IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS Connecting the Mobile World | IEEE Transactions on Machine Learning in Communications and Networking

Received 7 March 2023; revised 26 June 2023; accepted 29 July 2023.
Date of publication 7 August 2023; date of current version 18 August 2023.

The associate editor coordinating the review of this article and approving it for publication was C. Shen.

Digital Object Identifier 10.1109/TMLCN.2023.3303292

A Flexible Model Compression and Resource Allocation Scheme for Federated Learning

YIWEN HU[1] (Student Member, IEEE), TINGTING LIU[1] (Member, IEEE), CHENYANG YANG[1] (Senior Member, IEEE), YUANFANG HUANG[2], AND SHIQIANG SUO[2]

[1]School of Electronics and Information Engineering, Beihang University, Beijing 100191, China
[2]CICT Mobile Communication Technology Company Ltd., Beijing 100083, China

CORRESPONDING AUTHOR: T. LIU (ttliu@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2902002 and in part by the National Natural Science Foundation of China (NSFC) under Grant 62271024.

ABSTRACT Communication overhead has become one of the major constraints on the application of federated learning (FL). To reduce the overhead by trading off between the number of communication rounds and per-round latency, significant research efforts have been devoted to investigating joint optimization of model compression, client scheduling, and resource allocation to reduce the total training time. In order to reduce the complexity of joint optimization, the existing methods only consider the same compression level, unchanged participating clients, and identical round duration during training, resulting in low resource usage efficiency. In this paper, we propose a flexible model compression and resource allocation scheme to minimize the total communication time for FL in mobile networks. The proposed scheme can assign adaptive compression levels and communication resources to each client in each round. Simulation results show that the proposed scheme outperforms the start-of-the-art methods and is robust to outdated channel information.

INDEX TERMS Federated learning, model compression, resource allocation, model-free unsupervised learning.

I. INTRODUCTION

IN FEDERATED learning (FL), edge devices collaboratively train a model with the help of a central server [1]. Every device only exchanges models or gradients with the central server while keeping its data local. By using the computation and storage capabilities of edge devices, FL provides artificial intelligence (AI) capability for edge devices while avoids problems caused by uploading raw data to the central server, such as privacy leakage. FL is a promising technology for AI in the edge and has broad applications in wireless networks [2].

In FL, a typical training process involves multiple rounds of communication between the clients and the central server, causing massive communication overhead. Significant research efforts have been devoted to mitigating the communication overhead, which respectively reduce the number of communication rounds, per-round latency, and total training time [3].

To reduce the number of communication rounds, the existing approaches include: 1) Federated averaging (FedAvg) that uses multiple local iterations of stochastic gradient descent (SGD) in each round [4], 2) client scheduling that selects appropriate clients to participate in FL [5], and 3) optimization acceleration, such as momentum FL [6], which introduces the momentum to accelerate the convergence for FL.

To reduce per-round latency, model compression is a widely used method. It can reduce the amount of data for transmission through quantization, sparsification, and their combinations. The commonly used quantization methods consist of scalar quantization (e.g., probabilistic quantization in [4]) and vector quantization (e.g., universal quantization in [7]). Compared with vector quantization, scalar quantization is with higher quantization errors but a wider application due to its lower computational complexity and more adjustable quantization level. The typical sparsification methods contain Rand-M sparsification (considered in sketched update [4]) and Top-M sparsification (used in sparse binary compression (SBC) [8]), where M parameters with maximal magnitude are selected or randomly selected from local parameters or gradients. Compared with Rand-M sparsification, Top-M sparsification can reduce more communication costs without large impact on model accuracy. However, it is necessary to compensate for the accumulated errors. If some mobile clients cannot participate in each round, the stale accumulated errors may cause severe performance degradation [9].

In the presence of client heterogeneity, some clients in worse channel conditions or with lower computation power become laggards in model aggregation due to longer communication or computation times. To avoid the per-round latency being prolonged by these laggards, resource allocation and client scheduling methods have been designed for uplink transmission [10], [11], [12]. In [10], over-the-air computation (AirComp) was considered, where the truncated channel inversion power control and the client scheduling were optimized to maximize the receive signal-to-noise ratio (SNR). However, AirComp is sensitive to channel estimation and time synchronization errors [13]. Furthermore, the analog modulation mechanism is difficult to be incorporated with existing compression methods. In [11], the bandwidth allocation (BA) was optimized for orthogonal frequency domain multiple access (OFDMA) systems, which always allocates more bandwidth to the lagging clients. It reduces latency at the expense of using more wireless resources.

Although model compression can decrease the per-round latency, the compression error will increase the number of communication rounds [14]. By contrast, scheduling more clients can reduce the number of rounds, which however enhances per-round latency [15]. To reduce total training time, it is necessary to optimize the compression, scheduling, and resource allocation in multiple rounds jointly to balance the number of communication rounds and per-round latency. In [16], client scheduling, BA, and quantization were jointly optimized to minimize the total communication time given a fixed number of communication rounds. In [3], quantization level and BA were jointly optimized to minimize the total communication and computation time. In [15], client scheduling and BA were jointly optimized to maximize the model accuracy within given total training time. In [12], the number of local updates, model sparsity ratios among clients, and BA were jointly optimized to minimize the total energy consumption.

To solve a multi-round joint optimization problem, it is necessary to first derive a closed-from expression of total training time or loss function as the objective [3], [15], [16] and then optimize the model compression and resource allocation variables of all rounds. However, since the number of communication rounds for most learning tasks can be hundreds or thousands [12], [16], it is difficult to analyze the relationship between the objective and the optimization variables. Moreover, the computation complexity to optimize these variables simultaneously is unaffordable. To simplify the joint optimization problem, these existing studies introduced some constraints, say all clients must have the same amount of transmission data [15], [16] and the same quantization levels [3], the round durations during training must be identical [3], [15], all clients must participate FL in each round [3], [12], or the number of scheduling clients must remain constant [15]. These methods perform well in static wireless networks. However, mobile networks have the following characteristics. 1) **Changing participating clients**: When some mobile clients move out of the cell or their battery is with low energy, they cannot participate in every round. 2) **Dynamic available bandwidth**: Since FL is usually regarded as a service [17] in mobile networks that coexists with other wireless services, the available bandwidth for FL may change dynamically with traffic load. 3) **Time-varying channel gain**: For most FL tasks, the total training time ranges from a few minutes to several hours [3], [16]. It is unrealistic to expect static channels during training. Jointly optimizing resource allocation and model compression for mobile networks face the following challenges.

1) **Unknown channel information**: To solve the joint optimization problem, the central server is often assumed knowing the channel information of all clients in all rounds at the beginning of the FL. In [3], [12], and [16], it is assumed that the channel information is always unchanged during the training period, which is not reasonable for mobile networks. How to solve a joint optimization problem without future channel information is still an open problem.
2) **Low resource usage efficiency**: The above constraints to simplify the joint optimization problem in [3], [12], [15], and [16], reduce the flexibility of model compression or resource allocation. For example, when all clients have the same amount of transmission data, the optimal BA has to allocate more bandwidth to the clients with worse channel quality to reduce per-round latency [3], [11], [15], [16]. If the clients are allowed to use different compression levels, more bandwidth can be allocated to the clients with good channel quality, which can reduce the required resource to achieve expected learning performance. Similarly, to enhance resource usage efficiency, it is also necessary to adaptively adjust the compression level and per-round duration according to dynamic participating clients and available bandwidth.
3) **Unsuitable sparsification method**: In [12], Top-M sparsification with perfect error compensation (EC) was considered. Rand-M sparsification without error compensation is better suited for mobile clients, which however never been considered in existing joint optimization.

To address these challenges, we develop a flexible model compression and resource allocation scheme to minimize the total communication time for FL in mobile networks. In the proposed scheme, the model compression includes probabilistic quantization and Rand-M sparsification without error compensation, and the resource allocation consists of intra-round BA and inter-round uplink transmission time allocation (UTTA), which can adaptively adjust the compression level and allocate radio resources to different clients in different rounds. Our main contributions are summarized as follows:

1) To formulate a solvable joint optimization problem, we decompose the multi-round optimization problem that minimizes the total communication time into multiple single-round optimization problems that maximizes the convergence speed. The decomposed problem is independent of future channel information and is capable of balancing the number of rounds and the per-round latency, which reduces the computation complexity significantly without performance loss.
2) To optimize intra-round BA to improve resource usage efficiency, all clients can employ different sparsity ratios. Since our analysis reveals that the convergence speed depends on the average sparsity ratio of all clients rather than the minimal sparsity ratio, the proposed BA method allocates more bandwidth to the client with the maximal channel gain, which can achieve higher resource usage efficiency than the existing BA methods.
3) To optimize inter-round UTTA to maximize the convergence speed in the dynamic environment, it is necessary to analyze the closed-form expression of the convergence speed. Considering that the expression is too complicated to analyze for nonlinear learning models, such as deep neural networks (DNN), we use model-free unsupervised learning to optimize the UTTA. The proposed method can adapt to the dynamic bandwidth and participating clients, and time-varying channel gains and provide optimal uplink transmission time in arbitrary mobile scenario.

The rest of the paper is organized as follows. We introduce the system model in Section II and propose a flexible model compression and resource allocation scheme in Section III. The performance of the proposed scheme is evaluated in Section IV. Finally, we conclude the paper in Section V.

II. SYSTEM MODEL

Consider a mobile network consisting of a base station (BS) and K mobile clients, denoted by $\mathcal{K} = \{1, \cdots, K\}$, who collaboratively train a model through FL. The model parameter vector (referred to as model vector for short) is represented by $\boldsymbol{w} \in \mathbb{R}^{N_{\text{model}}}$, where N_{model} is the model size. For client k, the local empirical loss function is

$$L_k(\boldsymbol{w}) = \frac{1}{|\mathcal{D}_k|} \sum_{j=1}^{|\mathcal{D}_k|} \mathcal{L}\left(\boldsymbol{w}; \boldsymbol{x}_k^j, \boldsymbol{y}_k^j\right) \tag{1}$$

where $\left(\boldsymbol{x}_k^j, \boldsymbol{y}_k^j\right)$ is the jth sample in the local dataset $\mathcal{D}_k$, $\mathcal{L}\left(\boldsymbol{w}; \boldsymbol{x}_k^j, \boldsymbol{y}_k^j\right)$ is the loss of the jth sample, and $|\mathcal{D}|$ is the cardinality of set $\mathcal{D}$, i.e., the number of elements in $\mathcal{D}$.

The goal of FL is to minimize

$$L(\boldsymbol{w}) = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} L_k(\boldsymbol{w}) \tag{2}$$

A. MODEL COMPRESSION

To adapt to the varying participating clients, we consider the sketched update that includes probabilistic quantization and Rand-M sparsification without error compensation [4]. The relationship between the model vectors before and after compression can be expressed as

$$\tilde{\boldsymbol{w}} = \Phi(\boldsymbol{w}, \rho, b) = \bar{\boldsymbol{w}} \odot \boldsymbol{s} \tag{3}$$

where $\Phi(\cdot)$ is the compression function, $b \in [1, 32]$ is the number of quantization bits, $\rho = M/N_{\text{model}}$ is the sparsity ratio satisfying $\rho \in [0, 1]$, $\boldsymbol{w}$ is the model vector before compressing, $\bar{\boldsymbol{w}}$ is the model vector only after quantization, $\tilde{\boldsymbol{w}}$ is the model vector after both quantization and sparsification, $\boldsymbol{s}$ is a binary vector that denotes the pre-defined random sparsity pattern (i.e., a random mask), and $\odot$ denotes the element-wise multiplication operation.

For b-bit probabilistic quantization, it is necessary to first equally divide the range of the model parameters $[w_{\min}, w_{\max}]$ into $2^b - 1$ intervals, where $w_{\min} = \min_j\{w_j\}$ and $w_{\max} = \min_j\{w_j\}$, and w_j denotes the jth element of the model vector. Then, the nth interval can be expressed as $[q_n, q_{n+1}]$, where $n = 1, \cdots, 2^b - 1$, $q_n = w_{\min} + (n-1)\Delta$ and $\Delta = (w_{\max} - w_{\min})/(2^b - 1)$. When w_j falls in the nth interval, i.e., $q_n < w_j \le q_{n+1}$, the jth element becomes

$$\bar{w}_j = \begin{cases} q_n, & \text{with probability } \dfrac{q_{n+1} - w_j}{\Delta} \\ q_{n+1}, & \text{with probability } \dfrac{w_j - q_n}{\Delta} \end{cases} \tag{4}$$

For Rand-M sparsification, M elements are randomly selected from $\boldsymbol{s}$ and set to one, and the rest are set to zero. The sparse pattern $\boldsymbol{s}$ can be fully specified by a random seed, and therefore it is only required to send the nonzeros model parameters after sparsification [4].

It is worth noting that b has a limited range for adjustment, because in most cases 8-bit quantization is sufficient for not degrading performance, whereas $1 \sim 2$-bit quantization may degrade performance significantly [4]. As a result, rather than adjusting both ρ and b at the same time, we only adjust ρ with given value of b.

In the rth round, the BS first compresses the global model and then broadcasts the compressed global model $\tilde{\boldsymbol{w}}^{(r)} = \Phi\left(\boldsymbol{w}^{(r)}, \rho^{(r)}, b_{\text{down}}\right)$, where $\boldsymbol{w}^{(r)}$ is the global model before compression, $\rho^{(r)}$ is the sparsity ratio of the global model, and b_{down} is the number of bits for downlink quantization. The downlink communication overhead in the rth round is

$$D^{(r)} = \rho^{(r)} b_{\text{down}} N_{\text{model}} \tag{5}$$

We use FedAvg to reduce the total number of communication rounds. For client k, the local model vector at the ith iteration in the rth round can be obtained as

$$\boldsymbol{w}_k^{(r,i)} = \boldsymbol{w}_k^{(r,i-1)} + \mu \frac{\partial L_k\left(\boldsymbol{w}_k^{(r,i-1)}\right)}{\partial \boldsymbol{w}_k^{(r,i-1)}}, \quad i = 1, \cdots, N_{\text{iter}} \tag{6}$$

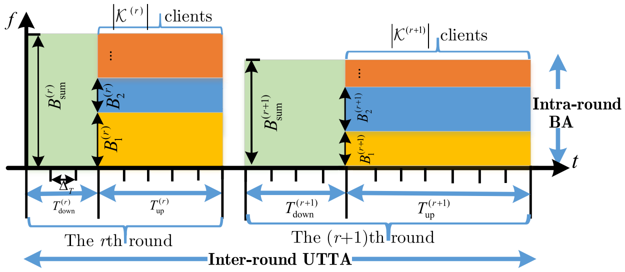


FIGURE 1. Communication resource allocation.

where μ is the learning rate, N_{iter} is the number of local iterations, $\mathcal{K}^{(r)} \subseteq \mathcal{K}$ is the set of participating clients in the rth round, and $\boldsymbol{w}_k^{(r,0)} = \tilde{\boldsymbol{w}}^{(r)}$.

After N_{iter} iterations, the local model update becomes $\nabla \boldsymbol{w}_k^{(r)} = \boldsymbol{w}_k^{(r,N_{\text{iter}})} - \boldsymbol{w}_k^{(r,0)}$. The compressed model update can be expressed by

$$\nabla \tilde{\boldsymbol{w}}_k^{(r)} = \Phi\left(\nabla \boldsymbol{w}_k^{(r)}, \rho_k^{(r)}, b_{\text{up}}\right) = \nabla \bar{\boldsymbol{w}}_k^{(r)} \odot \boldsymbol{s}_k \tag{7}$$

where $\rho_k^{(r)}$ is the sparsity ratio of client k, b_{up} is the number of bits for uplink quantization, $\nabla \bar{\boldsymbol{w}}_k^{(r)}$ is the local model update only with quantization, and $\boldsymbol{s}_k$ is the random sparsity pattern for client k.

The uplink communication cost required by client k is

$$D_k^{(r)} = \rho_k^{(r)} b_{\text{up}} N_{\text{model}} \tag{8}$$

The BS aggregates the model updates of all clients as $\nabla \tilde{\boldsymbol{w}}^{(r+1)} = \sum_{k \in \mathcal{K}^{(r)}} \nabla \tilde{\boldsymbol{w}}_k^{(r)} / \left|\mathcal{K}^{(r)}\right|$ to update the global model in the next round as

$$\begin{aligned} \boldsymbol{w}^{(r+1)} &= \boldsymbol{w}^{(r)} + \nabla \tilde{\boldsymbol{w}}^{(r+1)} \\ &= \boldsymbol{w}^{(r)} + \frac{1}{\left|\mathcal{K}^{(r)}\right|} \sum_{k \in \mathcal{K}^{(r)}} \Phi\left(\nabla \boldsymbol{w}_k^{(r)}, \rho_k^{(r)}, b_{\text{up}}\right) \end{aligned} \tag{9}$$

The global model converges when the following condition is satisfied

$$\frac{L\left(\boldsymbol{w}^{(r+1)}\right) - L\left(\boldsymbol{w}^*\right)}{L\left(\boldsymbol{w}^*\right)} \leq \varepsilon \tag{10}$$

where $\boldsymbol{w}^*$ is the optimal model vector of FL without model compression, and ε is the acceptable relative error to $\boldsymbol{w}^*$. Therefore, the number of rounds required for FL is

$$N_{\text{round}} = \arg\min_{r \in \mathbb{Z}} \left\{ r \middle| L(\boldsymbol{w}^{(r+1)}) \leq L_\varepsilon \right\} \tag{11}$$

where $L_\varepsilon = (1 + \varepsilon) L\left(\boldsymbol{w}^*\right)$ is the performance required for model convergence.

B. COMMUNICATION RESOURCE ALLOCATION

In Fig. 1, we show the communication resources in the considered scheme, where the bandwidth and the transmission time can change round by round. Let $B_{\text{sum}}^{(r)}$ and $T^{(r)} = T_{\text{down}}^{(r)} + T_{\text{up}}^{(r)}$ denote the total bandwidth and transmission time in the rth round, respectively, where $T_{\text{down}}^{(r)}$ and $T_{\text{up}}^{(r)}$ denote downlink and uplink transmission time, respectively.

We divide the round duration into multiple frames, each with a duration of Δ_T (in second-scale), and assume that the average channel gain (i.e., the large-scale channel gain including pathloss and shadowing) remains constant in each frame but may vary among frames. The transmission durations in the uplink and downlink contain $N_{\text{up}}^{(r)} = \lceil T_{\text{up}}^{(r)} / \Delta_T \rceil$ and $N_{\text{down}}^{(r)} = \lceil T_{\text{down}}^{(r)} / \Delta_T \rceil$ frames, respectively.

In the downlink transmission, the data rate per unit bandwidth in the tth frame is

$$R^{(r,t)} = \log_2\left(1 + \frac{P_{\text{BS}}}{N_0} \min_{k \in \mathcal{K}^{(r)}} \left\{h_k^{(r,t)}\right\}\right) \tag{12}$$

where P_{BS} is the transmit power of the BS per unit bandwidth, $h_k^{(r,t)}$ is the average channel gain from the BS to client k, and N_0 is the noise power spectral density.

Given the total available bandwidth $B_{\text{sum}}^{(r)}$ and the downlink transmission time $T_{\text{down}}^{(r)}$, the total number of bits in the downlink transmission is $C^{(r)} = \sum_{t=1}^{N_{\text{down}}^{(r)}} B_{\text{sum}}^{(r)} R^{(r,t)} \Delta_T$. Then, the maximal downlink sparsity ratio is

$$\begin{aligned} \rho^{(r)} &= \min\left\{C^{(r)} / (b_{\text{down}} N_{\text{model}}), 1\right\} \\ &= \min\left\{\frac{B_{\text{sum}}^{(r)} \Delta_T}{b_{\text{down}} N_{\text{model}}} \sum_{t=1}^{N_{\text{down}}^{(r)}} R^{(r,t)}, 1\right\} \end{aligned} \tag{13}$$

In the uplink transmission, clients use OFDMA to upload their model updates and avoid interference among clients. Each client is allocated non-overlapping bandwidth, as shown in Fig. 1. Let $B_k^{(r,t)}$ denote the allocated bandwidth to client k in the tth frame, then we have $\sum_{k \in \mathcal{K}^{(r)}} B_k^{(r,t)} \leq B_{\text{sum}}^{(r)}$, $t = 1, \cdots, N_{\text{up}}^{(r)}$, i.e., the sum of allocated bandwidths to all clients should not exceed the total available bandwidth $B_{\text{sum}}^{(r)}$ in each frame.

For client k, the data rate per unit bandwidth in the tth frame is

$$R_k^{(r,t)} = \log_2\left(1 + \frac{P_{\text{UE}}}{N_0} h_k^{(r,t)}\right) \tag{14}$$

where P_{UE} is the transmit power of each client per unit bandwidth.

Given the allocated bandwidth $B_k^{(r,t)}$ and the uplink transmission time $T_{\text{up}}^{(r)}$, the total number of bits transmitted by client k is $C_k^{(r)} = \sum_{t=1}^{N_{\text{up}}^{(r)}} B_k^{(r,t)} R_k^{(r,t)} \Delta_T$. Then, the maximal uplink sparsity ratio of client k is

$$\begin{aligned} \rho_k^{(r)} &= \min\left\{C_k^{(r)} / (b_{\text{up}} N_{\text{model}}), 1\right\} \\ &= \min\left\{\frac{\Delta_T}{b_{\text{up}} N_{\text{model}}} \sum_{t=1}^{N_{\text{up}}^{(r)}} B_k^{(r,t)} R_k^{(r,t)}, 1\right\} \end{aligned} \tag{15}$$

Since the total communication time can reflect the consumed communication resources by FL, we minimize the total communication time instead of the total training time, i.e.,

$$T_{\text{tot}} = \sum_{r=1}^{N_{\text{round}}} T^{(r)} = \sum_{r=1}^{N_{\text{round}}} \left(T_{\text{down}}^{(r)} + T_{\text{up}}^{(r)}\right) \tag{16}$$

III. FLEXIBLE MODEL COMPRESSION AND RESOURCE ALLOCATION

In this section, we introduce a flexible model compression and resource allocation scheme. We first formulate the optimization problem and decompose it into two subproblems, i.e., the BA optimization with given uplink transmission time and the UTTA optimization with the optimal BA. Then, we learn to optimize BA and UTTA for adapting to time-varying mobile networks. Finally, we summarize the proposed method.

A. PROBLEM FORMULATION

It is well understood that increasing the downlink and uplink sparsity ratios $\rho^{(r)}$ and $\rho_k^{(r)}$ can reduce the per-round duration $T^{(r)}$ but increase the number of communication rounds N_{round}. To minimize total communication time, the sparsity ratios must be optimized to balance $T^{(r)}$ and N_{round}. Because the sparsity ratios are determined by the allocated transmission time $T_{\text{down}}^{(r)}$ and $T_{\text{up}}^{(r)}$ as well as the allocated bandwidth $B_k^{(r,t)}$, as shown in (13) and (15), we need to optimize the intra-round BA and inter-round UTTA jointly to minimize total communication time. The optimization problem is formulated as follows

$$\mathbf{P}_1: \min_{T_{\text{down}}^{(r)}, T_{\text{up}}^{(r)}, \{B_k^{(r,t)}\}_{t=1}^{N_{\text{up}}^{(r)}}} T_{\text{tot}} = \sum_{r=1}^{N_{\text{round}}} \left(T_{\text{down}}^{(r)} + T_{\text{up}}^{(r)}\right) \tag{17a}$$

$$\begin{aligned} s.t.\ & T_{\text{down}}^{(r)},\ T_{\text{up}}^{(r)} \in \{n\Delta_T | n \in \mathbb{Z}\}, \\ & r = 1, \cdots, N_{\text{round}} && (17\text{b}) \\ & B_k^{(r,t)} \geq 0,\ t = 1, \cdots, N_{\text{up}}^{(r)} && (17\text{c}) \\ & \sum_{k \in \mathcal{K}^{(r)}} B_k^{(r,t)} \leq B_{\text{sum}}^{(r)},\ r = 1, \cdots, N_{\text{round}} \\ & && (17\text{d}) \\ & L(\mathbf{w}^{(N_{\text{round}}+1)}) \leq L_\varepsilon && (17\text{e}) \end{aligned}$$

Problem $\mathbf{P}_1$ is hard to solve for the following reasons. 1) **Unknown parameter** N_{round} in dynamic scenarios, 2) **Unavailable channel information** $h_k^{(r,t)}$, $r = 1, \cdots, N_{\text{round}}$ at the start of FL, and 3) **Unaffordable computation complexity** for practical systems by optimizing $2N_{\text{round}} + \sum_{r=1}^{N_{\text{round}}} |\mathcal{K}^{(r)}| N_{\text{up}}^{(r)}$ variables jointly. To solve the problems, we decompose a multi-round joint optimization problem into multiple single-round optimization problems. To achieve the performance close to the joint optimization, we find an appropriate objective function that can balance $T^{(r)}$ and N_{round} during one round.

In Fig. 2, we illustrate the relationship between the loss function and the total communication time of FL. When the global model converges, the cumulative loss decrement and communication time are $L(\mathbf{w}^{(1)}) - L_\varepsilon = \sum_{r=1}^{N_{\text{round}}} \Delta L^{(r)}$ and $\sum_{r=1}^{N_{\text{round}}} T^{(r)}$, respectively, where $\Delta L^{(r)} = L(\mathbf{w}^{(r)}) - L(\mathbf{w}^{(r+1)})$ is the loss decrement in the rth round.

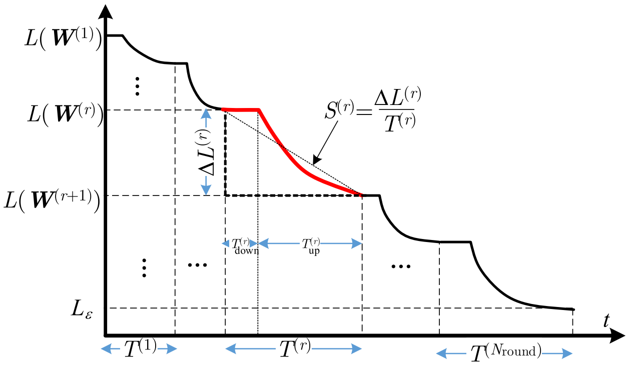


FIGURE 2. **Relationship between loss function and total communication time.**

Problem $\mathbf{P}_1$ aims to minimize $\sum_{r=1}^{N_{\text{round}}} T^{(r)}$ with given $\sum_{r=1}^{N_{\text{round}}} \Delta L^{(r)}$, which is equivalent to maximizing the *convergence speed* in the FL. We can express the convergence speed as

$$S \triangleq \frac{\sum_{r=1}^{N_{\text{round}}} \Delta L^{(r)}}{\sum_{r=1}^{N_{\text{round}}} T^{(r)}} \tag{18}$$

which can be further expressed as

$$S = \sum_{r=1}^{N_{\text{round}}} \beta^{(r)} S^{(r)} = S_{\text{past}}^{(r)} + \beta^{(r)} S^{(r)} + S_{\text{future}}^{(r)} \tag{19}$$

where $S^{(r)} = \Delta L^{(r)}/T^{(r)}$ is the convergence speed in the rth round, $\beta^{(r)} = T^{(r)}/\sum_{r=1}^{N_{\text{round}}} T^{(r)}$ is the proportion of transmission time for the rth round in the total transmission time, $S_{\text{past}}^{(r)} = \sum_{\tau=1}^{r-1} \beta^{(\tau)} S^{(\tau)}$ is the weighted convergence speed in the previous $r-1$ rounds, and $S_{\text{future}}^{(r)} = \sum_{\tau=r+1}^{N_{\text{round}}} \beta^{(\tau)} S^{(\tau)}$ is the weighted convergence speed in the subsequent $N_{\text{round}} - r$ rounds.

As shown in (19), the total convergence speed is the weighted sum of the past, present, and future convergence speeds. However, in the rth round, the past speed $S_{\text{past}}^{(r)}$ remains unchanged, and the future speed $S_{\text{future}}^{(r)}$ is unknown due to the unavailable channel information. Therefore, we can only maximize the present convergence speed $S^{(r)}$. It is important to note that $S^{(r)}$ is a random variable affected by random sparsification. Thus, the objective function is chosen as the statistically average convergence speed

$$J = \mathbb{E}_s\left\{S^{(r)}\right\} = \frac{\mathbb{E}_s\left\{\Delta L^{(r)}\right\}}{T^{(r)}} \tag{20}$$

where $\mathbb{E}_s$ denotes the expectation over the randomness of sparsity patterns.

Compared with the model update vector $\nabla \mathbf{w}_k$, the model vector $\mathbf{w}_k$ has many non-zero elements and is sensitive to the model sparsification. Moreover, the BS has more communication resources (e.g., more bandwidth and higher transmit power) to transmit the global model than the users, hence the model sparsification can not save much downlink communication time. Therefore, we take $\rho^{(r)} = 1$ into account and allocate downlink communication time to satisfy $C^{(r)} \geq b_{\text{down}} N_{model}$. Then, we only need to maximize $\mathbb{E}_s\left\{S^{(r)}\right\}$ by optimizing $T_{\text{up}}^{(r)}$ and $\{B_k^{(r,t)}\}$. The single-round optimization problem becomes

$$\mathbf{P}_2: \quad \max_{T_{\text{up}}^{(r)}, \{B_k^{(r,t)}\}_{t=1}^{N_{\text{up}}^{(r)}}} \quad J = \frac{\mathbb{E}_s\left\{\Delta L^{(r)}\right\}}{T^{(r)}} \tag{21a}$$

$$s.t.\ T_{\text{up}}^{(r)} \in \{n\Delta_T | n \in \mathbb{Z}\} \tag{21b}$$

$$B_k^{(r,t)} \geq 0, \quad t = 1, \cdots, N_{\text{up}}^{(r)} \tag{21c}$$

$$\sum_{k \in \mathcal{K}^{(r)}} B_k^{(r,t)} \leq B_{\text{sum}}^{(r)} \tag{21d}$$

Unlike $\mathbf{P}_1$, Problem $\mathbf{P}_2$ does not depend on N_{round} and only requires the channel information in the current round. Moreover, the number of variables to be optimized becomes $1 + |\mathcal{K}^{(r)}| N_{\text{up}}^{(r)}$, which reduces the computation complexity dramatically.

Because $T_{\text{up}}^{(r)}$ and $\{B_k^{(r,t)}\}$ have different time scales and are hard to optimize simultaneously, we further decompose $\mathbf{P}_2$ into two sub-problems $\mathbf{P}_3$ and $\mathbf{P}_4$. Specifically, $\mathbf{P}_3$ optimizes the intra-round BA to maximize $\mathbb{E}_s\left\{\Delta L^{(r)}\right\}$ with given $T_{\text{up}}^{(r)}$, and then $\mathbf{P}_4$ optimizes the inter-round UTTA to maximize $\mathbb{E}_s\left\{S^{(r)}\right\}$ with the optimal BA, as to be detailed in the following subsections.

B. BA OPTIMIZATION

1) PROBLEM FORMULATION AND OPTIMAL BA

Since the allocated bandwidth determines the sparsity ratio of each client $\rho_k^{(r)}$, we first investigate the impact of $\rho_k^{(r)}$ on the average loss decrement in each round $\mathbb{E}_s\left\{\Delta L^{(r)}\right\}$.

In the rth round, the loss decrement $\Delta L^{(r)} = L(\boldsymbol{w}^{(r)}) - L(\boldsymbol{w}^{(r+1)})$ can be expressed as

$$\begin{aligned}\Delta L^{(r)} &= \left(L(\boldsymbol{w}^{(r)}) - L(\bar{\boldsymbol{w}}^{(r+1)})\right) \\ &\quad - \left(L(\boldsymbol{w}^{(r+1)}) - L(\bar{\boldsymbol{w}}^{(r+1)})\right) \\ &= \Delta L_{\text{max}}^{(r)} - \Delta L_{\text{S}}^{(r)}\end{aligned} \tag{22}$$

where $\bar{\boldsymbol{w}}^{(r+1)} = \boldsymbol{w}^{(r)} + \sum_{k \in \mathcal{K}^{(r)}} \nabla \bar{\boldsymbol{w}}_k^{(r)} / \left|\mathcal{K}^{(r)}\right|$ is the global model vector without sparsization and $\boldsymbol{w}^{(r+1)} = \boldsymbol{w}^{(r)} + \sum_{k \in \mathcal{K}^{(r)}} \nabla \tilde{\boldsymbol{w}}_k^{(r)} / \left|\mathcal{K}^{(r)}\right| = \boldsymbol{w}^{(r)} + \sum_{k \in \mathcal{K}^{(r)}} \nabla \tilde{\boldsymbol{w}}_k^{(r)} \odot \boldsymbol{s}_k / \left|\mathcal{K}^{(r)}\right|$ is the global model vector after sparsization, $\Delta L_{\text{max}}^{(r)} = L(\boldsymbol{w}^{(r)}) - L(\bar{\boldsymbol{w}}^{(r+1)})$ is the maximal loss decrement without sparsization, and $\Delta L_{\text{S}}^{(r)} = L(\boldsymbol{w}^{(r+1)}) - L(\bar{\boldsymbol{w}}^{(r+1)})$ is the performance loss caused by sparsification.

From (22), the average loss decrement can be expressed as $\mathbb{E}_s\left\{\Delta L^{(r)}\right\} = \mathbb{E}_s\left\{\Delta L_{\text{max}}^{(r)}\right\} - \mathbb{E}_s\left\{\Delta L_{\text{S}}^{(r)}\right\}$. Since $\Delta L_{\text{max}}^{(r)}$ is not affected by the sparsification methods, we have $\mathbb{E}_s\left\{\Delta L_{\text{max}}^{(r)}\right\} = \Delta L_{\text{max}}^{(r)}$. Additionally, $\Delta L_{\text{S}}^{(r)}$ is always larger than or equal to zero, which can be expressed as $\Delta L_{\text{S}}^{(r)} = \left|\Delta L_{\text{S}}^{(r)}\right|$. Therefore, the average the loss decrement becomes

$$\mathbb{E}_s\left\{\Delta L^{(r)}\right\} = \Delta L_{\text{max}}^{(r)} - \mathbb{E}_s\left\{\left|\Delta L_{\text{S}}^{(r)}\right|\right\} \tag{23}$$

This indicates that to maximize $\mathbb{E}_s\left\{\Delta L^{(r)}\right\}$, we need to minimize the performance loss caused by sparsification $\mathbb{E}_s\left\{\left|\Delta L_{\text{S}}^{(r)}\right|\right\}$.

In FL, the learning model is usually nonlinear, such as DNNs, which makes it challenging to derive a closed-form expression for $\mathbb{E}_s\left\{\Delta L^{(r)}\right\}$. To address this, we find an upper bound on the performance loss by using Lipschitz continuity. Specifically, when the loss function is Lipschitz continuous [18], we have the following inequality

$$\mathbb{E}_s\left\{\left|\Delta L_{\text{S}}^{(r)}\right|\right\} \leq \alpha \mathbb{E}_s\left\{\left\|\boldsymbol{E}_{\text{S}}^{(r)}\right\|_p\right\} \tag{24}$$

where α is the local Lipschitz constant, $\boldsymbol{E}_{\text{S}}^{(r)} = \boldsymbol{w}^{(r+1)} - \bar{\boldsymbol{w}}^{(r+1)}$ is the model error caused by random sparsification, $\|\cdot\|_p$ is the L_p-norm of a vector, and p a non-negative integer.

This suggests that the performance loss can be reduced by minimizing the model error introduced by sparsification. In the following, we further to analyze the impact of $\rho_k^{(r)}$ on the model error.

In (24), Lipschitz continuity is defined using the L_p-norm, where the commonly used norms include the L_1-norm and the L_2-norm. The L_1-norm is the sum of the absolute values of the elements in a vector, while the L_2-norm is the square root of the sum of squared elements of the vector. Compared to the L_2-norm, the L_1-norm is more suitable for capturing the sparsity of a vector, which allows us to conveniently analyze the impact of random sparsification on the model error. Therefore, we consider the L_1-norm.

Proposition 1: When the loss function is Lipschitz continuous with the L_1-norm, and the L_1-norm of the model updates of all clients are bounded, the model error caused by random sparsification satisfies

$$\mathbb{E}_s\left\{\left\|\boldsymbol{E}_{\text{S}}^{(r)}\right\|_1\right\} \leq \left\|\nabla \boldsymbol{w}_{\text{max}}^{(r)}\right\| \left(1 - \bar{\rho}^{(r)}\right) \tag{25}$$

and the upper bound of the performance loss becomes

$$\mathbb{E}_s\left\{\left|\Delta L_{\text{S}}^{(r)}\right|\right\} \leq \alpha_1^{(r)} \left\|\nabla \boldsymbol{w}_{\text{max}}^{(r)}\right\| \left(1 - \bar{\rho}^{(r)}\right) \tag{26}$$

where $\alpha_1^{(r)}$ is the local Lipschitz constant in the rth round with the L_1-norm, $\left\|\nabla \boldsymbol{w}_{\text{max}}^{(r)}\right\| = \max_{k \in \mathcal{K}^{(r)}} \left\|\nabla \boldsymbol{w}_k^{(r)}\right\|_1$ is the maximal L_1-norm of all clients' model updates, and $\bar{\rho}^{(r)} = \sum_{k \in \mathcal{K}^{(r)}} \rho_k^{(r)} / \left|\mathcal{K}^{(r)}\right|$ is the average sparsity ratio of all clients.

Proof: See Appendix A. ■

The tightness of the upper bound in (25) depends on the differences in the norms of the client model updates. We can measure it by the ratio of the minimal and maximal L_1-norms, denoted as $\lambda^{(r)} = \left\|\nabla \boldsymbol{w}_{\text{min}}^{(r)}\right\| / \left\|\nabla \boldsymbol{w}_{\text{max}}^{(r)}\right\|$, where $\left\|\nabla \boldsymbol{w}_{\text{min}}^{(r)}\right\| = \min_{k \in \mathcal{K}^{(r)}} \left|\nabla \boldsymbol{w}_k^{(r)}\right|_1$ is the minimal L_1-norm among all clients' model updates. When $\lambda^{(r)}$ is close to one, the upper bound is tight. We will evaluate the values of $\lambda^{(r)}$ in the forthcoming simulation to show the tightness of this upper bound.

The tightness of the upper bound in (26) also depends on the value of the Lipschitz constant. Specifically, a small Lipschitz constant tightens the upper bound. To achieve a tighter upper bound, we introduce the local Lipschitz constant in each round $\alpha_1^{(r)}$ instead of the global Lipschitz constant in (24). Additionally, this Lipschitz constant also reflects the impact of model sparsification on the learning performance. When sparsification leads to a significant performance degradation, the Lipschitz constant tends to be large, otherwise the Lipschitz constant is small. Through analysis and simulation results in [4] and [8], it is clear that to find a balance between the number of rounds and per-round latency, the chosen sparsity ratios generally do not cause a significant performance loss. Therefore, when we optimize the sparsity ratio to maximize convergence speed, it is reasonable to assume that the Lipschitz constant is small and the upper bound is tight.

Minimizing this upper bound of the performance loss is a conservative design, which ensures the performance loss to be always within the bounds.

Proposition 1 indicates that one can maximize the average sparsity ratio $\bar{\rho}^{(r)} = \sum_{k\in\mathcal{K}^{(r)}} \rho_k^{(r)} / \left|\mathcal{K}^{(r)}\right|$ to reduce $\mathbb{E}_s\left\{\left|\Delta L_{\mathrm{S}}^{(r)}\right|\right\}$. Obviously, $\rho_k^{(r)}$ depends on the allocated bandwidth in all frames. To maximize $\bar{\rho}^{(r)}$ by optimizing BA in multiple frames, the BS should know the channel information in all frames at the start of each round. However, future channels of mobile clients are unknown. Consequently, we maximize the cumulative sparsity ratio until the current frame $\rho^{(r,t)} = \sum_{k\in\mathcal{K}^{(r)}} \rho_k^{(r,t)}$, where $\rho_k^{(r,t)}$ is the cumulative sparsity ratio of client k from the first frame until the tth frame. From (15), we have

$$\begin{aligned}\rho_k^{(r,t)} &= \min\left\{\sum_{\tau=1}^{t} \Delta\rho_k^{(r,\tau)}, 1\right\}\\ &= \min\left\{\rho_k^{(r,t-1)} + \Delta\rho_k^{(r,t)}, 1\right\}\end{aligned} \tag{27}$$

where $\Delta\rho_k^{(r,t)} = B_k^{(r,t)} R_k^{(r,t)} \Delta_T / \left(b_{\mathrm{up}} N_{\mathrm{model}}\right)$ is the increment of sparsity ratio in the tth frame, and $\rho_k^{(r,t-1)}$ is the cumulative sparsity ratio in the past frames, which is independent with the current BA method.

We try to remove the minimization operation in (27) because it makes $\rho_k^{(r,t)}$ hard to analyze and optimize. To meet $\rho_k^{(r,t-1)} + \Delta\rho_k^{(r,t)} \leq 1$, i.e., $\Delta\rho_k^{(r,t)} \leq \left(1 - \rho_k^{(r,t-1)}\right)$, the clients who have enough bandwidth to upload all model updates are no longer allocated more bandwidth. As a result, the allocated bandwidth for client k should satisfy

$$B_k^{(r,t)} \leq \frac{b_{\mathrm{up}} N_{\mathrm{model}} \left(1 - \rho_k^{(r,t-1)}\right)}{R_k^{(r,t)} \Delta_T}, \quad \forall k \in \mathcal{K}^{(r,t)} \tag{28}$$

where $\mathcal{K}^{(r,t)} = \{k | \rho_k^{(r,t-1)} < 1, k \in \mathcal{K}^{(r)}\}$ is the set of clients who has data to be transmitted in the tth frame.

When (28) holds, the cumulative sparsity ratio of client k becomes $\rho_k^{(r,t)} = \sum_{\tau=1}^{t} \Delta\rho_k^{(r,\tau)}$, and the sum of cumulative sparsity ratios becomes $\rho^{(r,t)} = \sum_{k\in\mathcal{K}^{(r)}} \rho_k^{(r,t)} = \rho^{(r,t-1)} + \Delta\rho^{(r,t)}$, where

$$\Delta\rho^{(r,t)} = \sum_{k\in\mathcal{K}^{(r,t)}} \Delta\rho_k^{(r,t)} = \sum_{k\in\mathcal{K}^{(r,t)}} \frac{B_k^{(r,t)} R_k^{(r,t)} \Delta_T}{b_{\mathrm{up}} N_{\mathrm{model}}} \tag{29}$$

We can select $\Delta\rho^{(r,t)}$ as the optimization objective to maximize $\rho^{(r,t)}$. The BA optimization problem can then be formulated as

$$\mathbf{P}_3 : \max_{\{B_k^{(r,t)}\}} \Delta\rho^{(r,t)} = \sum_{k\in\mathcal{K}^{(r,t)}} \frac{B_k^{(r,t)} R_k^{(r,t)} \Delta_T}{b_{\mathrm{up}} N_{\mathrm{model}}} \tag{30a}$$

$$s.t.\ 0 \leq B_k^{(r,t)} \leq \frac{b_{\mathrm{up}} N_{\mathrm{model}} \left(1 - \rho_k^{(r,t-1)}\right)}{R_k^{(r,t)} \Delta_T}, \ \forall k \in \mathcal{K}^{(r,t)} \tag{30b}$$

$$\sum_{k\in\mathcal{K}^{(r,t)}} B_k^{(r,t)} \leq B_{\mathrm{sum}}^{(r)} \tag{30c}$$

Since Problem $\mathbf{P}_3$ is a linear programming with linear constraints, we can use a simplex algorithm to obtain the optimal solution. From (30a), we know that the optimal solution always allocates the bandwidth to the client with the largest channel gain. If there is remaining bandwidth after allocation, the bandwidth will be allocated to the one with the best channel quality among other clients. Repeat this allocation until all bandwidth is allocated.

When the bandwidth of each frame is only allocated to one client, the closed-form expression of the optimal BA is obtained as follows

$$B_{\mathrm{Max},k}^{(r,t)} = \begin{cases} B_{\mathrm{sum}}^{(r)}, & \forall k^* = \arg\max_{k\in\mathcal{K}^{(r)}} \left\{h_k^{(r,t)}\right\} \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Hence, we call it the **maximal gain BA (Max-BA)**.

2) PERFORMANCE ANALYSIS

In the following, we analyze the performance of Max-BA. To show the performance gain of the optimal BA, we consider two classical allocation methods as baselines.

- **Equal BA (Equ-BA)**: the bandwidth is equally allocated to each client, i.e.,

$$B_{\mathrm{Equ},k}^{(r,t)} = \frac{B_{\mathrm{sum}}^{(r)}}{\left|\mathcal{K}^{(r)}\right|} \tag{32}$$

which does not require the channel information of any client.

- **Rate inversion BA (Inv-BA)**: To ensure that all clients have the same sparsity ratio, the allocated bandwidth is inversely proportional to the data rate, i.e.,

$$B_{\mathrm{Inv},k}^{(r,t)} = \left(\frac{1}{\left|\mathcal{K}^{(r)}\right|} \sum_{i\in\mathcal{K}^{(r)}} \frac{1}{R_i^{(r,t)}}\right)^{-1} \frac{B_{\mathrm{sum}}^{(r)}}{R_k^{(r,t)}} \tag{33}$$

TABLE 1. Order of Mean for different BA methods.

BA methods	Order of mean	Generalized mean
Max-BA	$p_{\text{Max}-\text{BA}} \to \infty$	Maximum
Equ-BA	$p_{\text{Equ}-\text{BA}} = 1$	Arithmetic mean
Inv-BA	$p_{\text{Inv}-\text{BA}} = -1$	Harmonic mean

To provide the close-form expression of $\bar{\rho}^{(r)}$, we focus on the case where all clients have balanced sparsity ratios, i.e., $0 < \rho_k^{(r)} < 1 \; \forall k \in \mathcal{K}^{(r)}$. Then, $\mathcal{K}^{(r,t)}$ remains unchanged among frames, i.e., $\mathcal{K}^{(r,t)} = \mathcal{K}^{(r)}$.

Proposition 2: When the sparsity ratios of all clients satisfy $0 < \rho_k^{(r)} < 1$, $\forall k \in \mathcal{K}^{(r)}$, the average sparsity ratios with Max-BA, Equ-BA, and Inv-BA methods can be expressed as

$$\bar{\rho}^{(r)} = \nu^{(r)} \sum_{t=1}^{N_{\text{up}}^{(r)}} \left(\frac{1}{\left|\mathcal{K}^{(r)}\right|} \sum_{k \in \mathcal{K}^{(r)}} \left(R_k^{(r,t)} \right)^p \right)^{\frac{1}{p}} \tag{34}$$

where $\nu^{(r)} = B_{\text{sum}}^{(r)} \Delta_T / (\left|\mathcal{K}^{(r)}\right| b_{\text{up}} N_{\text{model}})$ and p is the order of mean and its value is listed in Table 1.

Proof: See Appendix V-B. ■

According to the monotonicity of the generalized mean, we can compare the performance of different BA methods as follows.

Corollary 1: Since $p_{\text{Max}-\text{BA}} > p_{\text{Equ}-\text{BA}} > p_{\text{Inv}-\text{BA}}$ from Table 1, we always have

$$\bar{\rho}_{\text{Max}-\text{BA}}^{(r)} \geq \bar{\rho}_{\text{Equ}-\text{BA}}^{(r)} \geq \bar{\rho}_{\text{Inv}-\text{BA}}^{(r)} \tag{35}$$

where equality holds if and only if $R_k^{(r,t)} = \beta$, $\forall k \in \mathcal{K}^{(r)}$, $t = 1, \cdots, N_{\text{up}}^{(r)}$, and β is a constant. When the different between $R_k^{(r,t)}$ gets larger, the gap between $\bar{\rho}_{\text{Max}-\text{BA}}^{(r)}$, $\bar{\rho}_{\text{Equ}-\text{BA}}^{(r)}$, and $\bar{\rho}_{\text{Inv}-\text{BA}}^{(r)}$ also increases.

According to Corollary 1, Max-BA always achieves the maximal average sparsity ratio, Equ-BA is in the middle, and Inv-BA achieves the minimum.

Since Inv-BA in (33) is the optimal BA to maximize the minimal sparsity ratio $\rho_{\min}^{(r)} = \min_{k \in \mathcal{K}^{(r)}} \rho_k^{(r)}$, it always allocates more bandwidth to clients with lowest channel gains. However, Proposition 1 indicates that the convergence speed depends on $\bar{\rho}^{(r)}$ rather than $\rho_{\min}^{(r)}$. Therefore, it is not necessary to force all clients to use the same sparsity ratio. If the clients are allowed to have different sparsity ratios, they can choose better BA methods to improve resource usage efficiency.

It is worthy to note that the Max-BA method always allocates all resources to the clients with the largest channel gain, resulting in the imbalanced sparsity ratios among all clients. In other words, some clients have higher sparsity ratios, while others have lower sparsity ratios. Due to this imbalance in sparsity ratios, the global model is only learned from the local datasets of a few clients, which degrades the learning performance of FL. However, in mobile networks, the channel gain varies among frames, hence the client with the largest channel gain may change across rounds. When the channel conditions of different clients fluctuate significantly, the client with the largest channel gain will be different in each round, giving each client an opportunity to be allocated with bandwidth. However, when the client with the largest channel gain remains the same in each round, we can use the suboptimal Equ-BA method to avoid the imbalance. Therefore, we consider both the Max-BA and Equ-BA methods in optimizing inter-round UTTA in the next subsection.

C. UTTA OPTIMIZATION

Given the BA method, Problem $\mathbf{P}_2$ can be simplified to

$$\mathbf{P}_4 : \max_{T_{\text{up}}^{(r)}} \; J = \frac{\mathbb{E}_s \left\{ L(\boldsymbol{w}^{(r)}) - L(\boldsymbol{w}^{(r+1)}) \right\}}{T_{\text{down}}^{(r)} + T_{\text{up}}^{(r)}} \tag{36a}$$

$$s.t. \; T_{\text{up}}^{(r)} \in \{ n\Delta_T | n \in \mathbb{Z} \}, \; \forall r \tag{36b}$$

In Problem $\mathbf{P}_4$, the convergence speed J depends on the optimization variable $T_{\text{up}}^{(r)}$ and the system and environmental parameters. Let $\boldsymbol{x}^{(r)}$ denote a vector that contains the related system and environmental parameters, the objective function can be expressed as $J = \mathcal{J}\left(\boldsymbol{x}^{(r)}, T_{\text{up}}^{(r)}\right)$. Then, the optimal uplink transmission time for given $\boldsymbol{x}^{(r)}$ is $T_{\text{up}}^{(r)*} = \arg\max_{T_{\text{up}}^{(r)}} \mathcal{J}\left(\boldsymbol{x}^{(r)}, T_{\text{up}}^{(r)}\right)$.

Our previous analysis has indicates that it is difficult to obtain the closed-form expression of $\mathcal{J}\left(\boldsymbol{x}^{(r)}, T_{\text{up}}^{(r)}\right)$. Although exhaustive searching can yield the optimal solution, solving $\mathbf{P}_4$ at the beginning of each round leads to unacceptable computational complexity and decision-making latency. To cope with these issues, we consider a learning-based policy network to obtain $T_{\text{up}}^{(r)*}$. The input-output relation of the policy network is expressed as

$$\hat{T}_{\text{up}}^{(r)*} = f_{\text{p}}\left(\boldsymbol{x}^{(r)}; \boldsymbol{\theta}_{\text{p}}\right) \tag{37}$$

where $\hat{T}_{\text{up}}^{(r)*}$ is the learned optimal uplink transmission time from the policy network, and $\boldsymbol{\theta}_{\text{p}}$ contains the model parameters of the policy network.

To find a UTTA policy in dynamic environments, $\boldsymbol{x}^{(r)}$ should contain the parameters changing in each round, such as the available bandwidth, the set of clients participating in FL, and the channel gains of all clients. Hence, the input vector is

$$\boldsymbol{x}^{(r)} = \left[B_{\text{sum}}^{(r)}, \mathcal{K}^{(r)}, \left\{ h_k^{(r,t)} | \forall k \in \mathcal{K}^{(r)}, t = 1, \cdots, N_{\text{up}}^{(r)} \right\} \right] \tag{38}$$

When the policy network is obtained by deep learning, it encounters these troubles.

1) If supervised learning is used to learn the policy network, one has to use exhaustive searching to find the optimal solution for generating the training samples $(\boldsymbol{x}^{(r)}, T_{\text{up}}^{(r)*})$. This causes high computational complexity in generating labels.

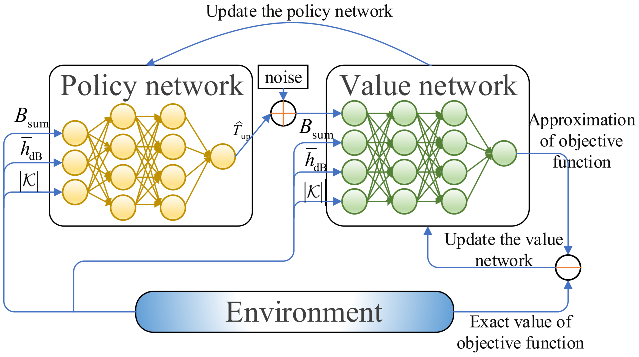


FIGURE 3. Structure of model-free unsupervised learning.

2) The dimension of input vector in (38) is $N_{\boldsymbol{x}} = 1 + |\mathcal{K}^{(r)}| + |\mathcal{K}^{(r)}|N_\mathrm{up}^{(r)}$, which depends not only on the varying number of clients $|\mathcal{K}^{(r)}|$, but also on the unknown parameter $N_\mathrm{up}^{(r)}$. It is with high cost to train such a DNN with the varying input dimension.
3) To obtain the input vector $\boldsymbol{x}^{(r)}$ in (38) at the start of each round, it is necessary to collect the channel information of all clients at all frames $\left\{h_k^{(r,t)}, t = 1, \cdots, N_\mathrm{up}^{(r)}\right\}$. However, the channel information from the second to the last frames is unknown.

In the following, we address these issues by resorting to the model-free unsupervised learning (MFUL) proposed in [19].

1) MODEL-FREE UNSUPERVISED LEARNING

The basic idea of MFUL, as shown in Fig. 3, is to introduce a value network to first estimate the objective function $\hat{J}$ and then use a policy network to obtain the optimal variable to maximize $\hat{J}$. The input-output relationship of the value network is denoted as

$$\hat{J} = f_\mathrm{v}\left(\boldsymbol{x}^{(r)}, T_\mathrm{up}^{(r)}; \boldsymbol{\theta}_\mathrm{v}\right) \tag{39}$$

where $\boldsymbol{\theta}_\mathrm{v}$ is the model vector of the value network. Here, "model-free" indicates that the objective function $J = \mathcal{J}\left(\boldsymbol{x}^{(r)}, T_\mathrm{up}^{(r)}\right)$ is unknown and estimated from the training data.

In $\mathbf{P}_4$, the constraint in (36b) is easy to satisfy by setting the output layer's activation function as ReLU and introducing a rounding down operation. We only need to optimize the value network to minimize the estimated error of objective $\hat{J}$, and then optimize the policy network to maximize $\hat{J}$. Consequently, using MUFL to solve $\mathbf{P}_4$ amounts to solving the following two optimization problems,

$$\mathbf{P}_5: \begin{cases} \min\limits_{\boldsymbol{\theta}_\mathrm{v}} \left(\mathcal{J}\left(\boldsymbol{x}^{(r)}, T_\mathrm{up}^{(r)}\right) - \hat{J}\right)^2 \\ s.t.\ \hat{J} = f_\mathrm{v}\left(\boldsymbol{x}^{(r)}, T_\mathrm{up}^{(r)}; \boldsymbol{\theta}_\mathrm{v}\right), \\ \max\limits_{\boldsymbol{\theta}_\mathrm{p}} f_\mathrm{v}\left(\boldsymbol{x}^{(r)}, \hat{T}_\mathrm{up}^{(r)}; \boldsymbol{\theta}_\mathrm{v}\right) \\ s.t.\ \hat{T}_\mathrm{up}^{(r)} = f_\mathrm{p}\left(\boldsymbol{x}^{(r)}; \boldsymbol{\theta}_\mathrm{p}\right) \end{cases} \tag{40}$$

According to (38) and (39), it is clear that the current UTTA policy $T_\mathrm{up}^{(r)}$ does not affect the future input vector and objective function $\boldsymbol{x}^{(r+1)}$ and $\mathbb{E}_{\boldsymbol{s}}\left\{S^{(r+1)}\right\}$. This indicates that Problem $\mathbf{P}_5$ is a typical non-Markov Decision Process (non-MDP) problem. As the MFUL can be viewed as a simplified version of reinforcement learning (RL) tailored to address non-MDP problems [19], we employ MFUL instead of RL.

2) MODEL INPUT SIMPLIFICATION

In order to keep the input dimension unchanged, we simplify the input of the policy network. According to Proposition 2, we can approximate the average sparsity ratio as follows.

Corollary 2: When the SNR is high enough such that $R_k^{(r,t)} \approx \log_2\left(P_\mathrm{UE}h_k^{(r,t)}/N_0\right)$, the average sparsity ratio can be approximated as

$$\bar{\rho}^{(r)} \approx \frac{B_\mathrm{sum}^{(r)}T_\mathrm{up}^{(r)}}{\left|\mathcal{K}^{(r)}\right| b_\mathrm{up}N_\mathrm{model}}\left(\log_2\left(\frac{P_\mathrm{UE}}{N_0}\right) + \bar{h}_\mathrm{dB}^{(r)}\right) \tag{41}$$

and

$$\bar{h}_\mathrm{dB}^{(r)} = \frac{1}{N_\mathrm{up}^{(r)}}\sum_{t=1}^{N_\mathrm{up}^{(r)}}\left(\frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}}\left(h_{\mathrm{dB},k}^{(r,t)}\right)^p\right)^{\frac{1}{p}} \tag{42}$$

where $p_\mathrm{Max-BA} \to \infty$, $p_\mathrm{Equ-BA} = 1$, and $h_{\mathrm{dB},k}^{(r,t)} = \log_2\left(h_k^{(r,t)}\right)$ is the logarithmic channel gain in the tth frame.

According to Corollary 2, in high SNR, the objective function $\mathbb{E}_{\boldsymbol{s}}\left\{S^{(r)}\right\}$ only depends on the average logarithmic channel gain of all clients $\bar{h}_\mathrm{dB}^{(r)}$ and the number of clients $\left|\mathcal{K}^{(r)}\right|$. Then, the input vector $\boldsymbol{x}^{(r)}$ becomes

$$\boldsymbol{x}^{(r)} = \left[B_\mathrm{sum}^{(r)}, \left|\mathcal{K}^{(r)}\right|, \bar{h}_\mathrm{dB}^{(r)}\right] \tag{43}$$

Consequently, the dimension of input vector becomes a constant $N_{\boldsymbol{x}} = 3$. Moreover, since $N_{\boldsymbol{x}}$ in (43) is much smaller than that in (38), the numbers of model parameters of the value and policy networks decrease, which reduces the training complexity.

3) END-TO-END (E2E) LEARNING

In (43), $\bar{h}_\mathrm{dB}^{(r)}$ also depends on the future channel information, which is unknown for the BS at the start of each round. Fortunately, the average channel gain over multiple frames and multiple clients $\bar{h}_\mathrm{dB}^{(r)}$ is highly correlated to the history channel gain in the previous round $\bar{h}_\mathrm{dB}^{(r-1)}$. Hence, we can predict the average channel gain from $\bar{h}_\mathrm{dB}^{(r-1)}$, i.e., $\hat{\bar{h}}_\mathrm{dB}^{(r)} = g(\bar{h}_\mathrm{dB}^{(r-1)})$, where $g(\cdot)$ denotes the prediction network.

Deep learning can be used to predict channel information and optimize uplink transmission time in an E2E manner. To this end, we re-express the input vector $\boldsymbol{x}^{(r)}$ as

$$\boldsymbol{x}_\mathrm{Mix}^{(r)} = \left[B_\mathrm{sum}^{(r)}, \left|\mathcal{K}^{(r)}\right|, \bar{h}_\mathrm{dB}^{(r-1)}\right] \tag{44}$$

Since $\boldsymbol{x}_\mathrm{Mix}^{(r)}$ consists of the information in both the rth and $(r-1)$th rounds, we add subscript "Mix" to distinguish $\boldsymbol{x}_\mathrm{Mix}^{(r)}$ in (44) from $\boldsymbol{x}^{(r)}$ in (43).

To enable E2E learning, the set of training samples is $\mathcal{D}_\mathrm{MFUL} = \left\{\left(\boldsymbol{x}_\mathrm{Mix}^{(r)}, T_\mathrm{up}^{(r)}, J\right)\right\}$, where $J = \mathcal{J}(\boldsymbol{x}^{(r)}, T_\mathrm{up}^{(r)})$

is the convergence speed related to the predictive channel information $\bar{h}_{\rm dB}^{(r)}$.

4) TRAINING PROCESS

The procedure of training the policy network and value network is shown in Fig. 3. First, the parameter vector of the value network is updated in the ith iteration by

$$\boldsymbol{\theta}_{\rm v}^{i+1} = \boldsymbol{\theta}_{\rm v}^{i} - \frac{\eta_{\rm v}}{|\mathcal{B}_i|} \times \sum_{\left(\boldsymbol{x}_{\rm Mix}^{(r)}, \tilde{T}_{\rm up}^{(r)}, \tilde{J}\right)\in\mathcal{B}_i} \frac{\partial\left(\tilde{J} - f_{\rm v}\left(\boldsymbol{x}_{\rm Mix}^{(r)}, \tilde{T}_{\rm up}^{(r)}; \boldsymbol{\theta}_{\rm v}^{i}\right)\right)^2}{\partial \boldsymbol{\theta}_{\rm v}^{i}} \tag{45}$$

where $\tilde{T}_{\rm up}^{(r)} = \hat{T}_{\rm up}^{(r)} + \xi$, $\hat{T}_{\rm up}^{(r)} = f_{\rm p}\left(\boldsymbol{x}_{\rm Mix}^{(r)}; \boldsymbol{\theta}_{\rm p}^{i}\right)$ is the output of the policy network, ξ denotes a Gaussian white noise with zero mean and variance σ_ξ^2 that is used to increase exploration opportunities around $\hat{T}_{\rm up}^{(r)}$, $\tilde{J} = \mathcal{J}(\boldsymbol{x}^{(r)}, \tilde{T}_{\rm up}^{(r)})$ is the corresponding convergence speed, $\mathcal{B}_i \in \mathcal{D}_{\rm MFUL}$ is the batch in the ith iteration, and $|\mathcal{B}_i|$ is the batch size, and $\eta_{\rm v}$ is the learning rate of the value network.

Then, according to the updated value network, the parameter vector of the policy network is updated by

$$\boldsymbol{\theta}_{\rm p}^{i+1} = \boldsymbol{\theta}_{\rm p}^{i} + \frac{\eta_{\rm p}}{|\mathcal{B}_i|} \sum_{\left(\boldsymbol{x}_{\rm Mix}^{(r)}, :, :\right)\in\mathcal{B}_i} \frac{\partial f_{\rm v}\left(\boldsymbol{x}_{\rm Mix}^{(r)}, \hat{T}_{\rm up}^{(r)}; \boldsymbol{\theta}_{\rm v}^{i+1}\right)}{\partial \hat{T}_{\rm up}^{(r)}} \frac{\partial \hat{T}_{\rm up}^{(r)}}{\partial \boldsymbol{\theta}_{\rm p}^{i}} \tag{46}$$

where $\eta_{\rm p}$ is the learning rate of the policy network.

To train the DNNs of the policy network and the value network offline, we assume the BS has a simulation dataset for a similar learning task as that in FL and generates training samples $\left(\boldsymbol{x}_{\rm Mix}^{(r)}, T_{\rm up}^{(r)}, J\right)$ from the simulation dataset. First, the BS needs to simulate FL using the simulation dataset and generate the global models and the local model updates. Then, to estimate the convergence speeds, the BS generates different channel information, compresses the local updates according to the corresponding BA and UTTA, aggregates the compressed local update to update the global model, and estimate the loss reductions on the simulation dataset.

Specifically, for given available bandwidth $B_{\rm sum}^{(r)}$, the set of participating clients $\mathcal{K}^{(r)}$, and channel information in the current round $h_k^{(r,t)}$, the BS can obtain the bandwidth allocated to each client $B_k^{(r,t)}$ and the uplink transmission time $T_{\rm up}^{(r)}$ from the policy network to calculate the maximal uplink sparsity ratio $\rho_k^{(r)}$ from (15). By applying different Rand-M sparsification and probabilistic quantization methods, the BS obtains the compressed local model updates and the aggregated new global models. By averaging the loss of different new global models on the simulation dataset, the final convergence speed can be estimated by

$$J = \frac{\hat{L}(\hat{\boldsymbol{w}}^{(r)}) - \mathbb{E}_{\boldsymbol{s}}\left\{\hat{L}\left(\hat{\boldsymbol{w}}^{(r)} + \nabla\tilde{\boldsymbol{w}}^{(r+1)}\right)\right\}}{T_{\rm down}^{(r)} + T_{\rm up}^{(r)}} \tag{47}$$

where $\hat{L}(\boldsymbol{w})$ is the estimated loss function on the simulation dataset, and $\hat{\boldsymbol{w}}^{(r)}$ and $\nabla\tilde{\boldsymbol{w}}^{(r+1)} = \sum_{k\in\mathcal{K}^{(r)}} \Phi\left(\nabla\hat{\boldsymbol{w}}_k^{(r)}, \rho_k^{(r)}, b_{\rm up}\right) / \left|\mathcal{K}^{(r)}\right|$ are the global model and global model update generated by the simulation dataset, respectively.

Existing works, such as [3], [11], and [15] assume all rounds having identical durations. By contrast, $T_{\rm up}^{(r)}$ obtained from MFUL can be adaptively adjusted according to the bandwidth, participating clients, and channel qualities. We call the UTTA with the unchanged $T_{\rm up}^{(r)}$ as "Fixed-UTTA" and the UTTA obtained from MFUL as "Adap-UTTA".

5) CHANNEL INFORMATION REPORTING

Different from the static scenario where the channels remains constant during the training period, the clients have to report their channels frequently to the BS in time-varying channels. For Max-BA, the BS needs to know the channel information of all clients at the beginning of the current frame, $h_{{\rm dB},k}^{(r,t)}$. While for Adap-UTTA, the BS needs to collect the channel gains of all frames in the previous round $h_{{\rm dB},k}^{(r-1,t)}$, $t = 1, \cdots, N_{\rm up}^{(r-1)}$.

There are two ways to report the channel information, respectively called per-frame reporting and per-round reporting, depending on whether the participating clients report their channel information once a frame or once a round. For the per-frame reporting, each client reports its channel gain at the beginning of each frame. Then, at the end of each round, the BS can collect the required channel gains for Adap-UTTA. In this way, the channel information for BA is perfect, which however is not applicable for new clients who did not participate in the previous round and not report their channel information. For the per-round reporting, the clients only report the channel gains from the past $N_{\rm up}^{(r-1)}$th frame to the current frame, i.e., $\left[h_{{\rm dB},k}^{(r-1,1)}, \cdots, h_{{\rm dB},k}^{(r-1,N_{\rm up}^{(r-1)})}, h_{{\rm dB},k}^{(r,1)}\right]$ at the beginning of each round. Then, the channel information for BA becomes outdated. Notably, as shown in (31), the channel information for the Max-BA method primarily is used to determine the prioritization of clients for resource allocation. When the channel information is imprecise, it is possible for the BS to incorrectly identify the client with the highest channel gain. If the channel information is not severely outdated, the Max-BA method may allocate resources to the client with the second highest channel gain. However, this does not lead to a significant decrease in the average sparsity rate, indicating the robustness of Max-BA to outdated channel information. This indicates the feasibility of adopting the inter-round reporting scheme for resource allocation. Compared with the per-frame reporting, the per-round reporting can reduce decision delay. For both per-frame reporting and per-round reporting, each client only needs to update no more than $(N_{\rm up}^{(r-1)}+1)$ channel gains. This overhead is far less than that caused by uploading the local model update, which can be ignored.

When the channel information is severely outdated, we can use the Equ-BA method, which does not require any channel information.

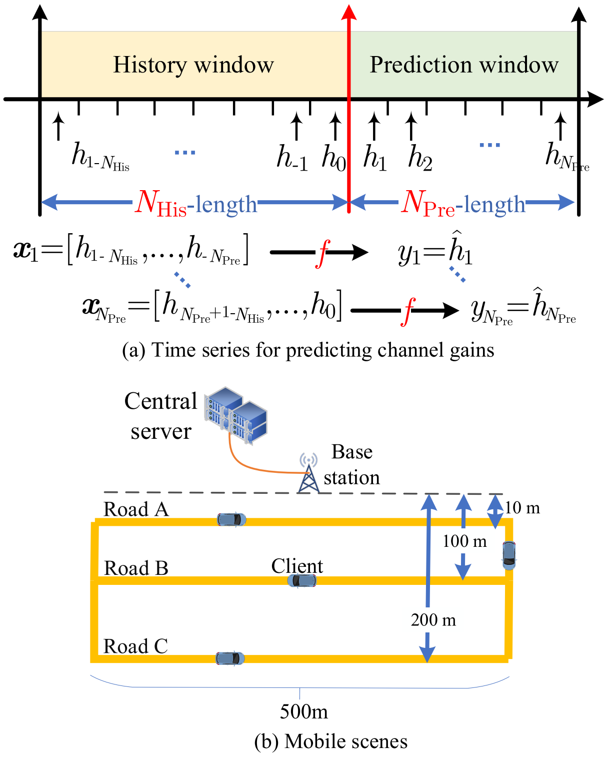
D. SUMMARY

Finally, we summarize the proposed model compression and resource allocation scheme, which includes Rand-M sparsification and probabilistic quantization for model compression and Adap-UTTA and Max-BA for resource allocation. Specially, the numbers of bits for downlink quantization and uplink quantization $b_{\rm down}$ and $b_{\rm up}$ are predetermined. The downlink transmission time $T_{\rm down}^{(r)}$ meets $C^{(r)} \geq b_{\rm down}N_{\rm model}$ to ensure that the downlink sparsity ratio $\rho^{(r)} = 1$. The uplink transmission time $T_{\rm up}^{(r)}$ is obtained by substituting $\boldsymbol{x}_{\rm Mix}^{(r)} = \left[B_{\rm sum}^{(r)}, \left|\mathcal{K}^{(r)}\right|, \bar{h}_{\rm dB}^{(r-1)}\right]$ into the trained policy network. In each frame, the BS allocates the bandwidth $B_k^{(r,t)}$ according to the channel information and calculates the cumulative sparsity ratio $\rho_k^{(r,t)}$ for client k. When some clients satisfy $\rho_k^{(r,t)} = 1$, the BS do not allocate any bandwidth to these clients.

E. POTENTIAL APPLICATIONS

The proposed method in this paper is applicable to various scenarios that require mobile client involvement in federated learning. Some typical use cases are listed as follows. 1) **Mobile client trajectory or channel prediction:** This involves predicting the trajectory or channel conditions of mobile clients based on their local location data. It can be applied in traffic management, route planning, wireless resource management, and mobility management. 2) **Intelligent traffic flow prediction:** This entails predicting traffic flow based on mobile client data such as speed, direction, and travel time. It can be used to optimize traffic flow and reduce congestion. 3) **Mobile client behavior analysis:** This involves analyzing client behavior patterns and providing personalized recommendations based on mobile client behavior data, such as app usage, clicks, and purchase preferences. 4) **Others:** There are various other applications where the proposed method can be used, such as mobile client health monitoring, mobile client speech recognition, mobile client image recognition, and object detection.

In these cases, the mobile clients exhibit mobility, resulting in variations in the number of participating clients, channel gains, and available bandwidth. The proposed model compression and resource allocation method has the following advantages: 1) adaptability to fluctuating client numbers, ensuring flexible resource allocation; 2) requirement of only the average channel gain over a period of time, avoiding the need for frequent reporting of channel quality and effectively reducing communication overhead. 3) dynamic adjustment of transmission duration based on available bandwidth, enhancing resource utilization. Therefore, the proposed method can be easily applied to these use cases.


History window
Prediction window
$h_{1\text{-}N_{\rm His}}$
$h_{\text{-}1}$
h_0
h_1
h_2
$h_{N_{\rm Pre}}$
$N_{\rm His}$-length
$N_{\rm Pre}$-length
$\boldsymbol{x}_1=[h_{1\text{-}N_{\rm His}},\dots,h_{\text{-}N_{\rm Pre}}] \xrightarrow{f} y_1=\hat{h}_1$
$\boldsymbol{x}_{N_{\rm Pre}}=[h_{N_{\rm Pre}+1\text{-}N_{\rm His}},\dots,h_0] \xrightarrow{f} y_{N_{\rm Pre}}=\hat{h}_{N_{\rm Pre}}$
Central server
Base station
Road A
Road B
Road C
Client
10 m
100 m
200 m
500m

(a) Time series for predicting channel gains

(b) Mobile scenes

FIGURE 4. **Channel gain predictor for vehicle clients.**

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the flexible model compression and resource allocation scheme. We first introduce the learning task and simulation setups, then justify the selected compression method and optimization objective, and finally compare the performance of different BA and UTTA methods.

A. LEARNING TASK AND SIMULATION SETUPS

We consider a task of predicting the average channel gains for vehicle clients, where Figs. 4(a) and 4(b) show the time series of channel gains for prediction and the trajectories of the vehicle clients.

As shown in Fig. 4(a), the average channel gains in the prediction window are predicted from those in the history window. Let $\boldsymbol{h}^{\rm Pre} = [\hat{h}_1, \cdots, \hat{h}_{N_{\rm Pre}}]^T$ and $\boldsymbol{h}^{\rm His} = [h_{1-N_{\rm His}}, \cdots, h_0]^T$ denote the vectors of the channel gains in the prediction and history windows, where $N_{\rm Pre}$ and $N_{\rm His}$ are the numbers of frames in the two windows, h_n is the average channel gain in the nth frame. We use the path loss model in 3GPP [20], the average channel gain can be expressed as $h_n = 36.8 + 36.7\log_{10}(d_n) + \log_{10}(\chi)$, where d_n is the distance between the client and the BS in the nth frame, χ is the shadowing and $\log_{10}(\chi)$ is a Gaussian random variable with zero mean and standard variance $\sigma_\chi = 8$ dB.

To reduce the complexity of the predictor, we consider the single-output predictor considered in [21], which only predicts the average channel gain in the nth frame at the $(n - N_{\text{Pre}})$th frame, i.e.,

$$\hat{h}_n = f\left(\boldsymbol{x}_n; \boldsymbol{w}\right), \quad n = 1, \cdots, N_{\text{Pre}} \tag{48}$$

where $\boldsymbol{w}$ is the prediction model vector and $\boldsymbol{x}_n = [h_{n-N_{\text{His}}}, \cdots, h_{n-N_{\text{Pre}}}]^T$ is the input vector. Then, the average channel gains in the prediction window is denoted as $\hat{\boldsymbol{h}}^{\text{Pre}} = [\hat{h}_1, \cdots, \hat{h}_N]$.

To minimize the average prediction error, the mean absolute error (MAE) is taken as the loss function, i.e.,

$$L = \mathbb{E}\left\{|\hat{h}_n - h_n|\right\} \tag{49}$$

To evaluate the performance of the predictor under varying time-varying channels, we consider a mobile network deployed in an urban area. The training and testing data are generated from simulated datasets. First, we generate the vehicle clients' trajectory dataset based on the road topology in Fig. 4(b). Subsequently, we generate the channel gains using the mentioned channel model in 3GPP. As shown in Fig. 4(b), the vehicle clients traverse three roads, each with a length of 500 m. The minimum distances between Roads A, B, C, and the BS are 10 m, 100 m, and 200 m, respectively. The vehicle clients strictly adhere to the road traffic safety regulations. The speed limits on urban roads vary among different countries, and for reference, we consult the Road Traffic Safety Law of the People's Republic of China [22]. In urban road scenarios, the typical maximal speed limit is around 60 km/h (approximately 16.7 m/s) or 80 km/h (approximately 22 m/s) or even higher. Therefore, in the training dataset, the vehicle speeds range from 18 km/h to 54 km/h (i.e. 5 m/s to 15 m/s). To evaluate the impact of outdated channel information, we consider a maximal speed of 90 km/h (i.e. 25 m/s) during the testing phase. For each client, the training set contains approximately 25,000 to 25,600 samples, while the testing set contains around 8,500 to 8,600 samples.

To reduce the MAE of prediction errors, the clients participate in FL to collaboratively train a fully connected DNN (FNN) as the predictor, where the parameters of the mobile network, predictor, and FNN are listed in Table 2.

TABLE 2. Parameters of mobile network, predictor, DNN.

	Parameter	Values
Mobile Network	Transmit power of the BS	43 dBm
	Transmit power of clients	23 dBm
	Noise power spectral density	-174 dBm/Hz
	Available bandwidth for FL	1∼3 KHz
	Num of clients	2∼6
	Duration of frame	1 s
Predictor	Length of history window N_{His}	100
	Length of prediction window N_{Pre}	41
	Input size of predictor	60
	Output size of predictor	1
DNN	Num of hidden layers	3
	Num of hidden neurons	30, 20, 10
	Num of model parameters N_{model}	2671
	Learning rate μ	0.03
	Optimization algorithm	SGD
	Activation function	Relu
	Batch size $\lvert\mathcal{B}\rvert$	128
	Num of iterations per round N_{iter}	200
	Num of downlink quantization bits b_{down}	8
	Num of uplink quantization bits b_{up}	4
	Acceptable relative error ε	3%

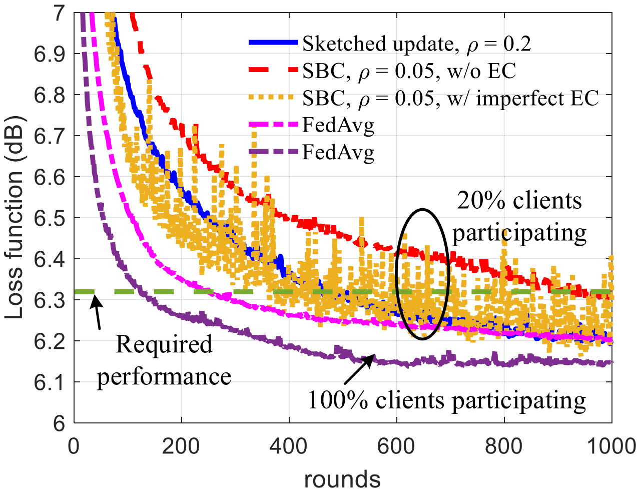


FIGURE 5. Impact of compression methods, $\left|\mathcal{K}^{(r)}\right| = 4$.

B. SIMULATION RESULTS

We compare sketched update [4] with SBC [8] in Fig. 5, which are respectively examples of Rand-M sparsification and Top-M sparsification, where their sparsity ratios are set according to the results in [4] and [8]. The performance of FedAvg without compression is also provided as a baseline. When some mobile clients cannot participate in each round, SBC either does not compensate the error or compensates the stale cumulative error, where the legend of the corresponding performance is "w/o EC" or "w/ imperfect EC". We can see that SBC suffers from severe performance degradation. By contrast, sketched update can achieve better learning performance without error compensation, which is applicable to mobile networks. In addition, as shown in Fig. 5, the minimal MAE achieved by FedAvg without model compression is $L\left(\boldsymbol{w}^{*}\right) = 6.15$ dB. When the acceptable relative error is $\varepsilon = 3\%$, the required MAE for model convergence is $L_{\varepsilon} = (1+\varepsilon)L\left(\boldsymbol{w}^{*}\right) = 6.15 \times 1.03 = 6.33$ dB.

In Table 3, we show the values of $\lambda^{(r)}$ to demonstrate the tightness of the upper bound on model error caused by random sparsification, as stated in Proposition 1. It can be observed that in each round, the values of $\lambda^{(r)}$ vary from 0.89 to 0.95. Therefore, the upper bound is tight.

In Fig. 6 and Fig. 7, we investigate the impact of the uplink transmission time T_{up} on the total communication time

TABLE 3. The ratio of the minimal and maximal norms of model updates.

Round number r	2	5	10	50	100
$\lambda^{(r)} = \left\Vert \nabla \boldsymbol{w}_{\min}^{(r)} \right\Vert / \left\Vert \nabla \boldsymbol{w}_{\max}^{(r)} \right\Vert$	0.95	0.90	0.89	0.90	0.91

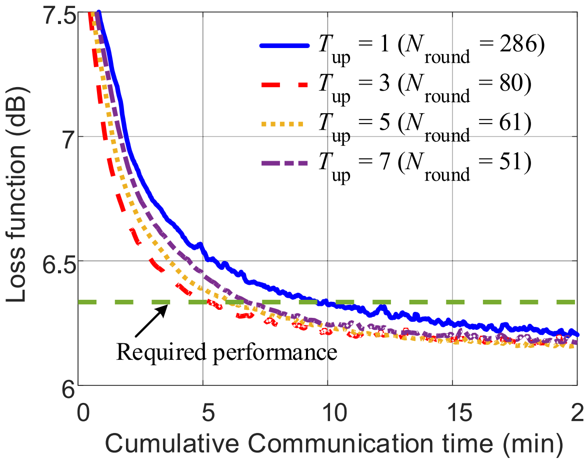
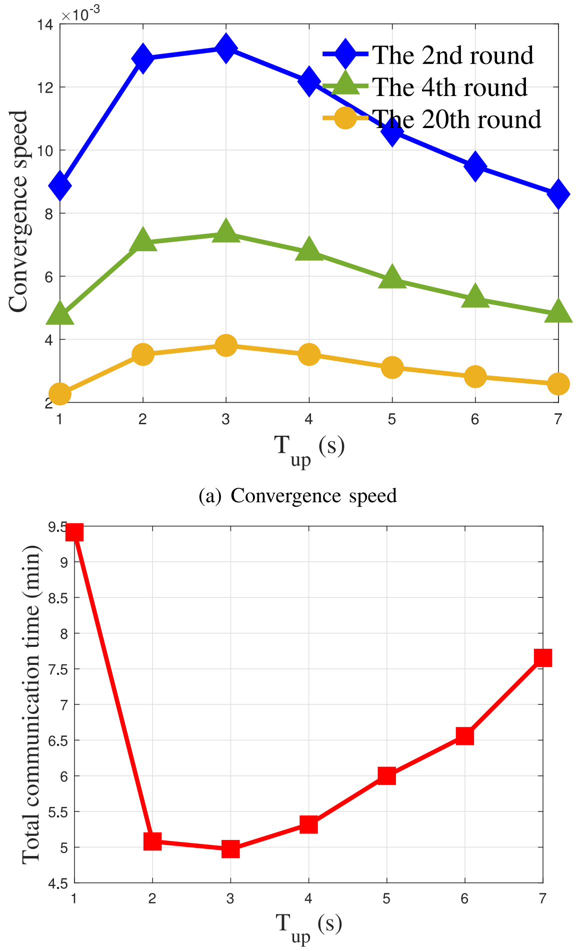


FIGURE 6. Learning curve, $\left|\mathcal{K}^{(r)}\right| = 4$.

and convergence speed, where Equ-BA is considered and the clients only move on Road A. The learning curves of different T_{up} are compared in Fig. 6. When T_{up} increases as 1, 3, 5, and 7 seconds, the number of communication rounds decreases monotonously as 286, 80, 61, and 51. Considering that the downlink transmission time is $T_{\text{down}} = 1$ second, the total communication time is $T_{\text{tot}} = N_{\text{round}} \left(T_{\text{down}} + T_{\text{up}}\right) = 572$, 320, 366, and 408 seconds. It decreases first, then increases, and reaches the minimum when $T_{\text{up}} = 3$ seconds. This indicates that there is an optimal value of T_{up} to balance the number of communication rounds and per-round delay. This explains why optimizing UTTA can reduce the total communication time.

In Fig. 7, we show the convergence speed and the total communication time. It can be seen that the convergence speed decreases with the round number r during the training process. No matter in which round, as T_{up} increases, the convergence speed increases first, then decreases, and reaches its maximum when $T_{\text{up}} = 3$ seconds. It suggests that minimizing the total communication time is equivalent to maximizing the convergence speed of each round. As a consequence, transforming the optimization problem from $\mathbf{P}_1$ to $\mathbf{P}_2$ does not incur performance loss.

To illustrate the performance gain provided by inter-round UTTA or intra-round BA, we first compare the performance of different BA methods with given uplink transmission time, and then compare the performance of different UTTA methods with the optimal BA (i.e., max-BA). The hyper-parameters of MFUL for Adap-UTTA are listed in Table 4, where the FNNs are used for both the policy network and value network. In Fig. 8, we compare the total communication time in different scenarios, when the available bandwidth $B_{\text{sum}}^{(r)}$ and the number of participating clients $\left|\mathcal{K}^{(r)}\right|$ are unchanged. In Fig. 9, the performance is shown when either $B_{\text{sum}}^{(r)}$ or $\left|\mathcal{K}^{(r)}\right|$ changes in different rounds, where clients only move on Road A.



(a) Convergence speed

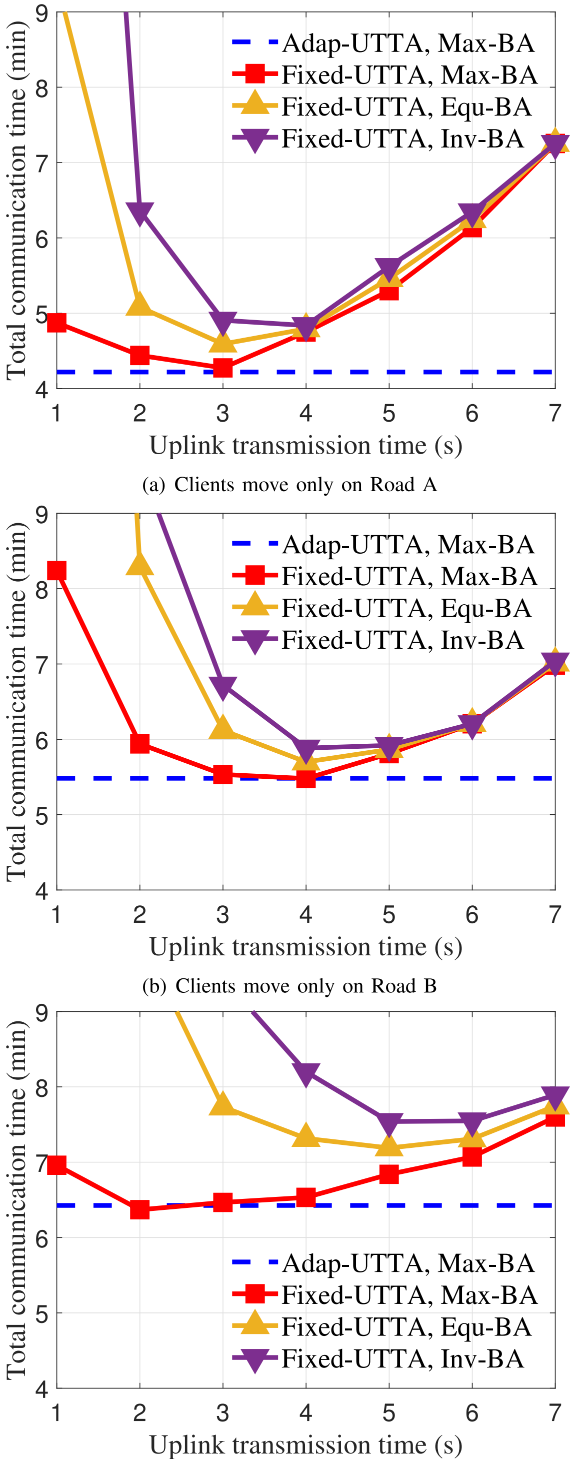
(b) Total communication time

FIGURE 7. Impacts of T_{up} with Equ-BA, $B_{\text{sum}}^{(r)} = 2$ KHz, $\left|\mathcal{K}^{(r)}\right| = 4$.

TABLE 4. Hyper-parameters of model-free unsupervised learning.

Hyper-parameters	Policy network	Value network
Num of input neurons	3	4
Num of output neurons	1	1
Num of hidden layers	3	2
Num of hidden neurons	[50, 40, 30]	[200,150]
Optimization algorithm	Adam	Adam
Learning rate	0.0003	0.005
Noise variance for exploration	0∼4	–
Num of training samples	10240	

To demonstrate the advantages of the proposed method, we need to compare it with existing joint optimization approaches [3], [11], [15], [16]. However, these approaches

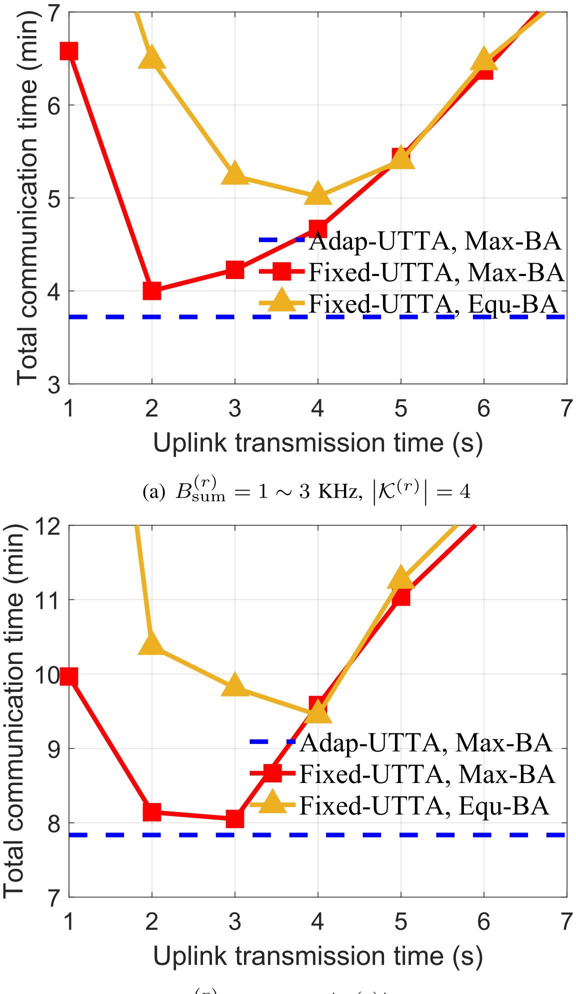


(a) Clients move only on Road A

(b) Clients move only on Road B

(c) Clients move on the three roads

FIGURE 8. **Total communication time versus T_{up}, $B_{\text{sum}}^{(r)} = 2$ KHz, $\left|\mathcal{K}^{(r)}\right| = 4$.**



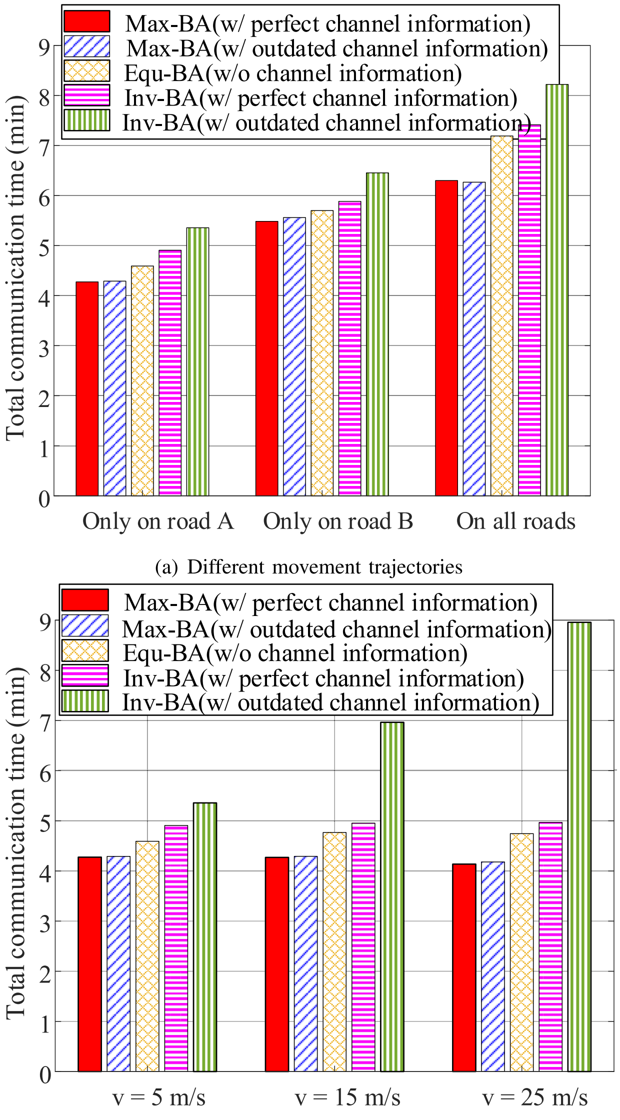
(a) $B_{\text{sum}}^{(r)} = 1 \sim 3$ KHz, $\left|\mathcal{K}^{(r)}\right| = 4$

(b) $B_{\text{sum}}^{(r)} = 2$ KHz, $\left|\mathcal{K}^{(r)}\right| = 2 \sim 6$

FIGURE 9. **Total communication time versus T_{up}.**

address joint user scheduling optimization [11], [15], [16], consider the impact of computation time on total round duration [3], [15], [16], and use the theoretical analysis results assuming static channel conditions during training [3], [16]. Since our focus is on optimizing the intra-round BA and inter-round UTTA methods, we compare them with the BA and UTTA methods in these studies. Specifically, when ignoring the impact of computation time, the BA algorithms in [3], [15], and [16] reduce to the Inv-BA method, which can be compared with our intra-round BA method. Additionally, the works in [3], [11], and [15] consider the assumption of identical round durations, which is referred to as the Fixed-UTTA method and can be compared with our inter-round UTTA method. The optimal performance of the Fixed-UTTA method is achieved by exhaustively finding all possible UTTA strategies. Therefore, if our Adap-UTTA method outperforms the Fixed-UTTA method with exhaustive search, we can conclude that our inter-round UTTA method surpasses the existing UTTA methods.

In Section V, we discuss two ways to collecting channel information, i.e., the per-frame and the per-round channel information reporting. Since the former can provide perfect channel information for BA, while the latter can only provide outdated channel information, we call the former as ‘‘perfect channel information’’ and the latter as ‘‘outdated channel information’’. We first provide the total communication time



(a) Different movement trajectories

(b) Different movement speeds

FIGURE 10. Total communication time with Adap-UTTA, $B_{\text{sum}}^{(r)} = 2$ KHz, $\left|\mathcal{K}^{(r)}\right| = 4$.

with "perfect channel information" in Figs. 8 and 9. Then, to show the robustness of BA methods to outdated channel information, the performance with "outdated channel information" is provided in Fig. 10.

As shown in Fig. 8, for an arbitrary given T_{up}, Max-BA always achieves the best performance, Equ-BA is in the middle, and Inv-BA is the worst. Since Max-BA requires the minimal total communication times, it means that Max-BA can always reach the maximal convergence speed for given T_{up}. Therefore, Max-BA achieve highest resource usage efficiency. When clients move on all three roads, the difference between the channel gains and the data rates of clients is the largest among all scenarios, hence the gap between the total communication time of three BA also gets the largest. This agrees with the analysis in Corollary 1.

When Max-BA is used, the optimal uplink transmission time T_{up}^* in three scenarios is 3, 4, and 2 seconds, respectively, which varies in different scenarios. To obtain T_{up}^*, Fixed-UTTA needs to collect the channel information of all clients in all rounds at the beginning of the FL and finds T_{up}^* by exhaustive searching. Adap-UTTA does not require the future channel information any more. It can always obtain the optimal value from the well-trained DNN with low complexity.

As shown in Fig. 9, T_{up}^* varies in different rounds as $B_{\text{sum}}^{(r)}$ or $\left|\mathcal{K}^{(r)}\right|$ changes. As a result, Fixed-UTTA with exhaustive searching cannot achieve the minimal total communication time. By contrast, Adap-UTTA can always adjust T_{up} adaptively according to the available bandwidth and the number of participating users, which can shorten the total communication time.

As Max-BA and Adap-UTTA can reach the required performance with the minimal total communication, the proposed scheme can improve resource usage efficiency to accelerate the convergence of FL.

We compare the total communication time of different BA methods with different channel information in Fig. 10, where Adap-UTTA is considered. Specially, the total communication time when clients move on different trajectories at 5 m/s is shown in Fig. 10(a), while the performance when clients move on Road A at different speeds is shown in Fig. 10(b). Since Equ-BA does not require any channel information, the performance of Equ-BA without channel information is provided as a baseline.

In all scenarios, the performance of Max-BA with outdated channel information is close to that with perfect channel information. By contrast, the performance of Inv-BA with outdated channel information suffers a considerable performance loss from Inv-BA with perfect channel information, especially when the movement speed of clients increases. Therefore, Max-BA is more robust to outdated channel information than Inv-BA. This is because for Max-BA, the channel information mainly determines which client to be allocated resources first, whereas for Inv-BA, the channel information determines how many resources are allocated to each user.

Moreover, with outdated channel information, the total communication time of Max-BA is always lower than that of Equ-BA and Inv-BA. This means that Max-BA outperforms existing BA methods even when using outdated channel information. Specifically, when the clients move on all three roads as shown in Fig. 10(a), the total communication time with Max-BA, Equ-BA, and Inv-BA is 6.2, 7.2, and 8.2 minutes, respectively. When clients move at 25 m/s, the total communication time of Max-BA, Equ-BA and Inv-BA is 4.1, 4.7, and 9.0 minutes.

V. CONCLUSION

In this paper, we proposed a flexible model compression and resource allocation scheme for federated learning in a bandwidth-limited mobile network, where the considered Rand-M sparsification does not require error compensation and is more suitable to the mobile clients than Top-M sparsification. The formulated single-round optimization problem to maximize the convergence speed achieves the same performance as the multi-round optimization problem, which not only avoids the difficulty of deriving the number of communication rounds and obtaining future channel information, but also reduces the complexity of solving joint optimization problem. By taking advantage of end-to-end model-free unsupervised learning, the designed intra-round bandwidth allocation and the inter-round uplink transmission time allocation can adaptively adjust the compression level and transmission resources to different clients in different rounds. The proposed scheme outperforms the existing methods in dynamic environments and is robust to the outdated channel information.

APPENDIX

A. PROOF OF PROPOSITION 1

From (9), we have $\boldsymbol{E}_{\mathrm{S}}^{(r)} = \sum_{k\in\mathcal{K}^{(r)}} \boldsymbol{E}_{\mathrm{S},k}^{(r)} / \left|\mathcal{K}^{(r)}\right|$, where $\boldsymbol{E}_{\mathrm{S},k}^{(r)} = \nabla\tilde{\boldsymbol{w}}_k^{(r)} - \nabla\bar{\boldsymbol{w}}_k^{(r)}$ is the update error of client k. Since the update errors are also independent among clients in the random sparsification, we have

$$\mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S}}^{(r)}\right\|_1\right\} = \frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}} \mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S},k}^{(r)}\right\|_1\right\} \tag{A.1}$$

Since $\nabla\tilde{\boldsymbol{w}}_k^{(r)} = \nabla\bar{\boldsymbol{w}}_k^{(r)} \odot \boldsymbol{s}_k$ from (7), we have $\boldsymbol{E}_{\mathrm{S},k}^{(r)} = \nabla\tilde{\boldsymbol{w}}_k^{(r)} - \nabla\bar{\boldsymbol{w}}_k^{(r)} = \nabla\bar{\boldsymbol{w}}_k^{(r)} \odot (\boldsymbol{1} - \boldsymbol{s}_k)$, where $\boldsymbol{1}$ denotes the vector consisting of all ones. Considering that $\boldsymbol{E}_{\mathrm{S},k}^{(r)}$ contains $N_{\mathrm{model}}(1-\rho_k)$ non-zero elements and the non-zero elements are statistical independent. Then, we have $\mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S},k}^{(r)}\right\|_1\right\} = (1-\rho_k)\left\|\nabla\boldsymbol{w}_k^{(r)}\right\|_1$. Upon substituting into (A.1), we have

$$\mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S}}^{(r)}\right\|_1\right\} = \frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}} (1-\rho_k)\left\|\nabla\boldsymbol{w}_k^{(r)}\right\|_1 \tag{A.2}$$

Considering $\left\|\nabla\boldsymbol{w}_k^{(r)}\right\|_1 \leq \left\|\nabla\boldsymbol{w}_{\max}^{(r)}\right\|$, $\forall k \in \mathcal{K}^{(r)}$, and substituting it into (A.2), we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S}}^{(r)}\right\|_1\right\} &\leq \left\|\nabla\boldsymbol{w}_{\max}^{(r)}\right\|\left(1 - \sum_{k\in\mathcal{K}^{(r)}}\frac{\rho_k^{(r)}}{\left|\mathcal{K}^{(r)}\right|}\right)\\ &\leq \left\|\nabla\boldsymbol{w}_{\max}^{(r)}\right\|\left(1-\bar{\rho}^{(r)}\right) \end{aligned}\tag{A.3}$$

According to the Lipschitz continuous with the local Lipschitz constant $\alpha_1^{(r)}$, we have $\mathbb{E}_{\boldsymbol{s}}\left\{\left|\Delta L_{\mathrm{S}}^{(r)}\right|\right\} \leq \alpha_1^{(r)}\mathbb{E}_{\boldsymbol{s}}\left\{\left\|\boldsymbol{E}_{\mathrm{S}}^{(r)}\right\|_1\right\}$. By substituting (A.3) into it, we obtain (26).

B. PROOF OF PROPOSITION 2

By substituting (31), (32), and (33) into (29) and $\bar{\rho}^{(r)} = \sum_{t=1}^{N_{\mathrm{up}}^{(r)}} \Delta\rho^{(r,t)} / \left|\mathcal{K}^{(r)}\right|$, we can obtain the average sparsity ratios of the BA methods as

$$\bar{\rho}_{\mathrm{Max-BA}}^{(r)} = \nu^{(r)}\sum_{t=1}^{N_{\mathrm{up}}^{(r)}} \max_{k\in\mathcal{K}^{(r)}}\left\{R_k^{(r,t)}\right\} \tag{B.1a}$$

$$\bar{\rho}_{\mathrm{Equ-BA}}^{(r)} = \nu^{(r)}\sum_{t=1}^{N_{\mathrm{up}}^{(r)}} \frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}} R_k^{(r,t)} \tag{B.1b}$$

$$\bar{\rho}_{\mathrm{Inv-BA}}^{(r)} = \nu^{(r)}\sum_{t=1}^{N_{\mathrm{up}}^{(r)}} \left(\frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}} \left(R_k^{(r,t)}\right)^{-1}\right)^{-1} \tag{B.1c}$$

The average sparsity ratios are proportional to the maximum, arithmetic mean, and harmonic mean of $R_k^{(r,t)}$, $k \in \mathcal{K}^{(r)}$, which are special cases of the generalized mean [23] as follows,

$$\begin{aligned}&\left(\frac{1}{\left|\mathcal{K}^{(r)}\right|}\sum_{k\in\mathcal{K}^{(r)}}\left(R_k^{(r,t)}\right)^p\right)^{\frac{1}{p}}\\ &= \begin{cases} \max\limits_{k\in\mathcal{K}^{(r)}}\left\{R_k^{(r,t)}\right\}, & \forall p \to \infty\\ \dfrac{1}{\left|\mathcal{K}^{(r)}\right|}\sum\limits_{k\in\mathcal{K}^{(r)}} R_k^{(r,t)}, & \forall p = 1\\ \left(\dfrac{1}{\left|\mathcal{K}^{(r)}\right|}\sum\limits_{k\in\mathcal{K}^{(r)}}\left(R_k^{(r,t)}\right)^{-1}\right)^{-1}, & \forall p = -1\end{cases}\end{aligned}\tag{B.2}$$

By substituting (B.2) into (B.1), we obtain (34).

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[2] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[3] P. Liu et al., "Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation," *Frontiers Inf. Technol. Electron. Eng.*, vol. 23, no. 8, pp. 1247–1263, Aug. 2022.

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

[5] B. Xu, W. Xia, J. Zhang, T. Q. S. Quek, and H. Zhu, "Online client scheduling for fast federated learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1434–1438, Jul. 2021.

[6] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[7] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2021.

[8] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[11] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, “Convergence of update aware device scheduling for federated learning at the wireless edge,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, Jun. 2021.

[12] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, “To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10.

[13] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, “Toward an intelligent edge: Wireless communication meets machine learning,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[14] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, “Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations,” *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 217–226, May 2020.

[15] 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.

[16] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, “Joint resource management and model compression for wireless federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[17] N. Kourtellis, K. Katevas, and D. Perino, “FLaaS: Federated learning as a service,” 2020, *arXiv:2011.09359*.

[18] B. De Baets, H. De Meyer, and R. Mesiar, “Lipschitz continuity of copulas w.r.t. $L-p$-norms,” *Nonlinear Anal.-Theory Method Appl.*, vol. 72, nos. 9–10, pp. 3722–3731, 2010.

[19] D. Liu, C. Sun, C. Yang, and L. Hanzo, “Optimizing wireless systems using unsupervised and reinforced-unsupervised deep learning,” *IEEE Netw.*, vol. 34, no. 4, pp. 270–277, Jul. 2020.

[20] *Further Advancements for E-UTRA Physical Layer Aspects*, document TR 36.814, V0.4.1, 3GPP, 2009.

[21] W. Zhang, Y. Liu, T. Liu, and C. Yang, “Trajectory prediction with recurrent neural networks for predictive resource allocation,” in *Proc. 14th IEEE Int. Conf. Signal Process. (ICSP)*, Aug. 2018, pp. 634–639.

[22] (2005). *Regulations on the Implementation of the Road Traffic Safety Law of the People’s Republic of China*. [Online]. Available: https://ailvxiong.cn/wp-content/uploads/2022/01/Regulations-on-the-Implementation-of-the-Road-Traffic-Safety-Law-of-the-Peoples-Republic-of-China.pdf

[23] P. S. Bullen, *Handbook of Means and Their Inequalities*. Norwell, MA, USA: Kluwer, 2003.

YIWEN HU (Student Member, IEEE) received the B.E. degree from the Shenyuan Honors College, Beihang University, China, in 2020, and the M.A. degree from the School of Electronics and Information Engineering, Beihang University, in 2023. His recent research interests include federated learning for wireless communications.

TINGTING LIU (Member, IEEE) received the Ph.D. degree in signal and information processing from Beihang University, China, in 2011. Since 2014, she has been an Associate Professor with the School of Electronics and Information Engineering, Beihang University. She has published more than 40 journal and conference papers in the field of signal processing for wireless communications. Her recent research interests include distributed machine learning for wireless resource allocation, interference management, and mobility management. She is recognized as one of the major contributors to the development of low-rate wireless personal area networks (LR-WPANs) standards IEEE 802.15.4-2006 and GB/T 15629.15-2007. She serves as a reviewer for several IEEE journals.

CHENYANG YANG (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Beihang University, China, in 1997. Since 1999, she has been a Full Professor with Beihang University. She has published over 300 papers in the fields of machine learning for wireless communications, energy efficient resource allocation, URLLC, wireless caching, and interference management. Her recent research interests include mobile/wireless AI and URLLC. She was supported by the first Teaching and Research Award Program for Outstanding Young Teachers of Higher Education Institutions from the Ministry of Education of China. She has served as an associate or the guest editor for several IEEE journals.

YUANFANG HUANG received the B.S. degree in automation from Tsinghua University, China, in 1997, and the M.A. degree in communication and information systems from the China Academy of Telecommunication Technology, China, in 2003. She is currently a Senior Engineer with CICT Mobile Communication Technology Company Ltd., China. She has long-term engaged in RAN intelligence and openness standardization in O-RAN alliance. Her main research interests include RAN architecture, RAN intelligence, and RAN openness in 5G/6G wireless mobile communication systems.

SHIQIANG SUO received the B.S degree in automation from Tsinghua University, China, in 1999, and the M.A degree in communication and information systems from the China Academy of Telecommunication Technology, China, in 2002. He has long-term engaged in wireless mobile communication systems (including 3G, 4G, 5G, and 6G) new technology research, verification and standardization (including 3GPP, ITU-R, and O-RAN alliance), and has obtained more than 400 authorized invention patents. He is currently the Vice General Manager of the Innovation Center, CICT Mobile Communication Technology Company Ltd., and responsible for the research of 6G and future new technologies, mainly focusing on massive MIMO, AI, integrated sensing and communication, and wireless network architecture.