t^k, l_t^k, and γ_t is provided in Algorithm 1.

Algorithm 1 Optimal Quantization Bit Width and Wireless Bandwidth Allocation with Latency Control

Input: Number of participating devices K_t, tolerance ϵ_0.
Form the selected device set $\mathcal{K}_t$ based on K_t, and compute the lower bound $\gamma_t^{\text{low}} = \underline{\gamma}_t^{(P4)}$ and upper bound $\gamma_t^{\text{high}} = \overline{\gamma}_t^{(P4)}$ based on (28) and (29), respectively;
repeat
  while $|\gamma_t^{\text{high}} - \gamma_t^{\text{low}}| > \epsilon_0$ **do**
    $\gamma_t^{\text{mid}} = (\gamma_t^{\text{low}} + \gamma_t^{\text{high}})/2$;
    Compute $l_{t,k}^*$ for each device $k \in \mathcal{K}_t$ using γ_t^{mid} based on Theorem 2;
    if $\sum_{k\in\mathcal{K}_t} l_{t,k}^* < 1$ **then**
      $\gamma_t^{\text{high}} = \gamma_t^{\text{mid}}$;
    else if $\sum_{k\in\mathcal{K}_t} l_{t,k}^* < 1 - \epsilon_0$ **then**
      $\gamma_t^{\text{low}} = \gamma_t^{\text{mid}}$;
    end if
  end while
  Calculate γ_t^* and $q_{t,k}^*$ using (36) and (31), respectively.
until $q_{t,k}^*$, $l_{t,k}^*$, and γ_t^* converge;
Set $\tilde{q}_{t,k}^* = \lfloor q_{t,k}^* \rfloor$ and re-optimize the corresponding $l_{t,k}^*$ and γ_t^* using Theorem 2 and the above bisection search;
return $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, and γ_t^*.

Algorithm 2 AQAFL

Sort the devices in ascending order based on their remaining computation time.
Initialize $K_t = 1$ to form the initial device $\mathcal{K}_t$;
Obtain optimized $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, and γ_t^* through Algorithm 1, and compute the current objective function in (22);
repeat
  Set $K_t = K_t + 1$ to form the selected device set $\mathcal{K}_t$;
  Obtain optimized $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, and γ_t^* via Algorithm 1;
  Compute the objective function in (22);
until the objective function in (22) no longer decreases;
return $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, γ_t^*, and $\mathcal{K}_t$.

C. Device Scheduling Strategy

As discussed earlier, the optimized $\tilde{q}_{t,k}^*$, $l_{t,k}^*$, and γ_t^* can be obtained for the given device set $\mathcal{K}_t$. Meanwhile, for a given number of participating devices K_t, the first K_t devices that complete local training are selected to form $\mathcal{K}_t$ for global updating [18]. Thus, in each round t, the devices are initially sorted in ascending order of their remaining computation time. Then, we gradually increase K_t and select the first K_t devices in the sorted list to form $\mathcal{K}_t$, until the objective function of problem $\mathbf{P3}$ no longer decreases. Therefore, based on Algorithm 1, the proposed AQAFL is given in Algorithm 2. In practical deployment, AQAFL is executed in each round on the server side to optimize K_t, q_t^k, and l_t^k for devices. The solution obtained by AQAFL is not guaranteed to be globally optimal. However, AQAFL can yield an effective solution with polynomial computational complexity, and its effectiveness is further validated by the experimental results.

In Algorithm 1, computing l_t^k for each device k has a complexity $\mathcal{O}(1)$. Therefore, the complexity of optimizing γ_t and l_t^k for all participating devices using bisection search is $\mathcal{O}(K\log_2(\frac{1}{\epsilon_0}))$. The computation of q_t^k and γ_t using (36) and (31) does not exceed this cost. Thus, the total complexity of Algorithm 1 is $\mathcal{O}(IK\log_2(\frac{1}{\epsilon_0}))$, where I is the number of alternating iterations. In Algorithm 2, sorting all devices requires a complexity of $\mathcal{O}(K\log_2 K)$, and at most K iterations of device selection are performed. Consequently, the proposed AQAFL has a polynomial time complexity $\mathcal{O}(K\log_2 K) + \mathcal{O}(IK^2\log_2(\frac{1}{\epsilon_0}))$.

In practical systems, the available channel state information (CSI) may become outdated and deviate from the actual channel conditions, thereby affecting optimization decisions. When outdated CSI overestimates the channel quality, the transmission rate will be overestimated and the communication latency will be underestimated, which may increase the actual round latency and the risk of violating the long-term latency constraint. Conversely, when outdated CSI underestimates the channel quality, the actual transmission rate is higher than the estimated rate, making the optimization decisions relatively conservative. However, this does not invalidate the convergence analysis established in Theorem 1, and the proposed AQAFL remains applicable, because CSI errors do not alter the mathematical forms of the local updates and model aggregation. More importantly, AQAFL can adaptively mitigate the impact of outdated CSI to some extent. At the end of each round, the latency queue d_t is updated according to the actual round latency. Since d_t serves as the weight of the latency term in problem $\mathbf{P2}$, AQAFL can adaptively adjust the trade-off between learning performance and latency in subsequent online optimization based on the actual latency state, thereby alleviating the impact of outdated CSI. To further reduce the risk of latency violation, a straightforward approach is to construct a conservative estimate of the channel gain based on the available CSI. If the actual channel gain is no lower than this conservative estimate, the actual transmission latency will not exceed the estimated latency. Since CSI is an environmental parameter, its treatment does not require modifications to AQAFL.

V. Experimental Results

In this section, we evaluate the effectiveness of the proposed AQAFL by comparing it with other methods on two benchmark datasets in terms of test accuracy, communication overhead, and latency consumption, as detailed in Section V-B. Meanwhile, we perform experiments to investigate the impact of different values of the Lyapunov weight V on the test performance of AQAFL in Section V-D.

A. Experimental Setup

We consider a circular network area with a radius of 600 m, where $K = 20$ devices are randomly distributed, and the server is located at the center. The total available uplink bandwidth for wireless transmission is $B = 20$ MHz. The path loss model is $128.1 + 37.6\log_{10}(d)$, where d denotes the distance (in km) between a device and the server, and the small-scale fading is modeled as Rayleigh fading. The noise power spectral density is $N_0 = -174$ dBm/Hz [46]. Unless otherwise specified, we adopt the following experimental settings.

1) Datasets and Models: We conduct experiments on two commonly used datasets, CIFAR-10 and CINIC-10. Following [15], [18], the Dirichlet distribution is used to conduct non-IID data partitioning, and the degree of data heterogeneity is controlled by the Dirichlet parameter α. For both datasets, we employ a ResNet-20 model with the same architecture as in [40], which consists of 269,722 parameters in total.

2) System Settings: According to the measured computational workload in [40], the number of floating point operations (FLOPs) needed for ResNet-20 to process one training sample of CIFAR-10 or CINIC-10 per iteration is 123.9 MFLOPs. We also follow [40] to base our calculations on the processing capabilities of the iPhone X, which has a processing capacity of 691.2 billion FLOPs (GFLOPs). We set the available CPU frequency to be randomly selected from the range [2, 3] GHz [40], [47]. Moreover, the downlink quantization bit width is set to 16. The transmit power of each device is randomly selected from the range [10, 26] dBm.

3) Baseline Methods: To the best of our knowledge, this paper is the first to consider the joint optimization of quantization bit width, bandwidth allocation, and device scheduling in asynchronous FL. Therefore, we select the following representative methods with similar objectives for comparison.

- Fully Asynchronous FL (Fully AsynFL): The server updates the global model once it receives one local update. A fixed quantization bit width is adopted for all devices.
- Asynchronous FL method FedBuff (AsynFL-Buffer) [14]: The server performs model aggregation after receiving N local updates, where N is the predefined number of participating devices per round. It does not optimize quantization bit width and allocates bandwidth uniformly.
- Asynchronous FL method (AsynFL) in [35]: In each round, the server updates the global model when the per-round latency constraint is reached or it receives N local updates, with no quantization optimization and uniform bandwidth allocation.
- Asynchronous FL with Optimization of Device Scheduling and Bandwidth Allocation (AsynFL-DB) [18]: This method optimizes the number of participating devices and bandwidth allocation for each round under the long-term latency constraint, but does not optimize quantization.
- Asynchronous FL with Quantization Bit Width and Bandwidth Allocation Optimization (AsynFL-QB), adapted from synchronous methods [27]: Following the asynchronous scheduling rule in [35], in each round the server aggregates at most N local updates under a per-round latency constraint, while the quantization bit width and bandwidth allocation of the participating devices are optimized by the synchronous method proposed in [27].

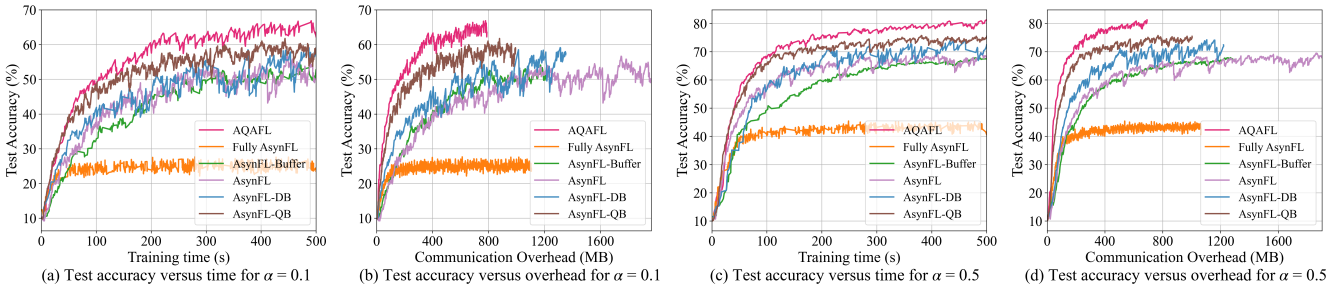
For the baselines without quantization optimization (Fully AsynFL, AsynFL-Buffer, AsynFL, and AsynFL-DB), we set the quantization bit width to 16 to minimize quantization error. For baselines AsynFL-Buffer, AsynFL, and AsynFL-QB that require specifying the number of participating devices, we present the performance of $N = 5$ due to space limitations, and the results on other values of N show similar conclusions.

4) Implementation Details: Since different asynchronous FL methods generally take different amounts of time to complete each round of training and model updating, following [17], [48], we do not fix the total number of training rounds, but instead set a unified maximum training time $T^{\max} = 500$ s for all methods. The training process continues until $T^{\max}$ is reached. This ensures a fair comparison and better reflects practical scenarios. Such a setting does not conflict with our theoretical analysis, since the derived results hold for any finite number of training rounds, and the proposed AQAFL can be executed online without knowing the total training rounds in advance once the target long-term average latency $\overline{T}$ is given. The $T^{\max}$ also serves as the long-term latency constraint for our proposed AQAFL and the baseline AsynFL-DB. When the total number of training rounds is not fixed, our AQAFL and the baseline AsynFL-DB both rely on $\overline{T}$, which is set to 1 s by default. Meanwhile, the baselines AsynFL and AsynFL-QB require per-round latency constraints. For a fair comparison, the per-round latency constraints of AsynFL and AsynFL-QB are also set as $\overline{T}$. In the experiments, we evaluate the test performance of the global model after each model aggregation. The quantization limits $\underline{\Delta}_{t,k}$ and $\overline{\Delta}_{t,k}$ are set to 0 and 0.5, respectively, since the absolute value of model weight updates rarely exceeds 0.5 in practice. The maximum allowed quantization bit width $q_{t,k}^{\max}$ for each device k is set to 32. The number of local epochs is set to 5. The local optimizer is SGD with learning rate of 0.01 and momentum of 0.9. The local batch size is set to 64.

B. Performance Evaluation of AQAFL

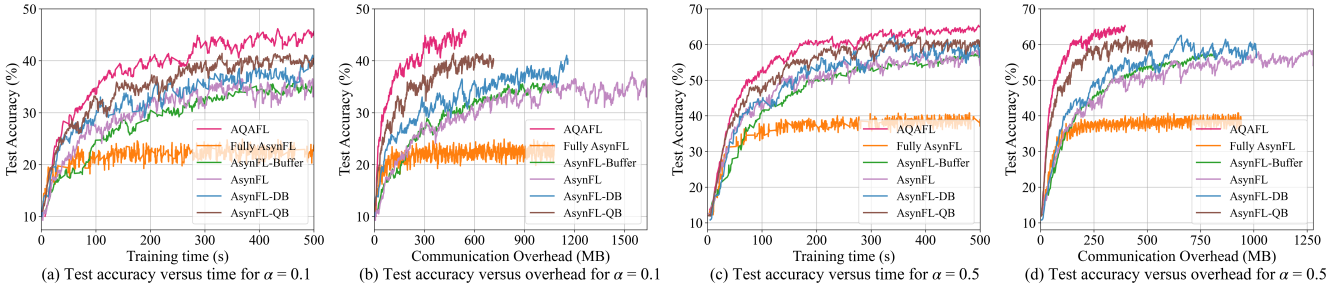
We first present the test accuracy of different methods with respect to training time and communication overhead on the CIFAR-10 and CINIC-10 datasets, as shown in Figures 1 and 2, where the Dirichlet parameter α is set to 0.1 and 0.5. The communication overhead here refers to the cumulative uplink communication volume (in MB) of the asynchronous FL system over the training process, since our study focuses on uplink optimization to alleviate the communication bottleneck caused by device uploads. It is observed that AQAFL significantly outperforms these baselines in all cases, achieving 4.1%–6.5% improvement in test accuracy under the same training time. Moreover, AQAFL reduces the communication overhead by 21%–69% compared with other methods, and is able to achieve the same test accuracy with lower overhead. It is worth noting that the baseline AsynFL-QB, which directly applies a synchronous optimization method to the asynchronous setting, yields considerably higher communication overhead and lower test accuracy than AQAFL. This highlights the necessity of tailored optimization for asynchronous FL. Additionally, while the degree of data heterogeneity affects the system computation latency and model performance, AQAFL consistently exhibits stable and superior performance.

To evaluate the practical feasibility of AQAFL, we further compare the average per-round wall-clock time (ms) required by AQAFL, AsynFL-DB, and AsynFL-QB for joint optimization on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$, since all these methods involve optimization processes. As shown in Table II, we also report their final test accuracy and communication overhead. The final test accuracy refers



(a) Test accuracy versus time for $\alpha = 0.1$ (b) Test accuracy versus overhead for $\alpha = 0.1$ (c) Test accuracy versus time for $\alpha = 0.5$ (d) Test accuracy versus overhead for $\alpha = 0.5$

Fig. 1. Comparison of test accuracy versus training time and communication overhead for different methods on the CIFAR-10 dataset. (a) and (b) show the results with Dirichlet parameter $\alpha = 0.1$, while (c) and (d) report the results under $\alpha = 0.5$.



(a) Test accuracy versus time for $\alpha = 0.1$ (b) Test accuracy versus overhead for $\alpha = 0.1$ (c) Test accuracy versus time for $\alpha = 0.5$ (d) Test accuracy versus overhead for $\alpha = 0.5$

Fig. 2. Comparison of test accuracy versus training time and communication overhead for different methods on the CINIC-10 dataset. (a) and (b) present the results under $\alpha = 0.1$, while (c) and (d) show the results under $\alpha = 0.5$.

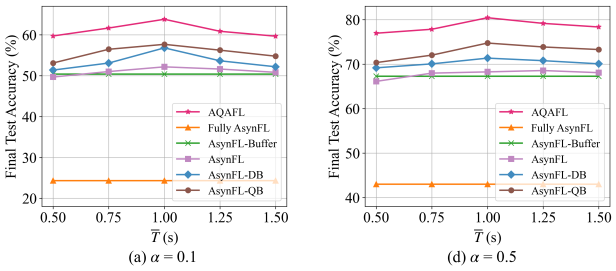
TABLE II
TEST ACCURACY, COMMUNICATION OVERHEAD, AND AVERAGE PER-ROUND WALL-CLOCK TIME FOR JOINT OPTIMIZATION ON THE CIFAR-10 AND CINIC-10 DATASETS WITH $\alpha = 0.5$.

Methods	CIFAR-10			CINIC-10		
	Accuracy (%)	Overhead (MB)	Time (ms)	Accuracy (%)	Overhead (MB)	Time (ms)
AsynFL-DB	71.34	1220.7	6.94	59.05	1012.1	6.17
AsynFL-QB	74.73	1002.9	9.36	60.64	522.4	8.62
AQAFL	80.42	691.2	7.63	64.75	396.3	6.43

to the average test accuracy over the last 10 seconds of the training process. It is observed that AQAFL achieves the highest test accuracy and the lowest communication overhead while maintaining a competitive optimization time.

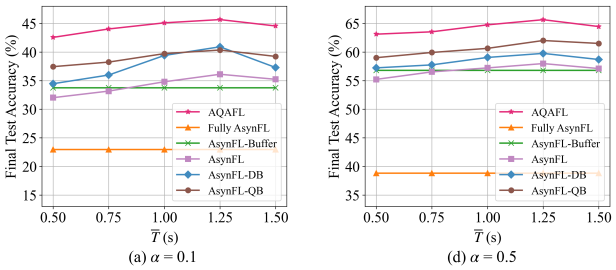
We further report the final test accuracy of different methods under varying $\overline{T}$, as shown in Figures 3 and 4. Note that Fully AsynFL and AsynFL-Buffer are not influenced by $\overline{T}$. The results show that AQAFL consistently achieves higher final test accuracy than other methods, and outperforms the best competing baseline by approximately 4.6%–6.7%. The above results demonstrate the robustness of AQAFL under different degrees of data heterogeneity. Due to space limitations and to better highlight the time and communication efficiency, we present the comparisons under $\alpha = 0.5$ in the following, and the results under $\alpha = 0.1$ show similar conclusions.

We next present the time required for different methods to reach the same target test accuracy under different $\overline{T}$ on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$, as shown in Tables III and IV. The target accuracies are set to 65% and 70% for CIFAR-10, and 55% and 60% for CINIC-10. As can



(a) $\alpha = 0.1$ (d) $\alpha = 0.5$

Fig. 3. Final average test accuracy over the last 10 seconds achieved by different methods under various $\overline{T}$ on the CIFAR-10 dataset. (a) shows the results under $\alpha = 0.1$, and (b) for $\alpha = 0.5$.



(a) $\alpha = 0.1$ (d) $\alpha = 0.5$

Fig. 4. Final average test accuracy over the last 10 seconds achieved by different methods under various $\overline{T}$ on the CINIC-10 dataset. (a) presents the results under $\alpha = 0.1$, and (b) for $\alpha = 0.5$.

be seen, AQAFL consistently requires less time than other methods to achieve the same target accuracy. Moreover, the baseline Fully AsynFL cannot reach the target accuracy due to its persistently low test accuracy. In addition, the baselines AsynFL and AsynFL-Buffer also fail to reach the higher target accuracy. These results demonstrate that the proposed AQAFL

TABLE III
TRAINING TIME TO REACH THE TARGET ACCURACY UNDER DIFFERENT $\overline{T}$ ON THE CIFAR-10 DATASET WITH $\alpha = 0.5$. '–' MEANS FAILURE TO REACH THE TARGET ACCURACY. ALL TIMES ARE REPORTED IN SECONDS.

Methods	$\overline{T} = 0.75$ s		$\overline{T} = 1$ s		$\overline{T} = 1.25$ s	
	65%	70%	65%	70%	65%	70%
Fully AsynFL	–	–	–	–	–	–
AsynFL-Buffer	325.5	–	325.5	–	325.5	–
AsynFL	215.1	–	165.7	–	206.2	–
AsynFL-DB	111.7	199.3	149.4	209.4	136.4	205.1
AsynFL-QB	101.3	131.7	87.5	154.3	89.6	150.7
AQAFL	64.2	90.4	79.1	116.8	86.4	124.5

TABLE IV
TRAINING TIME TO REACH THE TARGET ACCURACY UNDER DIFFERENT $\overline{T}$ ON THE CINIC-10 DATASET WITH $\alpha = 0.5$. '–' MEANS FAILURE TO REACH THE TARGET ACCURACY. ALL TIMES ARE REPORTED IN SECONDS.

Methods	$\overline{T} = 0.75$ s		$\overline{T} = 1$ s		$\overline{T} = 1.25$ s	
	55%	60%	55%	60%	55%	60%
Fully AsynFL	–	–	–	–	–	–
AsynFL-Buffer	383.1	–	383.1	–	383.1	–
AsynFL	313.2	–	242.7	–	209.8	–
AsynFL-DB	157.6	282.2	208.3	329.8	197.1	349.2
AsynFL-QB	193.9	455.1	145.4	286.7	170.6	258.4
AQAFL	128.9	218.6	115.4	169.5	110.8	181.9

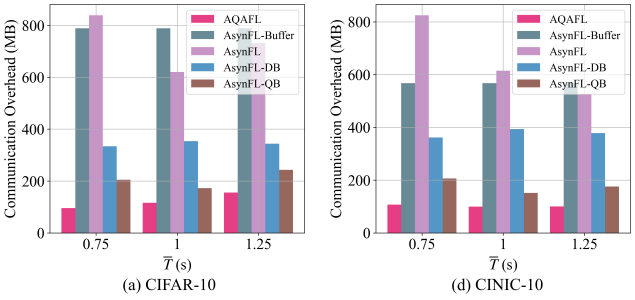


Fig. 5. Communication overhead of methods to reach the target test accuracy under different $\overline{T}$ on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$.

can effectively improve the training efficiency.

We also present the communication overhead required by different methods to reach the target accuracy under different $\overline{T}$, as shown in Figure 5. The target accuracy is set to 65% for CIFAR-10 and 55% for CINIC-10, since all methods except Fully AsynFL can reach these targets, enabling an effective comparison. Baseline Fully AsynFL is excluded from Figure 5 because its test accuracy is too low to meet the target accuracy. As can be observed, AQAFL incurs lower communication overhead than other methods, reducing the communication overhead by 32.6%–88.6%. These results demonstrate the superiority of AQAFL in terms of communication efficiency.

We further report the average number of participating devices per round and the average quantization bit width per device per round of the proposed AQAFL under different $\overline{T}$ on the CIFAR-10 and CINIC-10 datasets, as shown in Figure 6. It can be seen that both the average number of participating devices and quantization bit width gradually increase with $\overline{T}$.

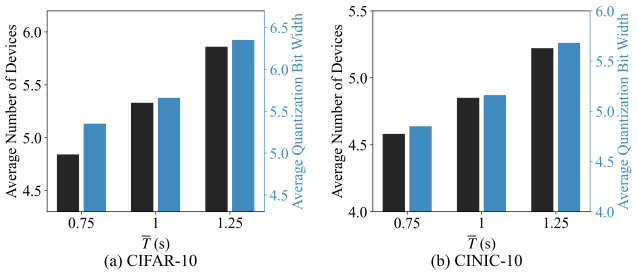


Fig. 6. Average number of participating devices per round and average quantization bit width per device per round of AQAFL under different $\overline{T}$ on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$

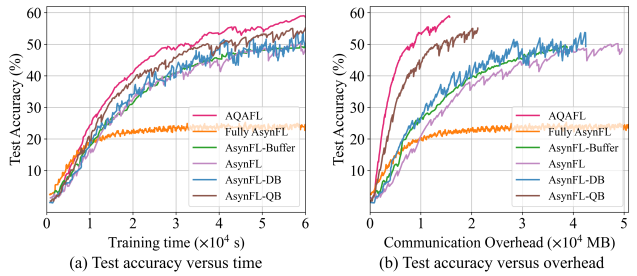
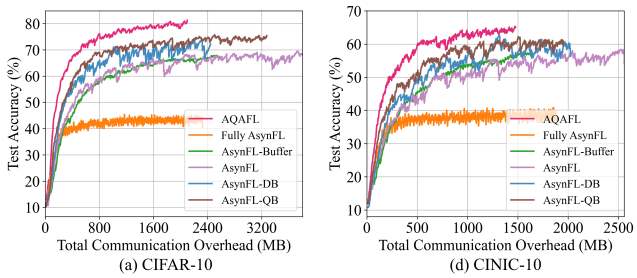


Fig. 7. Test accuracy versus training time and uplink communication overhead for different methods on the ImageNet-100 dataset using VGG11.

This is because a larger $\overline{T}$ corresponds to a looser target long-term average latency, which allows the server to schedule more devices and adopt higher quantization bit widths for them in each round. However, as shown in Figures 3 and 4, under the same long-term latency constraint, an overly large $\overline{T}$ may lead to insufficient convergence within the given training time budget, resulting in degraded final test accuracy. Therefore, it is necessary to choose an appropriate $\overline{T}$ in practice.

To further evaluate the scalability of AQAFL on larger models and datasets, we conduct experiments on the ImageNet-100 dataset [49] with $\alpha = 0.5$ using the VGG11 model adopted in [50]. We measure the computational workload utilizing the same approach in [40] and obtain that VGG11 requires 22.496 GFLOPs to process one training sample of ImageNet-100 per iteration. Accordingly, we set $T^{\max} = 6 \times 10^4$ s and compare the test accuracy of different methods with respect to training time and uplink communication overhead under $\overline{T} = 300$ s, as shown in Figure 7. It is observed that AQAFL still achieves the highest test accuracy and the lowest communication overhead when scaling to larger models and datasets.

Although our primary focus is on the uplink optimization, the downlink communication overhead is still evaluated for completeness. As shown in Figure 8, we present the test accuracy of different methods versus the total communication overhead on the CIFAR-10 and CINIC-10 datasets under $\overline{T} = 1$ s. The total communication overhead refers to the cumulative uplink and downlink volume (in MB) incurred during the training process, where the downlink overhead is influenced by the number of participating devices in each round. It is observed that the results exhibit trends and conclusions similar to those in Figures 1 and 2, achieving up to 44.8% overhead reduction, which further confirms the efficiency of AQAFL.



(a) CIFAR-10 (d) CINIC-10

Fig. 8. Test accuracy versus total communication overhead of different methods on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$.

TABLE V
TEST ACCURACY AND UPLINK COMMUNICATION OVERHEAD OF AQAFL AND ITS VARIANTS ON CIFAR-10 AND CINIC-10 WITH $\alpha = 0.5$.

Methods	CIFAR-10		CINIC-10	
	Accuracy (%)	Overhead (MB)	Accuracy (%)	Overhead (MB)
NQ	75.89	1058.4	61.08	789.1
ND	77.54	938.7	63.26	437.4
NB	78.26	1096.1	61.93	760.2
AQAFL	80.42	691.2	64.75	396.3

C. Effect of Adaptive Quantization Bit Width, Device Scheduling, and Wireless Bandwidth Allocation

To evaluate the impact of adaptive quantization bit width, device scheduling, and bandwidth allocation in AQAFL, we compare it with the following variants under $\overline{T} = 1$ s on the CIFAR-10 and CINIC-10 datasets with $\alpha = 0.5$.

- NQ (No Quantization Optimization): The device scheduling and bandwidth allocation are optimized using AQAFL with a fixed quantization bit width of 16.
- ND (No Device Scheduling Optimization): AQAFL is used to optimize quantization bit width and bandwidth allocation, and the server performs model aggregation after receiving $N = 5$ local updates.
- NB (No Bandwidth Optimization): AQAFL is applied to optimize quantization bit width and device scheduling with uniform bandwidth allocation.

Table V shows the final test accuracy and uplink communication overhead of these methods. It is observed that removing any of the optimization components will lead to a decrease in test accuracy and a substantial increase in communication overhead, demonstrating the necessity of joint optimization of these components. Moreover, these variants still outperform the baselines that optimize the same components.

D. Parameters Analysis

We evaluate the impact of the Lyapunov weight V on AQAFL under $\overline{T} = 1$ s on the CIFAR-10 dataset with $\alpha = 0.5$. As shown in Table VI, we report the training time and communication overhead required to reach target test accuracies of 65% and 75%, as well as the final average test accuracy over the last 10 seconds. As V increases, both the training time and communication overhead required to reach the target accuracy first decrease and then increase. A similar trend is observed for the final test accuracy, and AQAFL achieves the best overall performance at $V = 20$. Nevertheless, AQAFL consistently exhibits superior performance as V changes, demonstrating its robustness to the choice of V. In practice, a lightweight preliminary evaluation over a set of candidate V values can be conducted before training, and an appropriate value can be selected by jointly considering the learning and latency performance. Moreover, if the system requirements specify an acceptable optimality gap κ a priori, V can be chosen on the order of $1/\kappa$ based on Lemma 1 and Theorem 3.

TABLE VI
TRAINING TIME AND COMMUNICATION OVERHEAD TO REACH THE TARGET ACCURACY AND FINAL ACCURACY OVER THE LAST 10 SECONDS UNDER DIFFERENT V ON THE CIFAR-10 DATASET.

V	65%			75%			Final Accuracy
	Training Time (s)	Uplink Overhead (MB)	Total Overhead (MB)	Training Time (s)	Uplink Overhead (MB)	Total Overhead (MB)	
1	92.3	133.1	407.7	275.2	371.9	921.2	78.75%
5	86.1	127.9	379.9	225.3	339.2	857.3	79.21%
10	82.5	121.6	346.2	204.5	304.1	819.6	79.68%
20	79.1	116.4	323.8	179.4	259.1	753.9	80.42%
30	87.4	124.4	355.7	185.7	268.2	778.2	80.13%

VI. CONCLUSION

In this paper, we proposed AQAFL to improve the efficiency of asynchronous FL under limited wireless resources. We first conducted a convergence analysis for asynchronous FL with quantized transmissions, demonstrating the impact of the quantization errors induced in each round. Then, the quantization bit width, device scheduling, and bandwidth allocation were jointly optimized to minimize the convergence bound using the Lyapunov optimization framework under long-term latency and bandwidth constraints. Experiments have shown that compared with other state-of-the-art methods, AQAFL achieves lower communication overhead and higher test accuracy under the same training time. Nevertheless, AQAFL currently requires the Lyapunov weight V to be specified in advance. In future work, we will investigate adaptive mechanisms for adjusting V to further enhance the performance of AQAFL and facilitate its practical deployment.

REFERENCES

[1] T. Zhang, K.-Y. Lam, J. Zhao, and J. Feng, "Joint device scheduling and bandwidth allocation for federated learning over wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 24, no. 1, pp. 3–18, 2025.
[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017.
[3] Y. Zhao, C. Qiu, S. Cai, Z. Liu, Y. Wang, X. Wang, and Q. Hu, "Multi-granularity weighted federated learning for heterogeneous edge computing," *IEEE Trans. Serv. Comput.*, vol. 18, no. 1, pp. 270–287, 2025.
[4] S. Zhang, J. Li, L. Shi, M. Ding, D. C. Nguyen, W. Tan, J. Weng, and Z. Han, "Federated learning in intelligent transportation systems: Recent applications and open problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3259–3285, 2024.
[5] X. Zhang, H. Xing, Y. Shen, J. Xu, and S. Cui, "Age of information minimization in uav-enabled iot networks via federated reinforcement learning," *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2025.

[6] N. Lang, A. Cohen, and N. Shlezinger, "Stragglers-aware low-latency synchronous federated learning via layer-wise model updates," *IEEE Trans. Commun.*, vol. 73, no. 5, pp. 3333–3346, 2025.

[7] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[8] 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. Wirel. Commun.*, vol. 20, no. 1, pp. 453–467, 2021.

[9] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless iot networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, 2021.

[10] Z. Yan, D. Li, X. Yu, and Z. Zhang, "Latency-efficient wireless federated learning with quantization and scheduling," *IEEE Commun. Lett.*, vol. 26, no. 11, pp. 2621–2625, 2022.

[11] R. Zhang, X. Wu, Y. Zou, Z. Xie, P. Li, X. Cheng, F. Dressler, and D. Yu, "Fed-raa: Resource-adaptive asynchronous federated edge learning with theoretical guarantee," *IEEE. Trans. Mob. Comput.*, vol. 25, no. 6, pp. 8830–8843, 2026.

[12] Y. Xu, Z. Ma, H. Xu, S. Chen, J. Liu, and Y. Xue, "Fedlc: Accelerating asynchronous federated learning in edge computing," *IEEE. Trans. Mob. Comput.*, vol. 23, no. 5, pp. 5327–5343, 2024.

[13] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, H. Shan, W. Wang, and T. Q. S. Quek, "Asynchronous federated learning over wireless communication networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 9, pp. 6961–6978, 2022.

[14] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," 2022, *arXiv:2106.06639*.

[15] S. Sun, Z. Zhang, Q. Pan, M. Liu, Y. Wang, T. He, Y. Chen, and Z. Wu, "Staleness-controlled asynchronous federated learning: Accuracy and efficiency tradeoff," *IEEE. Trans. Mob. Comput.*, vol. 23, no. 12, pp. 12 621–12 634, 2024.

[16] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[17] H. Zhu, Y. Zhou, H. Qian, Y. Shi, X. Chen, and Y. Yang, "Online client selection for asynchronous federated learning with fairness consideration," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 4, pp. 2493–2506, 2023.

[18] Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Adaptive semi-asynchronous federated learning over wireless networks," *IEEE Trans. Commun.*, vol. 73, no. 1, pp. 394–409, 2025.

[19] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 5, pp. 923–927, 2022.

[20] S. Xie, D. Wen, C. You, Q. Chen, M. Bennis, and K. Huang, "Fedlodrop: Federated lora with dropout for generalized llm fine-tuning," *IEEE J. Sel. Areas Commun.*, vol. 44, pp. 3541–3556, 2026.

[21] D. Yang, Y. Hou, and W. Wu, "Convergence analysis of asynchronous federated learning with gradient compression for non-convex optimization," 2025, *arXiv:2504.19903*.

[22] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2150–2167, 2021.

[23] Y. Li, Y. Guo, M. Alazab, S. Chen, C. Shen, and K. Yu, "Joint optimal quantization and aggregation of federated learning scheme in vanets," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19 852–19 863, 2022.

[24] Y. Liu, P. Huang, F. Yang, K. Huang, and L. Shu, "Quasyncfl: Asynchronous federated learning with quantization for cloud–edge–terminal collaboration enabled aiot," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 59–69, 2024.

[25] T. Ortega and H. Jafarkhani, "Quantized and asynchronous federated learning," *IEEE Trans. Commun.*, vol. 73, no. 4, pp. 2361–2374, 2025.

[26] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 323–341, 2022.

[27] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless quantized federated learning: A joint computation and communication design," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2756–2770, 2023.

[28] S. Wang, M. Chen, C. G. Brinton, C. Yin, W. Saad, and S. Cui, "Performance optimization for variable bitwidth federated learning in wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 3, pp. 2340–2356, 2024.

[29] C. Chen, Y. Zhao, Z. Zhang, W. Li, and J. Wu, "Toward efficient and scalable asynchronous federated learning via stragglers version control," *IEEE. Trans. Mob. Comput.*, vol. 25, no. 2, pp. 2627–2643, 2026.

[30] X. Wang, Z. Li, S. Jin, and J. Zhang, "Achieving linear speedup in asynchronous federated learning with heterogeneous clients," *IEEE. Trans. Mob. Comput.*, vol. 24, no. 1, pp. 435–448, 2025.

[31] Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Adaptive model pruning for communication and computation efficient wireless federated learning," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 7, pp. 7582–7598, 2024.

[32] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 4, pp. 2457–2471, 2021.

[33] Y. Yang, S. Dang, and Z. Zhang, "An adaptive compression and communication framework for wireless federated learning," *IEEE. Trans. Mob. Comput.*, vol. 23, no. 12, pp. 10 835–10 854, 2024.

[34] X. Liu, S. Wang, Y. Deng, and A. Nallanathan, "Adaptive federated pruning in hierarchical wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 6, pp. 5985–5999, 2024.

[35] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "Safa: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Trans. Comput.*, vol. 70, no. 5, pp. 655–668, 2021.

[36] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[37] A. Kavis, K. Y. Levy, and V. Cevher, "High probability bounds for a class of nonconvex algorithms with adagrad stepsize," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[38] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "FedDisco: Federated learning with discrepancy-aware collaboration," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, vol. 202. PMLR, 23–29 Jul 2023, pp. 39 879–39 902.

[39] B. Fehrman, B. Gess, and A. Jentzen, "Convergence rates for the stochastic gradient descent method for non-convex objective functions," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[40] Z. Zhang, Z. Gao, Y. Guo, and Y. Gong, "Heterogeneity-aware cooperative federated edge learning with adaptive computation and communication compression," *IEEE. Trans. Mob. Comput.*, vol. 24, no. 3, pp. 2073–2084, 2025.

[41] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[42] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization*. Cham, Switzerland: Springer, 2014.

[43] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surv. Oper. Res. Manage. Sci.*, vol. 17, no. 2, pp. 97–106, 2012.

[44] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Adv. Comput. Math.*, vol. 5, no. 1, pp. 329–359, 1996.

[45] G. Wilkins and M. Gu, "A modified brent's method for finding zeros of functions," *Numer. Math.*, vol. 123, no. 1, pp. 177–188, 2013.

[46] X. Zhou, Y. Deng, H. Xia, S. Wu, and M. Bennis, "Time-triggered federated learning over wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 12, pp. 11 066–11 079, 2022.

[47] E. Yu, S. Liu, Q. Li, H. Chen, H. V. Poor, and S. Shamai, "Graph-based joint client clustering and resource allocation for wireless distributed learning: A new hierarchical federated learning framework with non-iid data," *IEEE. Trans. Mob. Comput.*, vol. 24, no. 5, pp. 3579–3596, 2025.

[48] C. Wang and H. Wu, "Energy optimization for federated learning on consumer mobile devices with asynchronous sgd and application co-execution," *IEEE. Trans. Mob. Comput.*, vol. 23, no. 11, pp. 10 235–10 250, 2024.

[49] Z. Zang, S. Li, D. Wu, G. Wang, K. Wang, L. Shang, B. Sun, H. Li, and S. Z. Li, "Dlme: Deep local-flatness manifold embedding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 576–592.

[50] G. Lee, M. Jeong, S. Kim, J. Oh, and S.-Y. Yun, "Fedsol: Stabilized orthogonal learning with proximal restrictions in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 12 512–12 522.



Xiaodong Li received the B.S. degree in Electronic and Information Engineering from Jilin University, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering, Harbin Institute of Technology, China. His research interests include distributed learning, collaborative AI, and federated learning.



Yulong Gao (Senior Member, IEEE) received the M.S.degree in communication and information systems from Harbin Engineering University, Harbin, China, in 2004, and the Ph.D. degree in information and communication engineering from the Harbin Institute of Technology, Harbin, in 2007. From May 2012 to May 2013, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. He is currently a Professor with the School of Electronics and Information Engineering, Harbin Institute of Technology. His current research interests include intelligent communication and intelligent signal processing. He was a recipient of numerous awards, including the First Prize in Science and Technology Advancement Award of Ministry of Education of china in 2012 and 2013, the First Prize of Technological Invention Award of Heilongjiang Province in 2012.



Fengyu Liang received the B.E. degree in Electronic and Information Engineering from Harbin Institute of Technology, China, in 2025. He is currently pursuing the master's degree with the School of Electronics and Information Engineering, Harbin Institute of Technology, China. His research interests include cooperative perception, wireless communication, and edge Intelligence.



Carla Fabiana Chiasserini (Fellow, IEEE) is a Full Professor with the Department of Electronics and Telecommunications, Politecnico di Torino, and an Affiliated Professor with Chalmers University of Technology, Sweden. She is also a Research Associate with CNR-IEIIT and CNIT. Her research interests include the design, modeling, and performance evaluation of mobile networks and services, as well as machine learning models for mobile applications and services.