Received 11 May 2024; revised 21 June 2024; accepted 30 June 2024. Date of publication 9 July 2024; date of current version 26 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3425531

Adaptive Sparsification and Quantization for Enhanced Energy Efficiency in Federated Learning

OUIAME MARNISSI, HAJAR EL HAMMOUTI (Member, IEEE), AND EL HOUCINE BERGOU (Associate Member, IEEE)

College of Computing, Mohammed VI Polytechnic University, Ben Guerir 43150, Morocco

CORRESPONDING AUTHOR: O. MARNISSI (e-mail: ouiame.marnissi@um6p.ma)

ABSTRACT Federated learning is a distributed learning framework that operates effectively over wireless networks. It enables devices to collaboratively train a model over wireless links by sharing model parameters rather than personal data. However, a key challenge in federated learning arises from the limited computational and communication resources of the devices. Therefore, optimizing energy consumption is crucial for practical implementations of federated learning. In this context, we address energy minimization by applying compression techniques that reduce the number of bits required for both local computation and uplink communications. We develop an optimization framework that aims to minimize the total energy consumption across all devices involved in the training process. This framework considers quantization levels for local computation and uplink transmission, as well as the level of sparsification for parameter transmission. The optimization is constrained by requirements on latency and the target accuracy. To solve this complex problem, we first derive the required number of global training rounds, to achieve the desired accuracy. We then employ an iterative algorithm to efficiently find the optimal parameters of the studied problem. Our numerical results show that the proposed approach achieves significant performance and considerably reduces the energy consumption compared to two different federated learning baseline schemes.

INDEX TERMS Energy efficiency, federated learning, quantization, sparsification.

I. INTRODUCTION

IN THE past few years, there has been exponential growth in machine learning (ML) applications for mobile phones [1]. These applications primarily rely on ML algorithms to predict user preferences, recommend personalized content, and propose appropriate services and products. Importantly, these applications harness locally generated data to train sophisticated ML models, including deep neural networks (DNNs). In fact, the performance of DNNs in solving a wide range of ML problems, such as image classification, speech recognition, and text translation, has exceeded expectations.

In classical ML schemes, the collected data is shared with a central server for processing and training. However, this centralized framework may compromise the user's privacy. To overcome this issue, a novel ML paradigm has emerged, namely federated learning (FL) [2]. In FL, devices locally train a shared ML model by iteratively sending ML updates, rather than their private data, to a central server. As a consequence, the user's privacy is preserved. Many FL works implement DNNs on mobile devices. However, this comes with several challenges. Initially, DNNs were trained over fast and power-hungry Graphic Processing Units (GPUs). Due to the limited battery of mobile devices and their restricted memory, training DNNs over mobile devices is not a trivial task. It requires efficient computation and memory access techniques to reduce the computational load and save the energy consumption. Moreover, the surge of FL as a distributed technique to collaboratively train ML models has led to an additional constraint: the communication bottleneck. Devices are required to transmit their locally trained models to a distant server. This process is limited by the network bandwidth and the substantial size of ML models, making the communication of dense gradient vectors within a restricted time frame a significant challenge [3], [4].

A. QNNS FOR EFFICIENT COMPUTATION

In practice, to train a DNN model, devices run several iterations of stochastic gradient descent (SGD). At each local iteration, the processor stores a high-dimensional vector of floating-point numbers that captures the weight values. It also maintains the real values of the model's activation functions. As a result, the storage, access, and multiplication-accumulation-computations (MAC) come with a heavy energy consumption. In most current implementations of DNNs, the network parameters are represented with 32-bit floating values. This high bit-precision along with the large dimension of a DNN's parameters results in Gigabytes of storage and millions of Floating-Point-Operations-Per-Second (FLOPS).

To reduce the computation and storage costs, quantized neural networks (QNNs) have been proposed [5], [6], [7], [8]. The key idea of QNNs is to use quantization techniques during the training and inference to save memory, time, and energy. In [9], the authors show that a significant gain in memory can be achieved when using scalar and structured quantization, while maintaining good accuracy. In [6], the authors are interested in the quantization scale. They show that high values of weights and activations do not occur very frequently. Consequently, they suggest adjusting the quantization magnitude to reduce the quantization noise. In [10], a variable-length quantization method is proposed. Two bit-precision levels are used to quantize the weights. The authors employ a measure to classify weights into two categories: low-bit precision and high-bit precision. They show that, using the proposed quantization scheme, a fast convergence is achieved with an accuracy comparable to full bit-precision models. An extreme case of quantization is binary neural networks (BNNs) where both weights and activations are represented with two sign bits (i.e., $+$ and $-$) [11], [12], [13]. A major advantage of such an approach is the conversion of MAC operations to bit-wise operations. In [12], it is shown that the accuracy drops by less than 1% while a significant reduction is achieved in terms of memory size and access.

B. COMMUNICATION-EFFICIENT TECHNIQUES FOR FL

In the context of FL, mobile devices are required to transmit their weights over a resource-constrained network. Although QNN reduces the number of transmitted bits, it does not take into account the constraints of the wireless channel. In general, the total number of exchanged bits during an FL process depends on three main parameters: the size of the training model, the number of communication rounds, and the number of devices participating in the training. As a consequence, reducing the communication overhead during an FL process can be performed in three ways. First, to reduce the communication overhead, it is suggested to decrease the number of participants per communication round. Partial client participation uses limited communication bandwidth, and when optimally designed, it can also accelerate the FL convergence and minimize the communication costs [14], [15], [16], [17].

The second technique for enhancing communication efficiency involves decreasing the frequency of model updates. Rather than transmitting parameters after each gradient descent iteration, research has demonstrated that performing a small number of iterations before sending an update can maintain the overall performance of FL with minimal impact [18]. Additionally, the number of local iterations can be dynamically adjusted to optimize both performance and communication overhead. For example, when the average model experiences a rapid change in the target loss function, local learners are asked to increase their frequency of updates in order to quickly adapt to the new optimization target [19].

Finally, to reduce the gradient vector's size, gradients are compressed [20], [21], [22] using sparsification and quantization techniques. The objective of sparsification is to send a sparse vector of the gradients estimates that includes only selected values of the gradients. One way to sparsify the gradient vector is by keeping the most impactful values of the gradients, and dropping the others. Although this approach, commonly known as top-k sparsification, requires additional computational resources to sort the gradient vector, it comes with a convergence guarantee when k is selected appropriately and the error from using sparsification is included in the gradients' updates [23]. To further reduce the communication expenses, sparsification is combined with quantization. Quantization consists of reducing the number of digits that are used to encode the gradient vector. Various quantization techniques have been proposed. One of these approaches is probabilistic quantization. According to this technique, a gradient estimate is randomly mapped to one of the closest predefined quantum values [24]. Another approach, dense quantization, enforces devices to reduce their gradients' values to a single bit representing the sign of the gradient [25].

One notable work that integrates quantization with sparsification is presented in [26]. In this work, the authors introduce an efficient FL framework with ultralow-bitwidth quantization. The framework applies a combination of quantization and sparsification strategies to compress both upstream and downstream data, thereby enhancing communication efficiency and reducing bandwidth requirements. In the same line, authors in [27] propose a quantization scheme for both uplink and downlink communication in FL in order to maximize the learning accuracy and convergence rate. Although these studies [26], [27] effectively reduce energy consumption through quantization and sparsification, they do not optimally tune the compression levels to specifically minimize energy usage. In this regard, the work in [28] considers the optimization of the energy consumption of an FL system. The authors formulate a multi-objective problem that minimizes both the energy consumption and the global number of communication rounds. In the problem they investigate, the authors focus on optimizing the number of encoding bits to minimize computation and communication energy, yet they overlook the benefits of sparsification. Additionally, they do not account for network latency [29], which, if not addressed properly, can significantly impede the FL scalability [30]. To address these oversights and better manage network latency, we propose an approach that involves transmitting only a subset of the model updates at a fixed precision level. The proposed approach aims to finely select the quantization and sparsification levels in a heterogeneous system such that the energy is minimized and both accuracy and latency constraints are respected.

C. CONTRIBUTION

In this paper, we present an approach titled Sparsified and Quantized Federated Learning (SQFL). The core concept is to select optimal compression levels to minimize the overall energy consumption while adhering to specified time budgets and achieving desired accuracy levels. Specifically, in scenarios with extreme network latencies, our method facilitates the exchange of only a minimal number of critical elements. To address this challenge, we analytically determine the target accuracy as a function of the number of communication rounds. We then use an iterative algorithm to identify the optimal compression settings. The efficacy of our approach is demonstrated through numerical results. Our main contributions are summarized as follows.

- We develop an optimization framework that minimizes the total energy consumed for local computation and wireless transmission by all devices during the training process. We aim to optimize the quantization levels of computation and communication jointly with the sparsification level used for transmission. In our formulation, we account for the transmission time along with the target accuracy.
- To address this optimization problem, we first establish a relationship between the convergence rate and the number of global training rounds, T. We demonstrate that the difference between the expected global loss function after T rounds and its minimum achievable value is upper-bounded by a function that depends on quantization and sparsification levels.
- Based on our upper-bound analysis, we formulate an approximate optimization problem. This upper-bound analysis provides insights into the impact of the bit-precision and sparsification ratio on the number of communication rounds required for convergence. It also establishes a trade-off between the duration of the communication round and the target accuracy.
- To tackle the approximate problem, we propose an iterative method that sequentially addresses optimization sub-problems and converges to effective quantization and sparsification levels. We also show that this approach yields low time complexity.
- Our numerical results show that the proposed method significantly reduces energy consumption when compared to two other FL baseline approaches. Additionally, it achieves results comparable to those of exhaustive research while substantially decreasing the execution time.

The remainder of this paper is organized as follows. In Section II, we describe the system model. Then, in Section III, we summarize the main steps of the proposed sparsified and quantized FL framework. Section IV formulates the problem mathematically and describes our convergence analysis. In Section V, we propose an iterative algorithm to solve the approximate problem and study its complexity. Simulation experiments are provided and analyzed in Section VI. Finally, Section VII draws the conclusions of our paper.

Notations: All vectors are denoted by bold font letters (e.g., $\boldsymbol{x}$), while scalars are denoted by normal font letters (e.g., y). We use $||.||$ to denote the L_2 norm. We represent the cardinality of a set S by $|S|$. The gradient of $F(\boldsymbol{x})$ is represented by $\nabla F(\boldsymbol{x})$. We denote the expectation with respect to a random variable X by $\mathbb{E}_X[.]$.

II. SYSTEM MODEL

In this section, we begin by describing the FL setup. We then outline the adopted quantization and sparsification schemes. Following this, we detail the computation and transmission models that were considered.

A. LEARNING MODEL

Consider an FL setup where a set $\mathcal{N}$ of N devices train collaboratively an ML model with the coordination of a central server. We suppose that each device k has a local dataset D_k with size $|D_k|$. To fit the parameters of the trained model, a loss function $F(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})$ is minimized,

$$F(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{N} \frac{|D_k|}{\sum_{j \in \mathcal{N}} |D_j|} f_k\big(\boldsymbol{w}, \boldsymbol{x}_k, \boldsymbol{y}_k\big) \tag{1}$$

where f_k is the loss function of device k, $\boldsymbol{w} \in \mathbb{R}^d$ is the ML parameter vector of size d, $\{\boldsymbol{x}_k, \boldsymbol{y}_k\}$ corresponds to the pairs input-output for the considered ML model.

Under the federated averaging algorithm, each device conducts local training on its dataset and transmits the resulting gradient vector to the central server, where it is averaged.

B. QUANTIZATION

To minimize computation and communication overhead, we assume that devices implement quantization. Consider w as an arbitrary element of $\boldsymbol{w}$ such that $|w| \leq M$. Let m be the number of bits used to encode w. When applying m-bit quantization, we divide the interval $[-M, M]$ into $2^m - 1$ segments. Each segment i is denoted by $[z_i, z_{i+1}]$, with z_i and z_{i+1} are the bounds of the segment. The stochastic rounding quantization for m bits is then defined as follows

$$Q(w) = \begin{cases} z_i & \text{w.p. } \frac{z_{i+1}-w}{z_{i+1}-z_i} \\ z_{i+1} & \text{w.p. } \frac{w-z_i}{z_{i+1}-z_i}. \end{cases} \tag{2}$$

It is important to note that the adopted quantization scheme is unbiased, i.e., $\mathbb{E}_Q[Q(w)] = w$ with $\mathbb{E}_Q$ the expectation over the quantization operator.

We implement quantization at two distinct levels to enhance efficiency. Initially, during local training of the machine learning model, a process we term *computation quantization*, is applied. In this phase, low-precision weights are utilized, which substantially reduces the memory requirements, access times, and arithmetic operations involved in training. This reduction significantly lowers the computation energy consumption. To further decrease energy usage and accommodate the resource constraints typical of wireless communication networks, a second level of quantization is applied prior to the transmission of the parameter vector. We call the second level of quantization *communication quantization*. We use n_k to denote the bit-precision of the computation quantization for device k and m_k to designate the communication quantization of the transmitted vector of user k.

C. SPARSIFICATION

To further minimize the data transmitted from devices to the server, we introduce a sparsification step for the last quantized local model vector prior to its transmission. Specifically, for each device k, a sparsification operator S_k parameterized by a parameter γ_k is applied to the quantized parameter vector. More precisely, S_k and γ_k satisfy the following property. For any $\boldsymbol{w} \in \mathbb{R}^d$, we consider a sparsification operator $S_k(.) : \mathbb{R}^d \to \mathbb{R}^d$ such that, $\exists \gamma \leq 1\ \forall \boldsymbol{w} \in \mathbb{R}^d$, we have

$$\mathbb{E}_{S_k} || S_k(\boldsymbol{w}) - \boldsymbol{w}||_2^2 \leq (1-\gamma_k)||\boldsymbol{w}||_2^2 \tag{3}$$

with $\mathbb{E}_{S_k}$ is the expectation over the sparsifier S_k.

This class of sparsifiers includes commonly used sparsifiers such as Top_θ (where only the θ highest absolute values of the vector elements are transmitted) and $Rand_\theta$ (where θ elements are picked uniformly at random for the transmission), with $\gamma = \frac{\theta}{d}$.

In the following subsection, we present the models adopted for FL computations and vectors' transmissions, along with their respective energy consumption models.

D. COMPUTATION AND TRANSMISSION MODEL

1) COMPUTATION MODEL

We assume that each device is equipped with a typical processing platform for neural networks, as described in [28] and [31]. This platform features a parallel neuron array and two levels of memory: a large main buffer and a smaller local buffer. The main buffer is substantial enough to store weights and activations, with half dedicated to storing inputs for the current layer and the other half to storing outputs. The second, smaller local buffer holds the weights and activations currently in use.

The energy consumption for reading from or writing to the local buffer is modeled to be equivalent to the energy of a single Multiply-Accumulate (MAC) operation, expressed as $E_{MAC}(n_k) = G(n_k/n_{max})^\beta$ where n_{max} represents the maximum precision level for computation. Here, G and β are constants related to the hardware specifications, with $G > 0$ and $1 < \beta < 2$. The energy required to access the main buffer is twice that of a MAC operation, calculated as $E_M = 2E_{MAC}(n_k)$. The total energy consumed by a device k for one local iteration is the sum of the computing energy $E_{cp}(n_k)$ and the energy costs $E_W(n_k)$ and $E_A(n_k)$ associated with retrieving weights and activations from the buffers, respectively. Accordingly, the total energy consumed for I local iterations is as follows.

$$E^C(n_k) = I\big(E_{cp}(n_k) + E_W(n_k) + E_A(n_k)\big) \tag{4}$$

where

$$\begin{aligned} E_{cp}(n_k) &= E_{MAC}(n_k)N_c + 2O_s E_{MAC(n_{max})} \\ E_W(n_k) &= E_M N_s + E_{MAC} N_c \sqrt{n_k/pn_{max}} \\ E_A(n_k) &= 2E_M O_s + E_{MAC} N_c \sqrt{n_k/pn_{max}}, \end{aligned} \tag{5}$$

with N_c representing the network complexity, N_s is the model size, i.e., number of weights and biases, and O_s is the total number of intermediate outputs throughout the whole network. $E_{cp}(n_k)$ is the sum of the energy consumed in biasing, batch normalization, and activation which are performed in full-precision n_{max} on each output feature O_s. $E_W(n_k)$ represents the energy once the weights are transferred from the main to the local buffer. Due to the parallelism in the activation levels, one weight is used simultaneously on $\sqrt{p}$ activations. Also, since a MAC unit can transfer more weights when quantizing with n_k bits compared to quantizing with n_{max} bits, we can reduce the access to the local buffer by $\sqrt{n_k/pn_{max}}$. The same applies to E_A as activations are retrieved/stored from/to the main buffer.

By substituting equation (2) into equation (4) and replacing each term with its corresponding formula, the computation energy consumption is obtained as follows

$$\begin{aligned} E^C(n_k) &= IG\left(\frac{n_k}{n_{max}}\right)^\beta \left(N_c + 2N_s + 2N_c\sqrt{\frac{n_k}{pn_{max}}} + 4O_s\right) \\ &\quad + 2IGO_s \\ &= B_1 n_k^\beta + B_2 n_k^{\beta+0.5} + B_3 \end{aligned}$$

where

$$\begin{aligned} B_1 &= \frac{IG(N_c + 2N_s + 4O_s)}{n_{max}^\beta}, B_2 = \frac{2N_c IG}{n_{max}^\beta \sqrt{pn_{max}}}, \\ B_3 &= 2IGO_s. \end{aligned} \tag{6}$$

2) TRANSMISSION MODEL

To transmit their parameter vectors, we assume that the N devices use orthogonal frequency domain multiple access (OFDMA). Each participating device k transmits its update with a power P_k during a designated time slot. Additionally, we account for transmission attenuation with distance. Accordingly, the received power of device k, at the server,

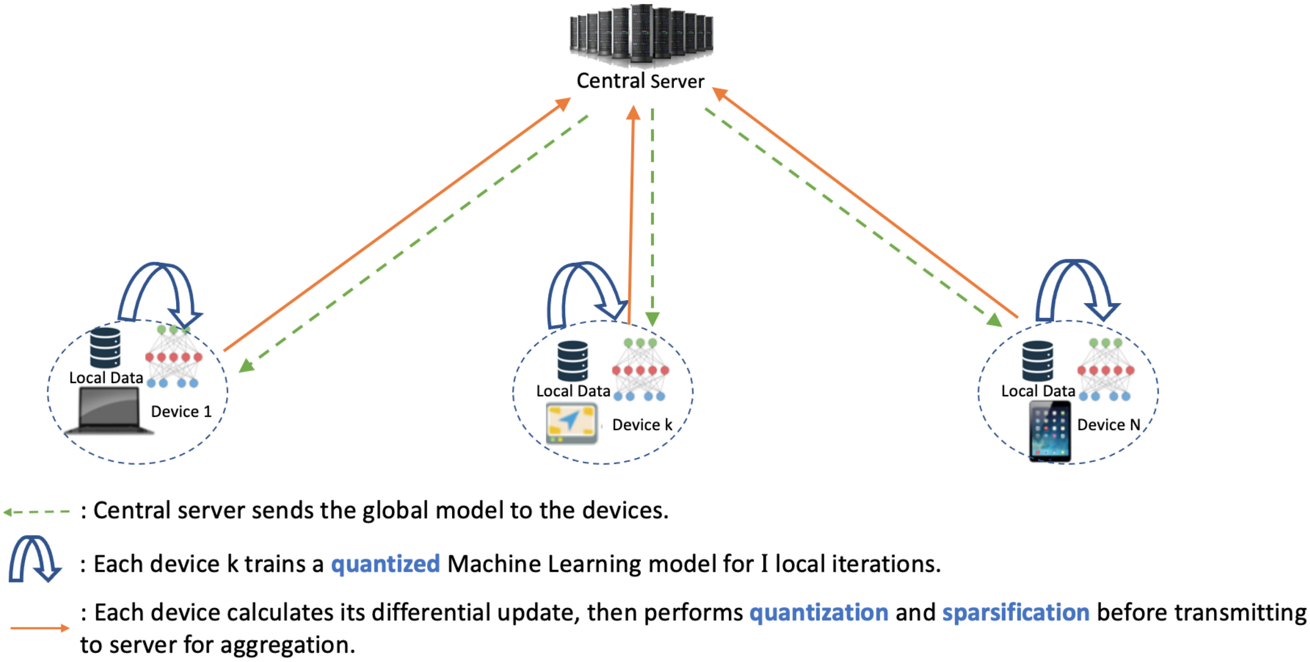

Central Server
Local Data
Device 1
Local Data
Device k
Local Data
Device N
: Central server sends the global model to the devices.
: Each device k trains a quantized Machine Learning model for I local iterations.
: Each device calculates its differential update, then performs quantization and sparsification before transmitting to server for aggregation.

FIGURE 1. SQFL Model.

is $P_k s_k^{-2}$, where s_k is the distance between device k and the server. Each device is allocated a bandwidth B. Consequently, the achievable data rate for device k can be calculated as

$$r_k = B\log_2\left(1 + \frac{P_k s_k^{-2}}{\sigma^2}\right), \tag{7}$$

where σ^2 is the power spectral density of the Gaussian noise. During each global round, a device k transmits a sparsified and quantized vector. Therefore, the size of the transmitted vector is $\gamma_k m_k d$ where d is the dimension of the ML model. Consequently, the transmission time, T_k, for the parameter vector is

$$T_k = \frac{\gamma_k m_k d}{r_k}. \tag{8}$$

Accordingly, the energy consumption for the uplink transmission is calculated as the product of the transmit power and the transmission time. This can be expressed as

$$E^{UL}(m_k, \gamma_k) = P_k T_k = P_k \frac{\gamma_k m_k d}{r_k}. \tag{9}$$

Finally, Figure 1 illustrates the previously described sparsified and quantized FL model over wireless networks. In the next section, we detail the general steps of the proposed FL approach under quantization and sparsification.

III. SPARSIFIED AND QUANTIZED FL ALGORITHM

The core concept of our proposed approach is to dynamically adjust the bit precision levels for quantization during both computation and communication phases, along with the sparsification ratio, for each device, with the objective to minimize the overall energy consumption. In the following, we explain how this approach is implemented within the framework of FL.

Following the standard federated averaging procedure (FedAvg) [2], each selected device locally trains its dataset by running multiple steps of SGD and sends the update to the central server for averaging. To reduce the computation energy, we assume that each device k trains a QNN with a precision number n_k. After I local iterations, selected devices send their updated parameter vectors to the server.

To lessen the communication overhead, a selected device k quantizes its update with a precision number m_k, then sparsifies the vector before transmitting it to the server. Algorithm 1 summarizes these steps. SQFL starts by selecting randomly K number of devices from $\mathcal{N}$ at each global round t. Each selected device k takes one step on the current model using its local data, then iterates the local update for I local rounds. The difference with FedAvg algorithm is that the weights used to update the local models are quantized with n_k precision bits.

Once device k completes I learning rounds, the differential update is computed $\boldsymbol{\delta}_{t+1}^{k} = \boldsymbol{w}_{t+1}^{k} - \boldsymbol{w}_{t}^{k}$ [27], where $\boldsymbol{w}_{t+1}^{k}$ and $\boldsymbol{w}_{t}^{k}$ are respectively defined in lines 13 and 12 of the algorithm. The update is then quantized with m_k precision bits (line 15) and sparsified (line 16) to obtain $\boldsymbol{\delta}_{t+1}^{k,Q^{m_k},S}$, which is transmitted to the server for aggregation. The new aggregated weight is computed $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{K}\sum_{k\in\mathcal{N}} \boldsymbol{\delta}_{t+1}^{k,Q^{m_k},S}$ and sent to the next participating devices. This process is repeated until converging to the target accuracy.

In the following section, we explore how to optimally select quantization levels and sparsification ratios to minimize the overall energy consumption while ensuring high accuracy and meeting latency requirements.

Algorithm 1 Sparsified and Quantized FL

```
$K$: number of devices indexed by $k$. Devices are randomly chosen from a set of $N$ devices. $\eta_t$: learning rate, t=0
Initialize $w^0$
repeat
    Device k do
        $w_{t,0}^k = w^k$
        for $\tau = 1, 2, \ldots, I$ local iterations do
            Compute the gradient $\nabla F^k(w_{t,\tau-1}^k)$
            Update the local model $w_{t,\tau}^k = w_{t,\tau-1}^k - \eta_t \times \nabla F^k(w_{t,\tau-1}^k)$
            Compute the quantized computation weight $w_{t,\tau}^{k,Q^{n_k}} = Q^{n_k}(w_{t,\tau}^k)$
            The new local weight is the quantized weight $w_{t,\tau}^k \leftarrow w_{t,\tau}^{k,Q^{n_k}}$
        end for
        $w_t^k = w_{t,0}^k$
        $w_{t+1}^k = w_{t,I}^k$
        Calculate the model differential update $\delta_{t+1}^k = w_{t+1}^k - w_t^k$
        Compute the quantized transmission update $\delta_{t+1}^{k,Q^{m_k}} = Q^{m_k}(\delta_{t+1}^k)$
        Apply sparsification to the quantized weight $\delta_{t+1}^{k,Q^{m_k},S} = S(\delta_{t+1}^{k,Q^{m_k}})$
    Global Server do
        Receive sparsified and quantized updates of the devices and perform aggregation $w_{t+1} = w_t + \frac{1}{K}\sum_{k\in\mathcal{N}} \delta_{t+1}^{k,Q^{m_k},S}$
        Send the global model $w^{t+1}$ to the devices
until target accuracy is reached
```

IV. ENERGY OPTIMIZATION FOR SPARSIFIED AND QUANTIZED FL

In this section, we begin by mathematically formulating the target optimization problem. Next, we perform a convergence analysis that allows us to replace the accuracy constraint in the original problem. This convergence analysis provides insights into the impact of the bit-precision and sparsification ratio on the number of communication rounds required for convergence. It also establishes a trade-off between the duration of the communication round and the target accuracy. Finally, we propose an approximation to the original optimization problem.

A. PROBLEM FORMULATION

Our main objective is to minimize the total energy consumption across all users while achieving a specified target accuracy and adhering to a latency constraint. Given a limit on the transmission time, it becomes important to send only a portion of the quantized weight to ensure convergence within the allowable time frame. To this purpose, we formulate an energy-efficient optimization problem. This problem aims to determine the optimal levels of sparsification and quantization by considering the trade-offs between communication overhead, computational cost, and system performance. Thus, our problem is formulated as follows

$$\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{\gamma}\,\mathbb{E}\left[\sum_{t=1}^{T}\sum_{k\in\mathcal{N}} E^{C}(n_k) + E^{UL}(m_k, \gamma_k)\right] \tag{10a}$$

$$\frac{\gamma_k m_k d}{r_k} \leq \tau \quad \forall k \in \mathcal{N}, \tag{10b}$$

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F\big(\boldsymbol{w}^*\big) \leq \epsilon, \tag{10c}$$

$$n_k \in \{1, \ldots, n_{\max}\} \quad \forall k \in \mathcal{N}, \tag{10d}$$

$$m_k \in \{1, \ldots, m_{\max}\} \quad \forall k \in \mathcal{N}, \tag{10e}$$

$$0 \leq \gamma_k \leq 1 \quad \forall k \in \mathcal{N}, \tag{10f}$$

where $\boldsymbol{m} = (m_k)_{k\in\mathcal{N}}$, $\boldsymbol{n} = (n_k)_{k\in\mathcal{N}}$, $\boldsymbol{\gamma_k} = (\gamma_k)_{k\in\mathcal{N}}$. Constraint (10b) ensures that the transmission time of device k does not exceed a round time threshold, thus preventing any devices from becoming stragglers in the system. Constraint (10c), where $F(\boldsymbol{w}^*)$ is the minimum value of F, guarantees the convergence to a target accuracy. Constraints (10d) and (10e) represent the possible values of quantization levels in computation and communication respectively. Finally, the sparsification level parameter is defined as a continuous variable that belongs to the interval [0, 1] in constraint (10f).

The studied problem is a challenging one for several reasons. First, it is important to note that the objective function and constraint (10b) are non-convex. Second, constraint (10c) is particularly challenging because it involves the minimum of the loss function, a value that cannot be determined prior to executing the FL training. Moreover, the presence of mixed variables, continuous and integers, further increases the complexity of finding a solution to the optimization problem.

To tackle this problem, we start by addressing the issue of the unknown minimum of the loss function. Specifically, we establish an upper-bound for the difference between the minimum of the loss function and its value after T iterations. The upper-bound enables us to construct a more restrictive optimization problem that effectively approximates the original optimization. The resulting approximate optimization remains challenging due to the non-convex nature of the objective function and the constraints, and the presence of mixed-integer variables. To address this, we first relax the integer constraints. We then decompose the problem into subproblems and propose an iterative approach to efficiently solve the approximate optimization problem.

In the next subsection, e derive an upper-bound of the convergence rate for the studied sparsified and quantized FL framework.

B. UPPER-BOUND FOR ACCURACY CONSTRAINT

To establish an upper bound for constraint (10c), we analyze the difference between the minimum value of the loss function and its value after T iterations under the following assumptions.

Assumption 1:

1) *L-smooth:* $F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{L}{2}||\boldsymbol{v} - \boldsymbol{w}||^2, \quad \forall \boldsymbol{v}, \boldsymbol{w}$

2) *$\mu-$strongly convex:* $F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{w}) + (\boldsymbol{v}-\boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{\mu}{2}||\boldsymbol{v}-\boldsymbol{w}||^2, \quad \forall \boldsymbol{v}, \boldsymbol{w}$
3) *Bounded variance for mini SGD:* The variance of stochastic gradients satisfies $\forall k = 1, \dots, N$
$$\mathbb{E}||\nabla F_k\left(\boldsymbol{w}_t^k, \zeta_t^k\right) - \nabla F_k\left(\boldsymbol{w}_t^k\right)||^2 \leq \sigma_k^2$$
4) *Uniformly bounded gradients*
$$\mathbb{E}||\nabla F_k\left(\boldsymbol{w}_t^k, \zeta_t^k\right)||^2 \leq H^2, \quad \forall k = 1, \dots N$$
5) *Bounded weight elements:* For constant $M \geq 0$, $||\boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{t}}||_\infty \leq M \quad \forall k = 1, \dots, N \ \forall t \in \{0, \dots, T\}$.

These assumptions are widely used in the convergence proof of various FL frameworks [27], [32]. To establish our theoretical results, we first show that the expected sparsified weight vector is bounded. This finding is outlined in the lemma below.

Lemma 1: Let S_k, $k \in \mathcal{N}$, be a sparsification operator that satisfies equation (3). The following inequality holds, $\forall \boldsymbol{w} \in \mathbb{R}^d$,
$$\mathbb{E}_{S_k}||S_k(\boldsymbol{w})||^2 \leq 4||\boldsymbol{w}||^2 \tag{11}$$

Proof: See Appendix C.1. ■

We leverage the result of this lemma to bound the expected error between the sparsified and quantized weight vector and the true vector value. This result is described in the subsequent lemma.

Lemma 2: Let Q_k and S_k, $k \in \mathcal{N}$ be quantization and sparsification operators that satisfy equations (2) and (3), respectively. Let $\boldsymbol{w} \in \mathbb{R}^d$ and $k \in \mathcal{N}$, the expected error between the quantized and sparsified vector and the actual vector can be upper-bounded as below
$$\mathbb{E}_{Q_k, S_k}||Q_k(S_k(\boldsymbol{w})) - \boldsymbol{w}||^2 \leq \frac{4dM^2}{(2^{m_k}-1)^2} + (1-\gamma_k)M^2, \tag{12}$$
where m_k is the bit-precision of the communication quantization scheme, γ_k is the sparsification rate of user k, d is the dimension of the weight vector, and M is the bound on the weight elements as described in Assumption (1–5).

Proof: See Appendix C.2. ■

Clearly, from equation (12), we can observe that as we increase the bit-precision level and the sparsification ratio, the error between the compressed vector (i.e., sparsified and quantized) and the true value of the weight vector is minimized.

In the following, we use the results of Lemma 2 to bound the difference between the minimum value of the loss function and its value after T iterations as described in the following theorem.

Theorem 1: Under Assumptions 1, with the learning rate $\eta_t = \frac{2}{\alpha+T}$, we have the following
$$\begin{aligned}&\mathbb{E}[F(\boldsymbol{w_T})] - F\left(\boldsymbol{w}^*\right)\\ &\leq \frac{2\kappa}{T+\alpha}\left[\frac{D}{\mu} + \left(2L + \frac{I\mu}{4}\right)||\boldsymbol{w_0} - \boldsymbol{w}^*||^2\right]\end{aligned} \tag{13}$$
where $\alpha = \max(8\kappa, I)$, and $\kappa = \frac{L}{\mu}$, and
$$\begin{aligned}D = D_1 + D_2 \sum_{k=1}^{N} \frac{1}{(2^{n_k}-1)^2} + D_3 \sum_{k=1}^{N}\\ \times\left[\frac{4d}{(2^{m_k}-1)^2} + (1-\gamma_k)\right],\end{aligned} \tag{14}$$
where $D_1 = \frac{4(N-K)}{K(N-1)} I^2 H^2 + \sum_{k=1}^{N} \frac{\sigma_k^2}{N^2} + 4(I-1)^2 H^2$, and $D_2 = \frac{d(1-\mu)}{N}$, and $D_3 = \frac{H^2 I^2}{N}$.

Proof: See Appendix C.3. ■

From equation (13), we infer that high quantization levels, n_k and m_k, along with high sparsification ratio γ_k can significantly decrease the number of rounds required for convergence. However, reducing the number of communication rounds does not necessarily shorten the duration of each round itself.

This trade-off between the number of rounds required for convergence, the bit-precision, the sparsification ratio and the duration of the communication round is captured in the following proposition.

Proposition 1: Assume homogeneous bit precision $m_k = m$ and sparsification ratio $\gamma_k = \gamma \ \forall k \in \mathcal{N}$. The approximate optimization problem, i.e., with the new constraint,
$$\frac{2\kappa}{T+\alpha}\left[\frac{D}{\mu} + \left(2L + \frac{I\mu}{4}\right)||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2\right] \leq \epsilon, \tag{15}$$
is feasible iff the following inequalities are satisfied
$$1 - \frac{\epsilon\mu(T+\alpha)}{2\kappa H^2 I^2} + \frac{4d}{(2^m-1)^2} \leq \gamma \leq \frac{r\tau}{dm}, \tag{16}$$
with $r = \max_{k \in \mathcal{N}} r_k$.

Proof: See Appendix C.4. ■

Equation (16) effectively illustrates the trade-off between accuracy and transmission delay. From the left-hand side of equation (16), it is evident that to enhance accuracy (i.e., achieve a smaller ϵ, a larger sparsification ratio γ is required. This implies that for higher convergence levels, a greater portion of the update values needs to be transmitted. Moreover, increasing the number of transmission bits m necessitates a decrease in the sparsification ratio γ to maintain the desired accuracy.

From the right-hand side of equation (16), a larger γ implies that the transmission delay threshold must also be increased. Conversely, transmitting a highly sparse update reduces the time required but at the expense of slower learning.

C. APPROXIMATE OPTIMIZATION FOR PROBLEM (10)

To efficiently solve problem (10), we first notice that the objective function can be simplified as follows
$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} \sum_{k \in \mathcal{N}} E^C(n_k) + E^{UL}(m_k, \gamma_k)\right]\\ &= \frac{TK}{N} \sum_{k=1}^{N} E^C(n_k) + E^{UL}(m_k, \gamma_k).\end{aligned} \tag{17}$$

Based on Theorem 1, we can approximate the problem (10), by letting the analytical convergence upper bound in (13) satisfy the convergence constraint in (10c), as follows

$$\underset{\boldsymbol{n},\boldsymbol{m},\boldsymbol{\gamma}}{\text{minimize}} \quad \frac{TK}{N}\sum_{k=1}^{N}\Big(E^{C}(n_k)+E^{UL}(m_k,\gamma_k)\Big)\frac{2\kappa}{T+\alpha} \tag{18a}$$

$$\text{subject to} \quad \left[\frac{D}{\mu}+\big(2L+\frac{I\mu}{4}\big)||\boldsymbol{w}_0-\boldsymbol{w}^{*}||^2\right] \tag{18b}$$

$$\leq \epsilon \text{Constraints (10b), (10d), (10e), (10f)} \tag{18c}$$

The approximated problem (18) is more constrained than problem (10). Therefore, any feasible point for problem (18) is also feasible for problem (10) [17], [33], [34]. Next, we take the equality in (18b) to obtain

$$\begin{aligned} T &= \frac{1}{\epsilon}2\kappa\left[\frac{D}{\mu}+\big(2L+\frac{I\mu}{4}\big)||\boldsymbol{w}_0-\boldsymbol{w}^{*}||^2\right]-\alpha \\ &= T_1D+T_2, \end{aligned} \tag{19}$$

where $T_1=\frac{2\kappa}{\epsilon\mu}$, and $T_2=\frac{2\kappa}{\epsilon}(2L+\frac{I\mu}{4})||\boldsymbol{w}_0-\boldsymbol{w}^{*}||^2-\alpha$. Then, we can change the original problem as below

$$\underset{\boldsymbol{n},\boldsymbol{m},\boldsymbol{\gamma}}{\text{minimize}} \quad A(\boldsymbol{n},\boldsymbol{m},\boldsymbol{\gamma}) \tag{20a}$$

$$\text{subject to} \quad \text{Constraints (10b), (10d), (10e), (10f)} \tag{20b}$$

with $A(\boldsymbol{n},\boldsymbol{m},\boldsymbol{\gamma}) = (T_1D+T_2)\times\frac{K}{N}\sum_{k=1}^{N}(E^{C}(n_k)+E^{UL}(m_k,\gamma_k))$

Note that the resulting optimization problem is non-convex because of the non-convexity of the objective function and constraints, and the presence of mixed-integer variables. To tackle this, we relax n_k and m_k as continuous variables. These relaxed variables will be rounded back to integer values for feasibility. Next, we propose an iterative algorithm to solve this problem.

V. ITERATIVE APPROACH FOR AN OPTIMIZED SQFL FRAMEWORK

In this section, we present an alternating algorithm to iteratively solve problem (20) through optimizing three subproblems, i.e., computation quantization subproblem, transmission quantization subproblem, and transmission sparsification subproblem.

A. COMPUTATION QUANTIZATION SUBPROBLEM

First, we assume that the quantization vector $\boldsymbol{m}$ and sparsification ratio vector $\boldsymbol{\gamma}$ are known. Therefore, the problem (20) can be simplified as

$$\underset{n}{\text{minimize}} \quad An \tag{21a}$$

$$\text{subject to} \quad \boldsymbol{n}A(\boldsymbol{n})1 \leq n_k \leq n_{\max} \quad \forall k\in\mathcal{N} \tag{21b}$$

Problem (21) is convex with regard to $\boldsymbol{n}$ and thus can easily be solved using convex optimization algorithms.

Proof: To prove the convexity of the optimization problem (21), we need to show that the second derivative of the objective function $A(\boldsymbol{n})$ is positive. We first simplify $A(\boldsymbol{n})$ as follow

$$\begin{aligned} A(n) = &\left[C_1+C_2\sum_{k=1}^{N}\frac{1}{(2^{n_k}-1)^2}\right]\frac{K}{N} \\ &\times\left[\sum_{k=1}^{N}B_1n_k^{\beta}+B_2n_k^{\beta+0.5}+B_3+C_3\right] \end{aligned}$$

where $C_1=\frac{2\kappa}{\epsilon\mu}(D_1+D_3\sum_{k=1}^{N}[\frac{4d}{(2^{m_k}-1)^2}+(1-\gamma_k)]+\mu(2L+\frac{I\mu}{4}))||\boldsymbol{w}_0-\boldsymbol{w}^{*}||^2)-\alpha$, and $C_2=\frac{2\kappa}{\epsilon\mu}D_2$, and $C_3=P_k\frac{\gamma_km_kd}{r_k}$. The variables D_1, D_2 and D_3 are defined in equation (14), and variables B_1, B_2 and B_3 are defined in equation (6). Let

$$A_1(n_k)=\frac{1}{(2^{n_k}-1)^2}{n_k}^{\beta}, \qquad A_2(n_k)=\frac{1}{(2^{n_k}-1)^2}n_k^{\beta+0.5} \tag{22}$$

We derive $A_1(n_k)$ two times to obtain the following

$$\begin{aligned} \partial[2]A_1(n_k)n_k = &\frac{(\beta-1)an_k^{\beta-2}}{\left(2_k^{n}-1\right)^2}-\frac{\beta 2^{n_k+2}ln(2)n_k^{\beta-1}}{(2^{n_k}-1)^3} \\ &+n_k^{\beta}\left(\frac{3\times 2^{2n_k+1}ln(2)^2}{(2^{n_k}-1)^4}-\frac{2^{n_k+1}ln(2)^2}{(2^{n_k}-1)^3}\right) \end{aligned} \tag{23}$$

To prove that $\partial[2]A_1(n_k)n_k \geq 0$, it is sufficient to prove that $[\frac{2^{n_k+1}ln(2)^2n_k^{\beta}}{(2^{n_k}-1)^3}](-\frac{2\beta}{ln(2)n_k}+\frac{3\times 2^{n_k}}{2^{n_k}-1}-1)\geq 0$, and thus, sufficient to prove that $A_{11}=-(2^{n_k}-1)2\beta+3\times 2^{n_k}ln(2)n_k-ln(2)n_k(2^{n_k}-1)\geq 0$

$$\begin{aligned} A_{11} &= 2^{n_k+1}n_kln(2)+2\beta-2^{n_k+1}\beta+n_kln(2) \\ &= 2^{n_k+1}(n_kln(2)-\beta)+2\beta+n_kln(2) \end{aligned} \tag{24}$$

Since $\beta<2$, to ensure the convexity of $A_1(n_k)$, it is sufficient to set $n_k\geq 3$.

We conduct similar reasoning to prove the convexity of $A_2(n_k)$. ■

B. TRANSMISSION QUANTIZATION SUBPROBLEM

With given computation quantization level $\mathbf{n}$ and transmission sparsification level $\boldsymbol{\gamma}$, problem (20) can be simplified as

$$\underset{\boldsymbol{m}}{\text{minimize}} \quad A(\boldsymbol{m}) \tag{25a}$$

$$\text{subject to} \quad \frac{\gamma_km_kd}{r_k}\leq\tau \quad \forall k\in\mathcal{N} \tag{25b}$$

$$1\leq m_k\leq m_{\max} \quad \forall k\in\mathcal{N} \tag{25c}$$

where

$$A(\boldsymbol{m})=\left[M_1+M_2\sum_{k=1}^{N}\frac{1}{(2^{m_k}-1)^2}\right]\frac{K}{N}\left[\sum_{k=1}^{N}M_3+M_4m_k\right],$$

with $M_1=\frac{2\kappa}{\epsilon\mu}(D_1+D_2\sum_{k=1}^{N}\frac{1}{(2^{n_k}-1)^2}+D_3\sum_{k=1}^{N}[(1-\gamma_k)]+\mu(2L+\frac{I\mu}{4})||\boldsymbol{w}_0-\boldsymbol{w}^{*}||^2)-\alpha$, and $M_2=D_34d$, and $M_3=$

Algorithm 2 SQFL Iterative Algorithm

```
Input: A feasible solution $(\mathrm{n}^0, \mathrm{m}^0, \boldsymbol{\gamma}^0)$, $\epsilon_1 > 0$ and iteration number $\ell = 1$
repeat
    With given $m^{\ell-1}$, $\boldsymbol{\gamma}^{\ell-1}$, solve the computing quantization subproblem and obtain the solution $n^\ell$
    With given $n^\ell$, $\boldsymbol{\gamma}^{\ell-1}$, solve the transmission quantization subproblem and obtain the solution $m^\ell$
    With given $n^\ell$, $m^\ell$, solve the transmission sparsification subproblem and obtain the solution $\boldsymbol{\gamma}^\ell$
    Set $\ell = \ell + 1$
until $|A(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{\gamma})^\ell - A(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{\gamma})^{\ell-1}| < \epsilon_1$
```

$E^C(n_k)$, and $M_4 = P_k \frac{\gamma_k d}{r_k}$. From (22), and by considering $\beta = 1$ we define $A_1(m_k) = \frac{1}{(2^{m_k}-1)^2} m_k$ which is convex as already proven in the previous subsection.

C. TRANSMISSION SPARSIFICATION SUBPROBLEM

Here, we focus on optimizing the sparsification ratio for all devices. Let the computation quantization vector $\boldsymbol{n}$ and transmission quantization vector $\boldsymbol{m}$ be fixed. The sparsification ratio optimization (20) can be written

$$\underset{\boldsymbol{\gamma}}{\text{minimize}} \quad A(\boldsymbol{\gamma}) \tag{26a}$$

$$\text{subject to} \quad \frac{\gamma_k m_k d}{r_k} \leq \tau \quad \forall k \in \mathcal{N} \tag{26b}$$

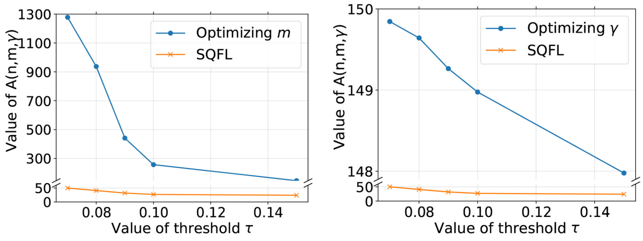
$$0 \leq \gamma_k \leq 1 \quad \forall k \in \mathcal{N} \tag{26c}$$

where

$$A(\boldsymbol{\gamma}) = \left[\Gamma_1 + \Gamma_2 \sum_{k=1}^{N} -\gamma_k\right] \frac{K}{N} \left[\sum_{k=1}^{N} \Gamma_3 + \Gamma_4 \gamma_k\right],$$

with $\Gamma_1 = \frac{2\kappa}{\epsilon\mu}(D_1 + D_2 \sum_{k=1}^N \frac{1}{(2^{n_k}-1)^2} + D_3 \sum_{k=1}^N [\frac{4d}{(2^{m_k}-1)^2} + 1] + \mu(2L + \frac{I\mu}{4})||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2) - \alpha$, and $\Gamma_2 = D_3$, and $\Gamma_3 = M_3$, and $\Gamma_4 = P_k \frac{m_k d}{r_k}$. The problem in (26) is quadratic and thus can be efficiently solved.

Finally, Algorithm 2 gives the solution of the optimization problem stated in (20) by iteratively solving problems in (21), (25), and (26). At each iteration, the optimal solutions of (21), (25), and (26) are obtained, and thus, the objective in (20) is decreased. Furthermore, since the parameters are all positive, the objective is also positive (i.e., lower-bounded by 0). Therefore, the convergence of the algorithm to a local optimum is guaranteed.



(a) Optimizing quantization value m. (b) Optimizing sparsification value γ.

FIGURE 2. Value of the objective function with respect to the optimized parameters.

D. COMPLEXITY ANALYSIS

To solve the problem (20) using Algorithm 2, the complexity depends on the number of iterations $\mathcal{L}$ of the iterative minimization Algorithm 2, and on the complexity needed to solve each of the three subproblems. Problems (21), (25), and (26) are solved with a complexity of $K\mathcal{O}\frac{1}{\epsilon_2}$, $K\mathcal{O}\frac{1}{\epsilon_3}$, and $K\mathcal{O}(\log_2 \frac{1}{\epsilon_4})$ respectively where ϵ_2, ϵ_3, and ϵ_4 are the accuracies of the subproblems [35]. As a result, the total complexity of Algorithm 2 can be evaluated as $\mathcal{O}(\mathcal{L}K(\frac{1}{\epsilon_2} + \frac{1}{\epsilon_3} + \log_2 \frac{1}{\epsilon_4}))$.

VI. SIMULATION EXPERIMENTS

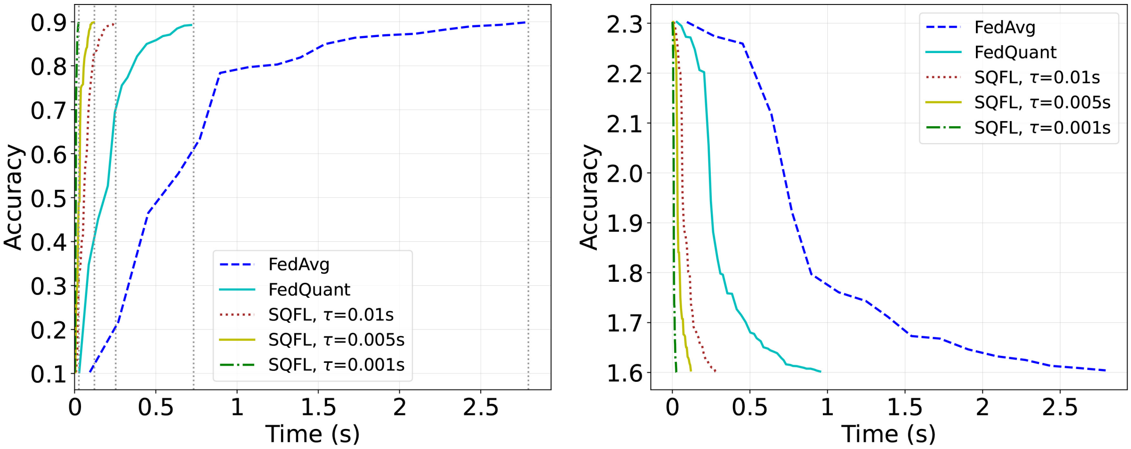
In this section, we conduct experiments to test the efficiency of our approach. The performance is measured in terms of accuracy, completion time, and consumed energy. We suppose that all devices have the same computational capabilities, hence, we focus on time and energy consumed for transmission. Our experiments are conducted using Keras with TensorFlow. We explore two different datasets: i): we train a convolutional neural network with 2×10^6 parameters on a non-iid partitioned MNIST [36], a handset of handwritten numbers from 0 to 9; ii): we train a convolutional neural network with 7×10^6 parameters on a non-iid partitioned CIFAR-10, a dataset of colored images from 10 categories. We consider an area of 1 km^2 where 100 devices are scattered randomly and communicate with a server in the middle of the area. We assume a total bandwidth of $B = 10$ MHz and a power $P = 0.1$ mW. The power spectral density of the Gaussian noise is equal to $\sigma^2 = 10^{-12}$. At each global round, we select 5 different devices at random to perform 7 local iterations.

To evaluate the performance of our proposed iterative algorithm, we compare it to an exhaustive search method on a limited range of $\boldsymbol{n}$, $\boldsymbol{m}$, and $\boldsymbol{\gamma}$ and for number of devices $K = 2$ and $K = 3$. The exhaustive search was conducted on a list of 8 elements from 8 to 15 for quantization levels $\boldsymbol{n}$ and $\boldsymbol{m}$ and on a list of 14 elements from 0.1 to 0.9 for sparsification level γ. Both the methods were performed on a MacBook pro with a 2.2 Ghz Quad-Core Intel Core i7 processor and 16 Go of RAM. The results are displayed in Table 1. While respecting the specific time threshold of $\tau = 5$ ms, and using the iterative algorithm, our method SQFL is giving the parameters that successfully minimize the objective value. The exhaustive search on the other hand is time-consuming method even on limited search range and especially when the number of devices increases. For example when $K = 3$, the execution time of SQFL is less × 30 than the exhaustive search.

To assess the impact of the problem's parameters on the objective function, we plot the value of the objective when optimizing one parameter while fixing the others (in their maximum values) in Figure 2. Since the computation parameter $\boldsymbol{n}$ is not involved in the transmission time

TABLE 1. Comparison with exhaustive search method on limited range of n, m, γ, and for $\tau = 0.005$ s.

K	Method	γ	n	m	Objective	Execution time
2	SQFL	$\gamma_1 = 0.23$ $\gamma_2 = 0.4$	$n_1 = 9$ $n_2 = 9$	$m_1 = 13$ $m_2 = 13$	16.4	5 s
2	Exhaustive Search	$\gamma_1 = 0.1$ $\gamma_2 = 0.4$	$n_1 = 9$ $n_2 = 9$	$m_1 = 14$ $m_2 = 12$	17.26	11 s
3	SQFL	$\gamma_1 = 0.23$ $\gamma_2 = 0.4$ $\gamma_3 = 0.21$	$n_1 = 10$ $n_2 = 10$ $n_3 = 10$	$m_1 = 13$ $m_2 = 13$ $m_3 = 13$	21.4	6 s
3	Exhaustive Search	$\gamma_1 = 0.2$ $\gamma_2 = 0.4$ $\gamma_3 = 0.17$	$n_1 = 10$ $n_2 = 10$ $n_3 = 10$	$m_1 = 14$ $m_2 = 12$ $m_3 = 13$	21.77	183 s



(a) Test accuracy.

(b) Train loss.

FIGURE 3. Time needed to achieve a target accuracy (90%) and target loss (1.6) for MNIST.

constraint, it remains fixed when varying the value of τ, although it has an impact on minimizing the objective function. Additionally, by considering the maximum values of the parameters (for example $\gamma_k = 1$ or $m_k = 32$) for a device k, the time threshold τ can not be less than 0.07 s. We could see that the quantization parameter $\boldsymbol{m}$ has a more important variation depending on the time budget. Furthermore, our proposed framework SQFL has the best performance among both schemes. This is because our approach jointly optimizes quantization values $\boldsymbol{n}$, $\boldsymbol{m}$, and sparsification value $\boldsymbol{\gamma}$ to reach the optimal minimization level.

We compare our proposed scheme with the following two different energy efficient FL schemes:

- FedAvg [2]: selects devices uniformly at random and exchanges the entire models between the server and selected devices.
- FedQuant [28]: optimizes quantization levels for both computation and communication and shares the entire model.

Besides, for our evaluation, we also consider different time thresholds τ leading to different quantization and sparsification levels:

- *Case when $\tau = 0.01$:* For this setup, the average results of the participating devices for $\boldsymbol{n}$ and $\boldsymbol{m}$ respectively are 11 and 13. Due to the limited time budget, only a portion of the update could be sent, the resulting average $\boldsymbol{\gamma}$ for the participating devices is 0.26.
- *Case when $\tau = 0.005$:* For this setup, the average of $\boldsymbol{n}$ and $\boldsymbol{m}$ are respectively equal to 11 and 14 and the average sparsification level $\boldsymbol{\gamma}$ is equal to 0.13.
- *Case when $\tau = 0.001$:* This setup represents the shortest time budget. Consequently, only 2% of the update is communicated, i.e., $\boldsymbol{\gamma} = 0.02$ on average. Moreover, the average quantization levels of the participating devices, $\boldsymbol{n}$ is equal to 11 and $\boldsymbol{m}$ is equal to 15.

Figure 3 shows the accuracy and loss with respect to time. The training duration is correlated with the amount of updates communicated. We can see that our method is outperforming the state of the art in terms of accuracy and loss. More precisely, when the time budget is limited, SQFL chooses the best compression levels that minimize the wireless resources while meeting the target accuracy. Hence, depending on the time threshold τ, our framework converges to a 90% accuracy in less than half a second. On

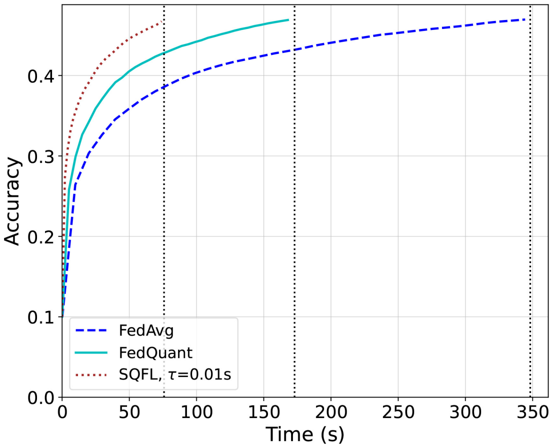


FIGURE 5. Time needed to achieve a target accuracy (45%) for CIFAR-10.

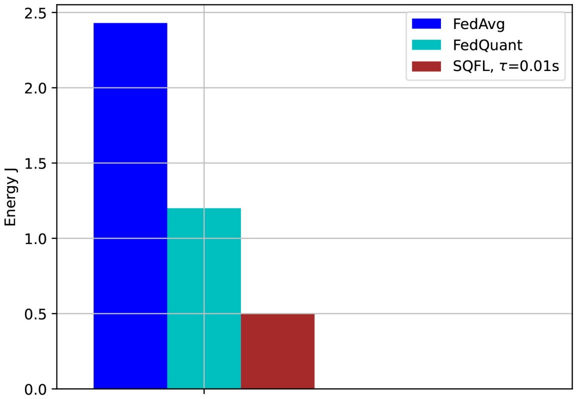


FIGURE 6. Energy needed to achieve (45%) for CIFAR-10.

the other hand, in the FedAvg setup, more than × 5 the time is needed to achieve the same accuracy. As for FedQuant, the convergence takes more time than SQFL as it uses only quantization.

In Figure 4, we analyze the consumed energy for the different approaches. In the graph on the left-hand side, we see that for each communication round, we consume less energy using SQFL. The graph on the right-hand side shows a considerable energy gain while using our method compared to FedAvg and FedQuant. More precisely, to achieve a target accuracy of 90%, we consume × 96 the energy in FedAvg and × 27 the energy in FedQuant. This is because SQFL aims to minimize the overall energy consumption while respecting the time and accuracy constraints.

We validate the previous conclusions by using a more complex dataset, namely CIFAR-10 in Figure 5 and Figure 6. To achieve the same accuracy, and using our method, we gain up to × 4.6 in speed compared to FedAvg and up to × 2.3 in speed compared to FedQuant. Furthermore, the total energy is considerably reduced. This is because our framework allows compression in two levels (training and communication), which saves both energy and time, without degrading the performance.

VII. CONCLUSION

In this paper, we present SQFL, an FL compression framework that tackles energy efficiency. Our method selects the optimal quantization parameters in both training and communication, and the best sparsification levels in transmission to minimize the total energy consumed while respecting the time constraint and ensuring convergence. To solve the proposed optimization problem, we first derive the convergence bound and express it with the parameter's variables, then we approximate the problem by stating the relationship between the total energy consumption in the objective and the convergence rate constraint. Finally, to solve the resulting approximated problem, we present an efficient iterative algorithm with guarantees of convergence and low time complexity. We perform diverse experiments to validate the efficiency of our approach. In particular, our numerical results show that SQFL significantly reduces energy consumption and total learning time when compared to benchmark schemes. Furthermore, it achieves results comparable to those of exhaustive research while considerably decreasing the execution time.

As a future work, we will investigate our approach in a dynamic setup where the position of the devices changes over time. Furthermore, we will explore the impact of other parameters (e.g., number of local and global iterations, and number of sampled devices) and how they affect the convergence rate and energy consumption.

APPENDIX

A. NOTATIONS

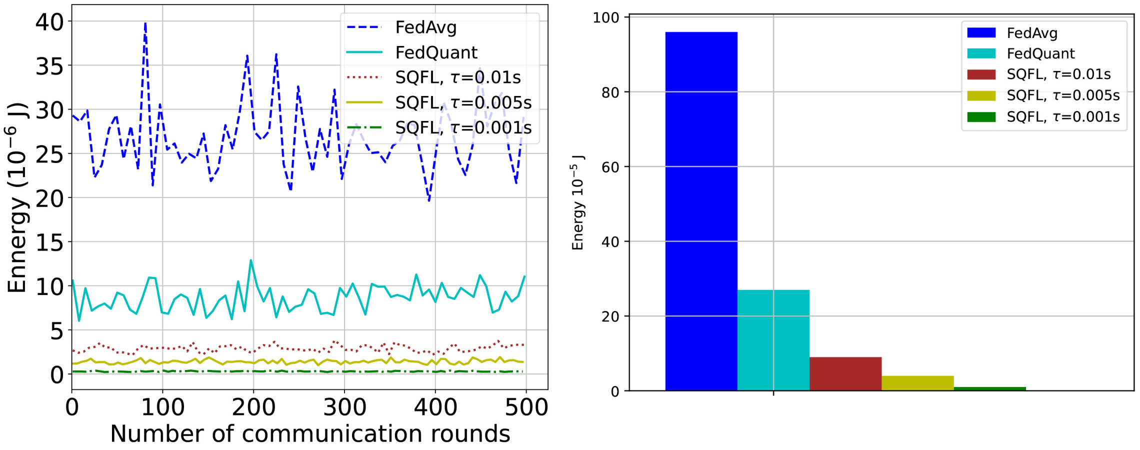
In our analysis, we consider four sources of randomness: stochastic gradients, random sampling of devices, quantization, and sparsification. We use different notations to distinguish their related expectations. $\mathbb{E}_G$ denotes the expectation over the stochastic gradients, $\mathbb{E}_{\mathcal{S}_t}$ designates the expectation over selected devices, $\mathbb{E}_Q$ represents the expectation over the quantization scheme, and finally $\mathbb{E}_S$ is the expectation over the sparsification. We also use $\mathbb{E}$ to denote the expectation over the joint four sources of randomness.

Let $\boldsymbol{w}_t^k$ be the parameter vector of the k^{th} device at the t^{th} iteration. We suppose that each device performs I local iterations before transmitting its local model to the server. Let $R_I = \{jI|j = 1, 2, \dots\}$ be the set of global iterations. For convenience, we adopt the following notations.

$$\boldsymbol{v}_{t+1}^k = \boldsymbol{w}_t^k - \eta_t \nabla F_k\left(\boldsymbol{w}_t^k, \zeta_t^k\right). \tag{27}$$

$$\boldsymbol{u}_{t+1}^k = \begin{cases} \boldsymbol{v}_{t+1}^k & \text{if } t+1 \notin R_I \\ \frac{1}{K}\sum_{i\in\mathcal{S}_{t+1}} \boldsymbol{v}_{t+1}^i & \text{if } t+1 \in R_I \end{cases} \tag{28}$$

$$w_{t+1}^k = \begin{cases} Q^n\left(\boldsymbol{v}_{t+1}^k\right) \\ \text{if } t+1 \notin R_I \\ \boldsymbol{w}_{t+1-I} + \frac{1}{K}\sum_{i\in\mathcal{S}_{t+1}} Q^m\left(S(\delta_{t+1}^i)\right)_{t+1} \\ \text{if } t+1 \in R_I \end{cases} \tag{29}$$



(a) Energy per round following different τ.   (b) Energy needed to achieve 90% following different τ.

FIGURE 4. **Consumed Energy for transmission for MNIST.**

In fact, if $t+1$ is a local iteration, the parameter vector $\boldsymbol{w}_{t+1}^k$ is equal to the quantized local parameter vector $Q^n(\boldsymbol{v}_{t+1}^k)$. The quantization of $\boldsymbol{v}_{t+1}^k$ is performed after each local update using n_k bit-precision (with some abuse of notation, we drop the k in Q^{n_k} when we represent the computation quantization). If $t+1$ is a global iteration, $\boldsymbol{w}_{t+1}^k$ is the aggregated sum of the compressed and quantized local vectors, i.e., $\frac{1}{K}\sum_{i\in\mathcal{S}_{t+1}} Q^m(S(\boldsymbol{v}_{t+1}^i))_{t+1}$. In a global iteration, we consider the communication quantization, which is represented by Q^m, where m indicates the bit-precision of the quantization. Typically, each device k transmits with m_k bit-precision. It also compresses its updates with γ_k compression rate. The vector $\boldsymbol{u}_{t+1}^k$ is the global model vector before any compression.

To facilitate the analysis, we consider the following virtual sequences: $\bar{\boldsymbol{v}}_{t+1} = \frac{1}{N}\sum_{k=1}^N \boldsymbol{v}_{t+1}^k$, $\bar{\boldsymbol{u}}_{t+1} = \frac{1}{N}\sum_{k=1}^N \boldsymbol{u}_{t+1}^k$, $\bar{\boldsymbol{w}}_{t+1} = \frac{1}{N}\sum_{k=1}^N \boldsymbol{w}_{t+1}^k$ which are the averaged sum over all devices of local updates, global updates before the compression, and the global weight vector respectively. We also use $\boldsymbol{g}_t = \frac{1}{N}\sum_{k=1}^N \nabla F^k(\boldsymbol{w}_t^k, \zeta_t^k)$, and $\bar{\boldsymbol{g}}_t = \frac{1}{N}\sum_{k=1}^N \nabla F^k(\boldsymbol{w}_t^k)$.

B. USEFUL LEMMAS

In the following, we present 4 important lemmas for the proof of our theorems. The Lemmas 3, 4 and 5 have been proved in [32] under Assumption 1 without considering any compression. However, they still hold under our quantization and sparsification schemes.

Lemma 3: Let Assumption 1 holds. If $\eta_t \le \frac{1}{4L}$, we have

$$\begin{aligned}\mathbb{E}_G||\bar{\boldsymbol{v}}_{t+1} - \boldsymbol{w}^*||^2 &\le (1-\eta_t\mu)\mathbb{E}_G||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2\\ &+\eta_t^2\mathbb{E}_G||\boldsymbol{g}_t - \bar{\boldsymbol{g}}_t||^2 + 6L\eta_t^2\gamma\\ &+2\mathbb{E}_G\frac{1}{N}\sum_{k=1}^N ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^k||^2. \end{aligned} \tag{30}$$

Lemma 4: Let Assumption 1 holds. It follows that

$$\mathbb{E}_G||\bar{\boldsymbol{g}}_t - \boldsymbol{g}_t||^2 \le \sum_{k=1}^N \frac{\sigma_k^2}{N^2}. \tag{31}$$

Lemma 5: Let Assumption 1 holds,η_t is non-increasing and $\eta_t \le 2\eta_{t+I}$ $\forall t \ge 0$. It follows that

$$\mathbb{E}_G\Big[\frac{1}{N}\sum_{k=1}^N ||\bar{\boldsymbol{w}}_t - \boldsymbol{w}_t^k||^2\Big] \le 4\eta_t^2(I-1)^2H^2. \tag{32}$$

We also use the following Lemma from [27] which still holds under our assumptions.

Lemma 6: Let Assumption 1 hold. We have $\forall t \ge 0$

$$\mathbb{E}_{\mathcal{S}_t}\big[\bar{\boldsymbol{u}}_{t+1}\big] = \bar{\boldsymbol{v}}_{t+1} \tag{33}$$

and for $t+1 \in R_I$, assume $\eta_t \le \eta_{t+I}$, it follows

$$\mathbb{E}||\bar{\boldsymbol{v}}_{t+1} - \bar{\boldsymbol{u}}_{t+1}||^2 \le \frac{N-K}{N-1}\frac{4}{K}\eta_t^2 I^2 H2\cdot \tag{34}$$

C. PROOFS

1) PROOF OF LEMMA 1

We have

$$\begin{aligned}&\mathbb{E}_S||\,S(\boldsymbol{w}) - \boldsymbol{w}||_2^2 - (1-\gamma)||\boldsymbol{w}||^2\\ &\quad= \mathbb{E}_S||\,S(\boldsymbol{w})||^2 - <\mathbb{E}[S(\boldsymbol{w})], \boldsymbol{w}> +\gamma||\boldsymbol{w}||^2\\ &\quad\le 0.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}_S||\,S(\boldsymbol{w})||^2 &\le 2\ <\mathbb{E}[S(\boldsymbol{w})], \boldsymbol{w}>\\ &\overset{(a)}{\le} 2||\mathbb{E}[S(\boldsymbol{w})]||\ ||\boldsymbol{w}||\end{aligned} \tag{35}$$

where (a) comes from Cauchy-Schwartz inequality. We also know that

$$||\mathbb{E}_S[S(\boldsymbol{w})]||^2 \le \mathbb{E}_S||\,S(\boldsymbol{w})||^2 \tag{36}$$

By combining equations (32) and (36), we obtain

$$||\mathbb{E}_S[S(\boldsymbol{w})]||^2 \le 2||\mathbb{E}_S[S(\boldsymbol{w})]||\ ||\boldsymbol{w}||.$$

if $\mathbb{E}_S[S(\boldsymbol{w})] \neq 0$, we obtain

$$||\mathbb{E}_S[S(\boldsymbol{w})]|| \le 2||\boldsymbol{w}|| \tag{37}$$

else if $\mathbb{E}_S[S(\mathbf{w})] = 0$, equation (37) remains correct. We replace by inequality (37) in inequality (32), and obtain

$$\mathbb{E}_S||\,S(\mathbf{w})\,||^2 \leq 4||\mathbf{w}||^2$$

2) PROOF OF LEMMA 2

We decompose the LHS of inequality (12) into 3 terms.

$$\begin{aligned}\mathbb{E}_{Q,S}&||Q(S(\mathbf{w})) - \mathbf{w}||^2\\ &= \underbrace{\mathbb{E}_{Q,S}\Big[||Q(S(\mathbf{w})) - S(\mathbf{w})||^2\Big]}_{B_1}\\ &+ \underbrace{\mathbb{E}_S\Big[||\,S(\mathbf{w}) - \mathbf{w}||^2\Big]}_{B_2}\\ &+ \underbrace{2\mathbb{E}_{Q,S} < Q(S(\mathbf{w})) - S(\mathbf{w}), S(\mathbf{w}) - \mathbf{w} >}_{B_3}\end{aligned}$$

First, we have,

$$\begin{aligned}B_3 &= 2\mathbb{E}_S < \mathbb{E}_Q[Q(S(\mathbf{w}))] - S(\mathbf{w}), S(\mathbf{w}) - \mathbf{w} >\\ &\overset{(a)}{=} 0 \qquad (38)\end{aligned}$$

(a) due to the unbiasedness of the quantization operator. Second, B_2 is bounded using equation (3). Finally, B_1 can be upper-bounded using the same reasoning as in Lemma 5 in [27]. In particular, according to Lemma 5 in [27] $\forall \mathbf{v} \in \mathbb{R}^d$, we have

$$\mathbb{E}_Q\Big[(Q(\mathbf{v}) - \mathbf{v})^2\Big] \leq \Big(\frac{|\mathbf{v}|}{2^{m_k} - 1}\Big)^2. \qquad (39)$$

Hence, for $\mathbf{w} \in \mathbb{R}^d$, we have

$$\begin{aligned}\mathbb{E}_{Q,S}||Q(S(\mathbf{w})) - \mathbf{w}||^2 &= \mathbb{E}_{Q,S}\Big[\sum_{i=1}^{d}(Q(S(w_i)) - S(w_i))^2\Big]\\ &= \mathbb{E}_S\Big[\sum_{i=1}^{d}\mathbb{E}_Q\Big[(Q(S(w_i)) - S(w_i))^2\Big]\Big]\\ &\overset{(a)}{\leq} \mathbb{E}_S\Big[\sum_{i=1}^{d}\frac{(|S(w_i)|)^2}{(2^{m_k} - 1)^2}\Big]\\ &\overset{(b)}{\leq} \frac{4dM^2}{(2^{m_k} - 1)^2} \qquad (40)\end{aligned}$$

(a) comes from equation (39), and (b) because $\mathbb{E}_S||\,S(w_i)\,||^2 \leq 4||w_i||^2$ (inequality (11)), and $||w_i|| \leq M$.

By combining equations (30), (3) and (40), we obtain

$$\mathbb{E}_{Q,S}||Q(S(\mathbf{w})) - \mathbf{w}||^2 \leq \frac{4dM^2}{(2^{m_k} - 1)^2} + (1 - \gamma_k)M^2.$$

3) PROOF OF THEOREM 1

We are interested in the convergence of global iterations, thus, our proof will focus on the case where $t+1 \in R_I$. We have

$$\begin{aligned}||\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*||^2 &= ||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1} + \bar{\mathbf{u}}_{t+1} - \mathbf{w}^*||^2\\ &= \underbrace{||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}||^2}_{U_1} + \underbrace{||\bar{\mathbf{u}}_{t+1} - \mathbf{w}^*||^2}_{U_2}\\ &\quad +2\underbrace{< \bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}, \bar{\mathbf{u}}_{t+1} - \mathbf{w}^* >}_{U_3}\\ &\leq U_1 + U_2 \qquad (41)\end{aligned}$$

We first bound the term U_1. We replace the values of $\bar{\mathbf{w}}_{t+1}$ and $\bar{\mathbf{u}}_{t+1}$ when $t+1 \in R_I$ in U_1 to obtain

$$\begin{aligned}U_1 &= ||\mathbf{w}_{t+1-I} + J - \frac{1}{K}\sum_{k\in\mathcal{S}_{t+1}} \mathbf{v}^k_{t+1}||^2\\ &= ||J - \sum_{k\in\mathcal{S}_{t+1}}\Big(\mathbf{v}^k_{t+1} - \mathbf{w}_{t+1-I}\Big)||^2\\ &= ||J - \boldsymbol{\delta}^k_{t+1}||^2 \qquad (42)\end{aligned}$$

with $J = \frac{1}{K}\sum_{k\in\mathcal{S}_{t+1}} Q^m(S(\boldsymbol{\delta}^k_{t+1}))_{t+1}$ We introduce the expectation over the quantization and the sparsification. We obtain

$$\begin{aligned}\mathbb{E}_{Q,S}||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}|| \leq \frac{1}{K}\sum_{k\in\mathcal{S}_{t+1}} \mathbb{E}_{Q,S}||Q^m\Big(S(\boldsymbol{\delta}^k_{t+1})\Big)_{t+1}\\ -\boldsymbol{\delta}^k_{t+1}||^2 \qquad (43)\end{aligned}$$

By using Lemma 2, we find

$$\begin{aligned}\mathbb{E}_{Q,S}||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}|| &\leq \frac{1}{K}\Omega||\boldsymbol{\delta}^k_{t+1}||^2\\ &= \frac{1}{K}\Omega||\sum_{\tau=t+1-I}^{t}\eta_\tau\nabla F_k\Big(\mathbf{w}^k_\tau, \zeta^k_\tau\Big)||^2\\ &\leq \frac{I}{K}\Omega\sum_{\tau=t+1-I}^{t}\eta^2_\tau||\nabla F_k\Big(\mathbf{w}^k_\tau, \zeta^k_\tau\Big)||^2\end{aligned} \qquad (44)$$

with $\Omega = \sum_{k\in\mathcal{S}_{t+1}}[\frac{4d}{(2^{m_k}-1)^2} + (1 - \gamma_k)]$ We take the expectation over the randomness of the stochastic gradient to get

$$\begin{aligned}\mathbb{E}_{Q,S,G}||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}|| &\leq \frac{I}{K}\Omega\sum_{\tau=t+1-I}^{t}\eta^2_\tau\mathbb{E}_G||\nabla F_k\Big(\mathbf{w}^k_\tau, \zeta^k_\tau\Big)||^2\\ &\leq \frac{I}{K}\Omega\sum_{\tau=t+1-I}^{t}\eta^2_{t+1-I}H^2\\ &= \frac{I^2}{K}\Omega\eta^2_{t+1-I}H^2\\ &\leq \frac{I^2}{K}\Omega\eta^2_t H^2 \qquad (45)\end{aligned}$$

We further take the expectation over the set of selected devices and obtain

$$\mathbb{E}_{Q,S,G,S_t}||\bar{\mathbf{w}}_{t+1} - \bar{\mathbf{u}}_{t+1}|| \leq \frac{\eta^2_t H^2 I^2}{N}\sum_{k=1}^{N}\Big[\frac{4d}{(2^{m_k} - 1)^2} + (1 - \gamma_k)\Big] \qquad (46)$$

To bound U_2, by using Lemmas 3, 4, 5 and 6, we can generalize the reasoning in [28] as

$$\begin{aligned} U_2 &= \mathbb{E}||\bar{\boldsymbol{u}}_{t+1} - \boldsymbol{w}^*|| \\ &\leq \frac{4(N-K)}{K(N-1)}\eta_t^2 I^2 H^2 + (1-\mu\eta_t)\mathbb{E}||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*||^2 \\ &\quad + \frac{\eta_t^2}{N^2}\sum_{k=1}^{N}\sigma_k^2 + 4\eta_t^2(I-1)^2H^2 \\ &\quad + \frac{1}{N}\sum_{k=1}^{N} -\frac{\mu\eta_t d}{(2^{n_k}-1)^2} + \frac{\eta_t^2 d}{(2^{n_k}-1)^2} \end{aligned} \tag{47}$$

By using the fact that $\eta_t \geq \eta_t^2$ we have

$$\mathbb{E}||\bar{\boldsymbol{w}}_{t+1} - \boldsymbol{w}^*|| \leq (1-\mu_t\eta_t)\mathbb{E}||\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*|| + \eta_t^2 D \tag{48}$$

where $D = \frac{4(N-K)}{K(N-1)}I^2H^2 + \sum_{k=1}^{N}\frac{\sigma_k^2}{N^2} + 4(I-1)^2H^2 + \frac{1}{N}\sum_{k=1}^{N}\frac{d}{(2^{n_k}-1)^2}(1-\mu) + \frac{H^2I^2}{N}\sum_{k=1}^{N}[\frac{4d}{(2^{m_k}-1)^2} + (1-\gamma_k)]$

Following the same steps in [27], we show that when $\eta_t = \frac{\beta}{t+\alpha}$, with $\beta \geq \frac{1}{\mu}$, $\alpha \geq 0$, $\eta_1 \leq \min(\frac{1}{\mu}, \frac{1}{4L}) = \frac{1}{4L}$, and $\eta_t \leq 2\eta_{t+I}$

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F\big(\boldsymbol{w}^*\big) \leq \frac{L}{2}\frac{\phi}{T+\alpha} \tag{49}$$

with $\phi = \max(\frac{\beta^2 D}{\beta\mu - 1}, (\alpha+1)\mathbb{E}||\boldsymbol{w}_0 - \boldsymbol{w}^*||$

We can also show, similarly to [27], that

$$\phi \leq \frac{4D}{\mu^2} + (\frac{8L}{\mu} + I)||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2.$$

Finally, we obtain

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F\big(\boldsymbol{w}^*\big) \leq \frac{2\kappa}{T+\alpha}\left[\frac{D}{\mu} + \big(2L + \frac{I\mu}{4}\big)||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2\right] \tag{50}$$

with $\alpha = \max(8\kappa, I)$, $\kappa = \frac{L}{\mu}$.

4) PROOF OF PROPOSITION 1

From (10), and for homogeneous devices, we upper bound γ

$$\gamma \leq \frac{r\tau}{dm} \tag{51}$$

We upper bound (13) by ϵ to obtain

$$\frac{2\kappa}{T+\alpha}\left[\frac{D}{\mu} + \big(2L + \frac{I\mu}{4}\big)||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2\right] \leq \epsilon$$

Then

$$\frac{D}{\mu} + \underbrace{\big(2L + \frac{I\mu}{4}\big)||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2}_{C} \leq \epsilon\frac{T+\alpha}{2\kappa}$$

Since C is a positive value, we upper bound D by

$$D \leq \frac{\epsilon\mu(T+\alpha)}{2\kappa}$$

Since D is the summation of positive values, we can have the following upper bound

$$H^2I^2\left[\frac{4d}{(2^m-1)^2} + (1-\gamma)\right] \leq \frac{\epsilon\mu(T+\alpha)}{2\kappa}$$

Finally, we have the following bound

$$\gamma \geq 1 - \frac{\epsilon\mu(T+\alpha)}{2\kappa H^2I^2} + \frac{4d}{(2^m-1)^2} \tag{52}$$

From (51) and (52), we get the result of Proposition 1.

REFERENCES

[1] S. Arabi, H. El Hammouti, E. Sabir, H. Elbiaze, and M. Sadik, “RAT association for autonomic IoT systems,” *IEEE Netw.*, vol. 33, no. 6, pp. 116–123, Nov./Dec. 2019.

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1–10.

[3] O. Marnissi, H. El Hammouti, and E. Bergou, “Client selection in federated learning based on gradients importance,” in *Proc. AIP Conf. Proc.*, 2024, pp. 1–6.

[4] O. Marnissi, H. El Hammouti, and E. Bergou, “Joint probability selection and power allocation for federated learning,” in *Proc. IEEE Future Netw. World Forum (FNWF)*, 2023, pp. 1–6.

[5] Z. Cai, X. He, J. Sun, and N. Vasconcelos, “Deep learning with low precision by half-wave Gaussian quantization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5918–5926.

[6] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, “Pact: Parameterized clipping activation for quantized neural networks,” 2018, *arXiv:1805.06085*.

[7] P. Micikevicius et al., “Mixed precision training,” 2017, *arXiv:1710.03740*.

[8] H. Liu, F. He, and G. Cao, “Communication-efficient federated learning for heterogeneous edge devices based on adaptive gradient quantization,” in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[9] Y. Gong, L. Liu, M. Yang, and L. Bourdev, “Compressing deep convolutional networks using vector quantization,” 2014, *arXiv:1412.6115*.

[10] Z. Aojun, Y. Anbang, G. Yiwen, X. Lin, and C. Yurong, “Incremental network quantization: Towards lossless CNNs with low-precision weights,” 2017, *arXiv:1702.03044*.

[11] M. Courbariaux, Y. Bengio, and J.-P. David, “BinaryConnect: Training deep neural networks with binary weights during propagations,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[12] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–11.

[13] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, “Binary neural networks: A survey,” *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107281.

[14] N. Takayuki and Y. Ryo, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[15] Y. H. Howard, L. Zuozhu, Q. S. Tony, and H. V. Poor, “Scheduling policies for federated learning in wireless networks,” *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[16] H. T. Nguyen, B. Wei, Z. Albert, N. H. N. Minh, and S. H. Choong, “Federated learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[17] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, “Tackling system and statistical heterogeneity for federated learning with adaptive client sampling,” in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1739–1748.

[18] L. Liu, J. Zhang, S. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[19] M. Kamp et al., “Efficient decentralized deep learning by dynamic model averaging,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2018, pp. 393–409.

[20] M. Zhang et al., “Joint compression and deadline optimization for wireless federated learning,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 7, pp. 7939–7951, Jul. 2024.

[21] H. Xu et al., “GRACE: A compressed communication framework for distributed machine learning,” in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 561–572.

[22] A. Dutta et al., "On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning," in *Proc. AAAI*, 2020, pp. 1–8.

[23] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2017, pp. 1–6.

[24] W. Wen et al., "TernGrad: ternary gradients to reduce communication in distributed deep learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1508–1518.

[25] S. Frank, F. Hao, D. Jasha, L. Gang, and Y. Dong, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1–5.

[26] Y. Ji and L. Chen, "FedQNN: A computation–communication-efficient federated learning framework for IoT with low-bitwidth neural network quantization," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2494–2507, Feb. 2023.

[27] Z. Sihui, S. Cong, and C. Xiang, "Design and analysis of uplink and downlink communications for federated learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 1–48.

[28] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "Green, quantized federated learning over wireless networks: An energy-efficient design," 2022, pp. 1–16.

[29] P. Kairouz et al., "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[30] L. Zhu, H. Lin, Y. Lu, Y. Lin, and S. Han, "Delayed gradient averaging: Tolerate the communication latency in federated learning," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 1–13.

[31] B. Moons, D. Bankam, and M. Verhelst, *Embedded Deep Learning Algorithms, Architectures and Circuits for Always-on Neural Network Processing*. Cham, Switzerland: Springer, 2019.

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–26.

[33] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[34] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "On the tradeoff between energy, precision, and accuracy in federated quantized neural networks," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 2194–2199.

[35] Y. Nesterov, *Lectures on Convex Optimization*, vol. 137. Berlin, Germany: Springer, 2018.

[36] Y. LeCun, C. Cortes, and C. J. C. Burges. "MNIST handwritten digit database." 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist