Entropy-Adaptive Federated Learning with Efficient Bit Allocation over Wireless Channels

Shayan Mohajer Hamidi and Ben Liang
Department of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Emails: s.mohajer@utoronto.ca, liang@ece.utoronto.ca

***Abstract*—A key bottleneck in federated learning (FL) is the high communication cost of transmitting large local model updates. This paper proposes Entropy-Adaptive FL (EA-FL), a novel framework that integrates information-theoretic coding into the FL pipeline to reduce the bit rate of encoded updates. In EA-FL, clients train local models under an entropy constraint on their updates, ensuring they are compressible and require fewer bits when entropy coded. Accounting for wireless channel imperfections, we analyze the statistical behavior of the decoded updates at the server and derive bounds on the first and second derivatives of the EA-FL loss function, establishing its Lipschitz continuity. These results enable a convergence analysis that explicitly captures the effects of quantization error, channel noise, and aggregation. Building on this, we develop an optimization framework for efficient bit allocation across clients under a fixed total quantization budget. Extensive experiments show that (i) EA-FL outperforms state-of-the-art quantized FL methods in rate-accuracy trade-offs, and (ii) the proposed bit allocation scheme significantly improves over uniform allocation under the same bit budget.**

I. Introduction

Federated learning (FL) enables multiple edge devices or clients to collaboratively train a shared model by exchanging local updates instead of raw data [1], with the process orchestrated by a central parameter server (PS). A key bottleneck in FL arises from the need to transmit large volumes of updated model parameters from clients to the PS over uplink channels, which often have limited bandwidth [2], [3]. This issue can be addressed by selecting a subset of users through scheduling techniques [4], or by minimizing the amount of data each client transmits using methods such as quantization [5], [6], sparsification [7], [8], and model pruning [9]. Among these, *quantization* has proven particularly effective, compressing local updates into lower-bit representations to reduce communication (see [5], [6] and references therein).

In practical systems, quantized updates are commonly followed by an entropy coding stage [10], which further reduces communication cost by leveraging the statistical redundancy in the quantized symbols [11]–[13]. However, existing works treat FL training and the quantization-plus-coding process as two separate components. Specifically, they assume that local models are trained using the standard FL loss function, and only then focus on optimizing the quantization and encoding of the resulting updates. This decoupled treatment inevitably leads to sub-optimal performance in terms of the rate–accuracy trade-off.

To address this limitation, we propose a novel FL framework that integrates rate-awareness directly into the local training process, by modifying the local loss functions to account for the bit rate of the encoded updates. Specifically, our approach guides local model training toward producing updates that are inherently more compressible. We begin by noting that when entropy coding is applied, the actual communication cost is approximately equal to the entropy of the probability mass function (PMF) of the quantized updates. Consequently, we explicitly encourage low-entropy update distributions during local training, thereby enabling more efficient compression in the entropy coding stage.

To realize this, we propose Entropy-Adaptive FL (EA-FL), a novel FL framework that incorporates information-theoretic coding principles directly into the training process to significantly reduce the bit rate of encoded local updates. In this framework, before transmission to the PS, each client quantizes its local updates using the stochastic quantization method introduced in [11], [14], [15]. Leveraging the probabilistic structure of this quantization scheme, we compute the entropy of the quantized updates and incorporate it into the training objective. Specifically, each client minimizes a composite loss that combines the conventional local loss with the entropy of its local updates. This encourages updates that are inherently compressible in an information-theoretic sense, i.e., updates that result in a lower bit rate when entropy coding is applied.

The contributions of this paper are summarized as follows:

- We propose EA-FL, a novel FL framework that jointly optimizes model training and communication efficiency by incorporating entropy-based regularization into the local training objective. This approach promotes low-entropy update distributions, enabling more effective compression through entropy coding.
- Within EA-FL, we characterize the statistical properties of the decoded updates at the PS. In addition, we derive the Lipschitz constant of the proposed composite local loss functions, along with bounds on their first- and second-order derivatives. This enables us to analyze the convergence of EA-FL by deriving bounds that explicitly characterize its behavior as a function of quantization error, channel heterogeneity across clients, and model aggregation in FL.
- Based on the derived convergence bounds, we formulate an optimization problem to determine an efficient bit

allocation across clients, subject to a fixed total quantization budget. Since solving this optimization problem is challenging, we propose an efficient sub-optimal solution in semi-closed-form.
- Through extensive experiments on FEMNIST [16], CIFAR-10 [17] and Tiny-ImageNet [18] datasets, we show that EA-FL outperforms existing quantized FL algorithms in terms of achieving a better rate-accuracy trade-off. The method is evaluated under varying levels of heterogeneity, and the impact of key hyperparameters is analyzed.

Notation. Scalars are represented using non-bold letters (e.g., w and W), while vectors are denoted by bold lowercase letters (e.g., $\boldsymbol{w}$). The i-th entry of $\boldsymbol{w}$ is denoted by $\boldsymbol{w}[i]$, and $\boldsymbol{w}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{w}$. The entropy of vector $\boldsymbol{w}$ is denoted by $\mathsf{H}(\boldsymbol{w})$. Sets are denoted using calligraphic letters (e.g., $\mathcal{W}$), and the cardinality of a set $\mathcal{W}$ is represented by $|\mathcal{W}|$. Mathematical expectation is shown by $\mathbb{E}\{\cdot\}$. For a positive integer K, the set $\{1,\ldots,K\}$ is shortened as $[K]$. The real axis is represented by $\mathbb{R}$.

II. Related Works

Communication-efficient methods in FL are often combined with entropy coding techniques to achieve greater compression. In cases where quantization is employed, QSGD [11], for example, encodes the positions of nonzero elements using Elias integer coding [19]. A more recent study [12] employs Elias omega coding to recursively encode binary prefixes, further improving compression efficiency. In a different line of work, [13] introduces a modified Lloyd quantizer [20] that jointly optimizes quantization distortion and bit rate. Sparsification-based methods are also commonly paired with entropy coding to enhance compression efficiency. In [21], [22], local updates are sparsified by selecting entries with large magnitudes, and the positions of nonzero elements are compressed using Golomb coding [10]. Similarly, [23] applies threshold-based sparsification followed by run-length coding to compress the resulting sparse vectors.

A variety of advanced techniques to apply entropy coding in FL have also been proposed in the literature. For instance, [24] proposes a predictive coding scheme where each value is estimated based on previous samples, and the residuals are entropy coded. In [25], the authors propose Fed-CVLC, which dynamically adjusts code lengths based on model update dynamics, demonstrating promising results on public datasets. To address data correlation across clients, [26] introduces content compression coding, which leverages inter-client similarities to reduce redundancy and transmission cost.

It is important to emphasize that all of the aforementioned works treat FL training and the quantization-plus-coding process as two separate components. This decoupled design typically results in suboptimal performance compared with our proposed EA-FL, which directly integrates communication cost into the training objective.

Lastly, it is worth noting that [27], [28] takes a step toward integrating communication efficiency into the training process. However, their focus lies primarily on online optimization by considering the temporal variation during training, so it uses a completely different optimization approach. In addition, they only consider a simplified version of the communication by modeling the temporal variation as the squared difference between successive model vectors, so they never model the actual bit length. Our work instead embeds an explicit per-round entropy penalty, giving a differentiable surrogate for the *exact* code length.

III. System Model

A. FL System

We consider a wireless FL system consisting of a PS and M edge devices. Each device, indexed by $m \in \{1,\ldots,M\}$, holds a private local dataset $\mathcal{D}_m$ of size $|\mathcal{D}_m|$. The collective objective is to train a global model on the PS that generalizes well across the data distributions of all participating devices, without exposing their local datasets. The empirical local training loss function for device m is defined as $f_m(\boldsymbol{w}) \triangleq \frac{1}{|\mathcal{D}_m|}\sum_{\boldsymbol{\xi}\in\mathcal{D}_m} l(\boldsymbol{w};\boldsymbol{\xi})$, where $\boldsymbol{w} \in \mathbb{R}^D$ is the global model parameter vector, $\boldsymbol{\xi}$ is a particular data sample, and $l(\cdot)$ is the sample-wise training loss associated with each data sample. Then, the global training loss function is

$$f(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|}\sum_{m\in[M]} |\mathcal{D}_m|\, f_m(\boldsymbol{w}) \;=\; \sum_{m\in[M]} \alpha_m f_m(\boldsymbol{w}), \quad (1)$$

where $\mathcal{D}$ denotes the global dataset, i.e., $\mathcal{D} = \bigcup_{m=1}^{M}\mathcal{D}_m$, and $\alpha_m = \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$. Here, we employ the standard federated averaging (FedAvg) algorithm [1] for iterative model training in FL, where the PS updates the global model by aggregating gradients computed from the local loss functions across all devices. The objective is to find the optimal global model $\boldsymbol{w}^\star$ that minimizes the global training loss function $f(\boldsymbol{w})$. Each iteration of the algorithm that results in a global model update is referred to as a communication round. The operations performed during communication round t are as follows:

1) **Downlink phase:** The PS broadcasts the current global model parameter vector $\boldsymbol{w}^t$ to all devices.
2) **Client update (mini-batch SGD):** Starting from $\boldsymbol{w}_m^{t,0} = \boldsymbol{w}^t$, device m performs K local steps:

$$\text{Sample } \mathcal{B}_m^{t,\tau} \subset \mathcal{D}_m,\ \ |\mathcal{B}_m^{t,\tau}| = B, \quad (2a)$$

$$g_m^{t,\tau} \triangleq \frac{1}{B}\sum_{\boldsymbol{\xi}\in\mathcal{B}_m^{t,\tau}} \nabla l\big(\boldsymbol{w}_m^{t,\tau-1};\boldsymbol{\xi}\big)\,, \quad (2b)$$

$$\boldsymbol{w}_m^{t,\tau} = \boldsymbol{w}_m^{t,\tau-1} - \eta\, g_m^{t,\tau}, \quad \tau = 1,\ldots,K, \quad (2c)$$

$$\boldsymbol{w}_m^{t+1} = \boldsymbol{w}_m^{t,K}. \quad (2d)$$

   Here $\eta > 0$ is the local learning rate and B is the mini-batch size (and $B=1$ recovers single-sample SGD).
3) **Uplink phase:** Each client transmits the model update $\boldsymbol{w}_m^{t+1} - \boldsymbol{w}^t$ to the PS via uplink wireless channels.
4) **Model aggregation:** The PS forms the new global model

$$\boldsymbol{w}^{t+1} \;=\; \boldsymbol{w}^t + \sum_{m\in[M]} \alpha_m \,\big(\boldsymbol{w}_m^{t+1} - \boldsymbol{w}^t\big)\,. \quad (3)$$

The FL process proceeds iteratively and continues for a total of T communication rounds before completing the training.

B. Quantized FL Mechanism

In a quantized FL setting, each client transmits a quantized version of its local model update, denoted by $\mathsf{Q}(\boldsymbol{w}_m^{t+1} - \boldsymbol{w}^t)$, where $\mathsf{Q}(\cdot)$ represents the quantization function. The PS then computes the global model update by aggregating these quantized updates as $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \sum_{m\in[M]} \alpha_m \, \mathsf{Q}\left(\boldsymbol{w}_m^{t+1} - \boldsymbol{w}^t\right)$.

In this work, we focus on the quantized FL setup under heterogeneous client channel conditions. We first present the employed quantization scheme in Section III-B1, followed by a detailed discussion of the channel modeling for individual clients in Section III-B2.

1) Quantization Operation: For the quantization operation, we adopt the same parametrizable lossy quantizer introduced in [11], [14], [15]. For notational convenience, we define the local model update at client m as $\boldsymbol{\delta}_m^t \triangleq \boldsymbol{w}_m^{t+1} - \boldsymbol{w}^t$ and $\boldsymbol{\delta}_m^{t,\tau} \triangleq \boldsymbol{w}_m^{t,\tau-1} - \boldsymbol{w}^t$ throughout the remainder of this work.

First, the range of the D-dimensional vector $\boldsymbol{\delta}_m^t$ is quantified by $\delta_m^{t,\max} = \max\{|\boldsymbol{\delta}_m^t[d]| \;\big|\; d = 1, 2, \cdots, D\}$, where $|\cdot|$ denotes the absolute value operator. We assume that client m allocates q_m^t bits to quantize each of the D components of $\boldsymbol{\delta}_m^t$. Accordingly, the range $\delta_m^{t,\max}$ is partitioned into $2^{q_m^t} - 1$ uniform intervals, with quantization levels (or reconstruction points) given by $c_u = \frac{u\,\delta_m^{t,\max}}{2^{q_m^t}-1}$, $u = 0, 1, \ldots, 2^{q_m^t} - 1$. We also denote the set of all quantization levels as $\mathcal{C} = \{c_u\}$.

Now, suppose $|\boldsymbol{\delta}_m^t[d]| \in [c_u, c_{u+1})$. Then, the quantized value of $\boldsymbol{\delta}_m^t[d]$ is given by

$$\mathsf{Q}(\boldsymbol{\delta}_m^t[d]) = \begin{cases} \mathsf{sgn}(\boldsymbol{\delta}_m^t[d])\, c_u, & \text{w.p.} \quad \frac{c_{u+1} - |\boldsymbol{\delta}_m^t[d]|}{c_{u+1} - c_u}, \\ \mathsf{sgn}(\boldsymbol{\delta}_m^t[d])\, c_{u+1}, & \text{w.p.} \quad \frac{|\boldsymbol{\delta}_m^t[d]| - c_u}{c_{u+1} - c_u}, \end{cases} \tag{4}$$

where $\mathrm{sgn}(\cdot)$ denotes the sign function, and w.p. stands for "with probability". We also denote the aggregated quantized values across all D dimensions of $\boldsymbol{\delta}_m^t$ as $\mathsf{Q}(\boldsymbol{\delta}_m^t) = [\mathsf{Q}(\boldsymbol{\delta}_m^t[1]), \cdots, \mathsf{Q}(\boldsymbol{\delta}_m^t[D])]$.

2) Transmission Scheme and Channel Model: Before transmitting $\mathsf{Q}(\boldsymbol{\delta}_m^t[d])$ to the PS, client m encodes it using an entropy coding technique (such as Huffman coding) for additional compression, resulting in $\mathsf{Enc}(\mathsf{Q}(\boldsymbol{\delta}_m^t[d]))$. The encoded data $\mathsf{Enc}(\mathsf{Q}(\boldsymbol{\delta}_m^t[d]))$ is then transmitted over a wireless noisy channel. Similarly to [29], we model the wireless link as a binary symmetric channel (BSC) with a bit-flip probability p_m^t. While in practice, entropy-coded data may suffer from synchronization loss due to single-bit errors, it is common to absorb this effect into an overall bit error probability [30].

The PS decodes the received, noisy version of $\mathsf{Enc}(\mathsf{Q}(\boldsymbol{\delta}_m^t[d]))$ to recover an estimate denoted by $\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t[d])$. It then computes the global model update by aggregating these quantized updates as

$$\boldsymbol{w}^{t+1} \;=\; \boldsymbol{w}^t + \sum_{m\in[M]} \alpha_m \, \widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t). \tag{5}$$

According to the described quantization scheme, the transmitted representation of $\mathsf{Q}(\boldsymbol{\delta}_m^t)$ consists of three components: the quantization range, the signs of each element, and the indices corresponding to quantization levels. Accordingly, the total bit length required to encode $\mathsf{Q}(\boldsymbol{\delta}_m^t)$ is given by

$$\ell_m^t = b_{\boldsymbol{\delta}_m^t} + D + b_{\mathrm{f}}, \tag{6}$$

where $b_{\boldsymbol{\delta}_m^t}$ denotes the average number of bits required to encode the quantization indices using an entropy coding technique, and b_{f} is the number of bits used to represent the dynamic range, typically stored in standard floating-point format (e.g., $b_{\mathrm{f}} = 32$). If the dimension D is large enough, which is the case for deep learning models, the average code length per symbol using a prefix code will converge to the entropy of the source [31]. Therefore, $b_{\boldsymbol{\delta}_m^t} \approx \mathsf{H}(\boldsymbol{\delta}_m^t)$, where $\mathsf{H}(\cdot)$ denotes the entropy function, and for notational simplicity, we denote $\mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{\delta}_m^t\right)\right)$ by $\mathsf{H}\left(\boldsymbol{\delta}_m^t\right)$ hereafter. Thus ℓ_m^t can be approximated as

$$\ell_m^t = \mathsf{H}(\boldsymbol{\delta}_m^t) + D + b_{\mathrm{f}}. \tag{7}$$

IV. Entropy-Adaptive Federated Learning

In this section, we present the proposed EA-FL framework. As outlined in Section I, the objective of EA-FL is to reduce the actual uplink communication cost, characterized by (7). To achieve this, EA-FL enforces an entropy constraint on the local updates during training as elaborated in Section IV-A.

A. Local Loss Functions in EA-FL

In EA-FL, each client minimizes its local loss function subject to a bound on the entropy of the quantized update. The optimization problem is formulated as

$$\min_{\boldsymbol{w}_m} \quad f_m(\boldsymbol{w}_m) \tag{8}$$

$$\text{subject to} \quad \mathsf{H}\left(\boldsymbol{w}_m - \boldsymbol{w}^t\right) \le R^{\mathrm{target}}, \tag{9}$$

where $R^{\mathrm{target}} > 0$ is a predefined threshold.

Alternatively, the constrained optimization problem in (8) can be reformulated as an unconstrained problem using the method of Lagrange multipliers:

$$\min_{\boldsymbol{w}_m} \quad f_m(\boldsymbol{w}_m) + \lambda\, \mathsf{H}\left(\boldsymbol{w}_m - \boldsymbol{w}^t\right), \tag{10}$$

where $\lambda > 0$ is a regularization parameter that controls the trade-off between minimizing the local loss and the entropy of the quantized update. For the ease of notation, we define

$$\mathcal{L}_m(\boldsymbol{w}_m) \triangleq f_m(\boldsymbol{w}_m) + \lambda\, \mathsf{H}\left(\boldsymbol{w}_m - \boldsymbol{w}^t\right) \tag{11}$$

To solve (10), each client performs K local SGD steps, following the standard FL update pattern described in (2). However, the gradient update rule is modified to incorporate the entropy regularization term, resulting in the following update equation:

$$\boldsymbol{w}_m^{t,\tau} = \boldsymbol{w}_m^{t,\tau-1} - \eta \left[\nabla \mathcal{L}_m(\boldsymbol{w}_m^{t,\tau-1})\right]. \tag{12}$$

To execute the update rule in (12), it is necessary to compute $\mathsf{H}\left(\boldsymbol{\delta}_m^{t,\tau}\right)$ as well as its corresponding gradient. We provide a detailed explanation of this computation in (IV-B).

B. Entropy Computation of Quantized Updates

We observe that the probabilistic quantization rule defined in (4) induces a PMF, which maps each input value $\boldsymbol{\delta}_m^{t,\tau}[d]$ to a quantized level $\hat{\delta} \in \mathcal{C}$. Specifically, for any given u and $|\boldsymbol{\delta}_m^{t,\tau}[d]| \in [c_u, c_{u+1})$, the conditional PMF is expressed as

$$P\left(\hat{\delta} \mid \boldsymbol{\delta}_m^{t,\tau}[d]\right) = \begin{cases} \frac{c_{u+1}-|\boldsymbol{\delta}_m^{t,\tau}[d]|}{c_{u+1}-c_u}, & \hat{\delta} = \mathsf{sgn}(\boldsymbol{\delta}_m^{t,\tau}[d])c_u, \\ \frac{|\boldsymbol{\delta}_m^{t,\tau}[d]|-c_u}{c_{u+1}-c_u}, & \hat{\delta} = \mathsf{sgn}(\boldsymbol{\delta}_m^{t,\tau}[d])c_{u+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Note that the dimensionality of $P\left(\hat{\delta} \mid \boldsymbol{\delta}_m^{t,\tau}[d]\right)$ is $2^{q_m^t} \times 2$, where $2^{q_m^t}$ corresponds to the number of quantization levels, and the factor of 2 accounts for the sign introduced by the sgn operation.

Now, consider selecting an entry δ uniformly at random from the vector $\boldsymbol{\delta}_m^{t,\tau}$, and let $\hat{\delta} \in \mathcal{C}$ denote its corresponding quantized value obtained via (4). The marginal PMF of $\hat{\delta}$ is then given by

$$P(\hat{\delta}) = \frac{1}{D} \sum_{d\in[D]} P(\hat{\delta} \mid \boldsymbol{\delta}_m^{t,\tau}[d]), \qquad \forall \hat{\delta} \in \mathcal{C}. \tag{13}$$

If entropy coding (e.g., Huffman coding) is used to encode the quantized update vector, then the average number of bits required per weight is approximately equal to the Shannon entropy of $\hat{\delta}$, given by $\mathsf{H}(\hat{\delta}) = \mathsf{H}\big(\frac{1}{D}\sum_{d\in[D]} P(\cdot\,|\boldsymbol{\delta}_m^{t,\tau}[d])\big)$. Accordingly, the total number of bits required to represent $\boldsymbol{\delta}_m^{t,\tau}$ is given by

$$\mathsf{H}\big(\boldsymbol{\delta}_m^{t,\tau}\big) = D\,\mathsf{H}\big(\frac{1}{D}\sum_{d\in[D]} P(\cdot\,|\boldsymbol{\delta}_m^{t,\tau}[d])\big). \tag{14}$$

Now, that $\mathsf{H}\big(\boldsymbol{\delta}_m^{t,\tau}\big)$ is computed, as per (12) during communication round t, and local iteration τ, the local client m updates its model as

$$\boldsymbol{w}_m^{t,\tau} = \boldsymbol{w}_m^{t,\tau-1} - \eta\left[\nabla f_m(\boldsymbol{w}_m^{t,\tau-1}) + \lambda \nabla \mathsf{H}\big(\boldsymbol{\delta}_m^{t,\tau-1}\big)\right]. \tag{15}$$

Remark 1. *Note that the entropy function in* (14) *is differentiable with respect to its parameters, $\boldsymbol{\delta}_m^{t,\tau}[d]$, $\forall d \in [D]$. The only potential point of non-differentiability arises when $\boldsymbol{\delta}_m^{t,\tau}[d]$ is exactly zero, due to the absolute value function in* (4)*. However, this event has zero probability, and does not occur in practice.*

Remark 2. *Practical communication systems, which any FL method requires, already perform entropy counting and coding. EA-FL simply reuses these steps. For entropy-gradient computation, EA-FL reuses the same probability histogram built for entropy estimation, requiring only a few scalar multiplications. This leads to negligible computational cost.*

C. Convergence Analysis

In this section, we present theoretical results on the convergence behavior of EA-FL. To facilitate the analysis, we begin by stating two standard assumptions commonly adopted in the literature (e.g., [11], [15]).

Assumption 1: The local loss function $f_m(\cdot)$ is L_f-smooth, i.e., there exists a constant $L_f > 0$ such that

$$\|\nabla f_m(\boldsymbol{w}) - \nabla f_m(\boldsymbol{w}')\| \le L_f \|\boldsymbol{w} - \boldsymbol{w}'\|, \;\; \forall\, \boldsymbol{w}, \boldsymbol{w}' \in \mathbb{R}^D. \tag{16}$$

Assumption 2: The stochastic gradient of $f_m(\cdot)$ is unbiased $\mathbb{E}[\nabla f_m(\boldsymbol{w}, \boldsymbol{\xi})] = \nabla f_m(\boldsymbol{w})$, and its second moment is bounded as $\mathbb{E}\|\nabla f_m(\boldsymbol{w}, \boldsymbol{\xi})\|^2 \le \sigma_f^2$ $\forall m \in [M]$.

Note that $\mathsf{H}(\cdot)$ is generally a non-convex function, and therefore we cannot assume convexity for the composite objective $\mathcal{L}_m(\cdot)$. Non-convex loss functions are commonly encountered in training deep neural networks. In such settings, it is well-known that SGD may converge to a local minimum or a saddle point. In the sequel, we prove that EA-FL exhibits the same behavior.

We begin by characterizing the statistical behavior of the entropy regularization term. The following lemma establishes bounds on its gradient and shows that the function is smooth.

Lemma 1. *Suppose that for all $\hat{\delta} \in \mathcal{C}$, the marginal PMF of $\boldsymbol{\delta}_m^{t,\tau}$ satisfies $P(\hat{\delta}) \ge \mathsf{P}_0$. Then, the entropy function $\mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})$ satisfies the following properties:*

- *Its gradient is bounded as $\left\|\frac{\partial \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial \boldsymbol{\delta}_m^{t,\tau}}\right\|^2 \le \frac{(2^{q_m^t}-1)^2}{D(\delta_m^{t,\max})^2}|\ln \mathsf{P}_0|^2$.*
- *It is $L_{h,m}$-smooth, where the smoothness constant is given by $L_{h,m} = \frac{2(2^{q_m^t}-1)^2}{\mathsf{P}_0 D^{3/2}(\delta_m^{t,\max})^2}$.*

Proof. Please refer to Appendix A for the proof. □

Now, let $\kappa \triangleq \max_m \frac{2^{q_m^t}-1}{\delta_m^{t,\max}}$. Then, for all $m \in [M]$, the entropy function $\mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})$ satisfies $\left\|\frac{\partial \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial \boldsymbol{\delta}_m^{t,\tau}}\right\|^2 \le \frac{\kappa^2}{D}|\ln \mathsf{P}_0|^2$. Let $L_h \triangleq \frac{2\kappa^2}{\mathsf{P}_0 D^{3/2}}$. The following result follows directly from the linearity of Lipschitz constants under addition of smooth functions.

Corollary 1. *$\mathcal{L}_m(\cdot)$ is L-smooth, where $L = L_f + \lambda L_h$.*

Next, we characterize the statistical properties of the decoded local updates received at the PS, denoted by $\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)$. The following lemma states that $\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)$ is an unbiased estimate of the true local update, and establishes an upper bound on the variance induced by both the quantization process and the transmission over a noisy communication channel.

Lemma 2. *The quantized local update $\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)$ is an unbiased estimator of the true local model update $\boldsymbol{\delta}_m^t$, i.e., $\mathbb{E}[\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)] = \boldsymbol{\delta}_m^t$, and the variance is bounded by*

$$\mathbb{E}\left[\left\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\right\|^2\right] \le \frac{D(\delta_m^{t,\max})^2}{4(2^{q_m^t}-1)^2} + \left[1-(1-p_m^t)^{q_m^t}\right]\frac{D(\delta_m^{t,\max})^2}{(2^{q_m^t}-1)^2}\frac{2^{q_m^t}+1}{3}. \tag{17}$$

Proof. Please refer to Appendix B. □

Corollary 2. *From Lemma 2, it directly follows that*

$$\mathbb{E}\left[\left\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)-\boldsymbol{\delta}_m^t\right\|^2\right]\leq \overline{Q}_m^t\|\boldsymbol{\delta}_m^t\|^2, \tag{18}$$

where $\overline{Q}_m^t=\frac{D}{4(2^{q_m^t}-1)^2}+\left[1-(1-p_m^t)^{q_m^t}\right]\frac{D}{3}\frac{2^{q_m^t}+1}{(2^{q_m^t}-1)^2}$.

Define $\mathcal{L}(\boldsymbol{w})\triangleq\sum_{m\in[M]}\alpha_m\mathcal{L}_m(\boldsymbol{w}_m)$. Denote by $\boldsymbol{w}^\star$ the global minimizer of $\mathcal{L}$, and by $\mathcal{L}^\star$ the corresponding optimal objective value, i.e., $\mathcal{L}^\star=\mathcal{L}(\boldsymbol{w}^\star)$. The following theorem establishes the convergence bound for EA-FL when $K=1$.

Theorem 1 (for $K=1$)**.** *Suppose Assumptions 1 and 2 hold. Then, the convergence of EA-FL satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2\leq\frac{\mathcal{L}\left(\boldsymbol{w}^0\right)-\mathcal{L}^\star}{T\eta_t}$$
$$+\frac{1}{T}\sum_{t=0}^{T-1}LM\eta_t\left(\sigma_f^2+\lambda^2\frac{\kappa^2}{D}|\ln\mathsf{P}_0|^2\right)\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right).$$

Proof. To facilitate the proof of our results, we introduce an auxiliary variable $\tilde{\boldsymbol{w}}^{t+1}=\sum_{m\in[M]}\alpha_m\boldsymbol{w}_m^{t+1}$, with $\boldsymbol{w}^{t+1}$ obtained from (5). We first establish the following lemmas whose proofs are deferred to Appendix C.

Lemma 3. $\mathbb{E}\left[\boldsymbol{w}^{t+1}\right]=\mathbb{E}\left[\tilde{\boldsymbol{w}}^{t+1}\right]$.

Lemma 4. *With Assumption 1, we have* $\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^{t+1}\right)\leq\mathbb{E}\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right)+\frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\|^2$.

Lemma 5. *The stochastic gradient of $\mathcal{L}_m(\cdot)$ is unbiased $\mathbb{E}[\nabla\mathcal{L}_m(\boldsymbol{w},\boldsymbol{\xi})]=\nabla\mathcal{L}_m(\boldsymbol{w})$, and the second moment of the stochastic gradient is bounded as $\mathbb{E}\|\nabla\mathcal{L}_m(\boldsymbol{w},\boldsymbol{\xi})\|^2\leq\sigma^2$ for all $m\in[M]$, where $\sigma^2=2\sigma_f^2+2\lambda^2\frac{\kappa^2}{D}|\ln\mathsf{P}_0|^2$.*

Lemma 6. *With Assumptions 1 and 2, we have* $\mathbb{E}\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right)\leq\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right)-\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2+\frac{ML\eta_t^2\sigma^2}{2}\sum_m\alpha_m^2$.

Lemma 7. $\mathbb{E}\left[\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right]^2\leq M\eta_t^2\sigma^2\sum_m\alpha_m^2\overline{Q}_m^t$.

Now, we begin proving the theorem:

$$\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^{t+1}\right)\leq\mathbb{E}\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right)+\frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\|^2 \tag{19}$$
$$\leq\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right)-\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2+\frac{ML\eta_t^2\sigma^2}{2}\sum_m\alpha_m^2$$
$$+\frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\|^2 \tag{20}$$
$$\leq\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right)-\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2+\frac{ML\eta_t^2\sigma^2}{2}\sum_m\alpha_m^2$$
$$+\frac{L}{2}M\eta_t^2\sigma^2\sum_m\alpha_m^2\overline{Q}_m^t \tag{21}$$
$$=\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right)-\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2$$
$$+\frac{L}{2}M\eta_t^2\sigma^2\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right), \tag{22}$$

where (19) is due to Lemma 4, (20) is due to Lemma 6, (21) is due to Lemma 7.

We now sum (22) over $t=0,\cdots,T-1$, and rearrange the terms, which yields

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2\leq\mathcal{L}\left(\boldsymbol{w}^0\right)-\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) \tag{23}$$
$$+\sum_{t=0}^{T-1}\frac{L}{2}M\eta_t^2\sigma^2\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right) \tag{24}$$
$$\leq\mathcal{L}\left(\boldsymbol{w}^0\right)-\mathcal{L}^\star+\sum_{t=0}^{T-1}\frac{L}{2}M\eta_t^2\sigma^2\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right). \tag{25}$$

Dividing T on both sides of (25) leads to

$$\frac{1}{T}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2$$
$$\leq\frac{\mathcal{L}\left(\boldsymbol{w}^0\right)-\mathcal{L}^\star}{T}+\frac{1}{T}\sum_{t=0}^{T-1}\frac{L}{2}M\eta_t^2\sigma^2\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right),$$

which concludes the theorem. □

Remark 3. *For a constant stepsize $\eta=\mathcal{O}(1/\sqrt{T})$, the first term in Theorem 1 decays as $\mathcal{O}(1/T)$ while the second term is $\mathcal{O}(1/\sqrt{T})$; the dominant term is therefore $\mathcal{O}(1/\sqrt{T})$. This matches the standard rate of distributed SGD without quantization [32]. When the stochastic-gradient variance $(\sigma_f=0)$ and the quantization error vanish, the second term disappears and the bound improves to $\mathcal{O}(1/T)$.*

Theorem 2 (for $K>1$)**.** *Suppose Assumptions 1 and 2 hold. Then, the convergence of EA-FL satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2\leq\frac{2(\mathcal{L}(\boldsymbol{w}^0)-\mathcal{L}^\star)}{T\eta_tK}$$
$$+\left[\frac{1}{T}\sum_{t=0}^{T-1}LM\eta_t\left(\sigma_f^2+\lambda^2\frac{\kappa^2}{D}|\ln\mathsf{P}_0|^2\right)\right]\times$$
$$\left[\frac{1}{3}L\eta_t(2K^2-3K+1)\sum_m\alpha_m^2+2K\sum_m\alpha_m^2\left(1+\overline{Q}_m^t\right)\right]$$

Proof. To prove Theorem 2, we extend Lemmas 6 and 7 to the case where $K>1$, accounting for the drift across local updates. Applying the same bounding techniques as in (19) for Theorem 1 then yields the desired result. The full proof is deferred to the extended version of the paper. □

V. Quantization Bits Allocation

In this section, we develop a bit allocation strategy for quantization by minimizing the upper bound on the convergence rate, subject to a global quantization resource budget. To this end, from Theorems 1 and 2, we observe that the convergence bound can be improved by minimizing the term $\sum_{m\in[M]}\alpha_m^2\left(1+\overline{Q}_m^t\right)$, where $\overline{Q}_m^t$ is defined below (18).

This motivates allocating the quantization levels $\{q_m^t\}_{m\in[M]}$ in a way that minimizes this expression, thereby tightening the

convergence bound. Accordingly, we formulate the following optimization problem:

$$\min_{q_m^t \in \mathbb{N}} \sum_{m\in[M]} \alpha_m^2 \overline{Q}_m^t \tag{26a}$$

$$\text{s.t.} \quad \sum_{m\in[M]} q_m^t \le R, \tag{26b}$$

where we assume that the total number of quantization bits for allocation among the clients in each round is bounded by R.

Toward a tractable solution to this problem, we begin by relaxing the integer constraint $q_m^t \in \mathbb{N}$ to $q_m^t \ge 0$. Under this relaxation, it is straightforward to verify that $\overline{Q}_m^t$ is a monotonically decreasing function of q_m^t. Thus, the optimal solution is attained when the inequality constraint is active, i.e., $\sum_{m=1}^M q_m^t = R$. We proceed to solve this optimization problem using the Karush–Kuhn–Tucker (KKT) conditions.

The Lagrangian for the relaxed problem is given by $\sum_{m=1}^M \alpha_m^2 \overline{Q}_m^t + \lambda \left(\sum_{m=1}^M q_m^t - R\right)$. A KKT point $\{q_m^\star\}$ satisfies the first-order optimality conditions:

$$\frac{\partial}{\partial q_m} \alpha_m^2 \overline{Q}_m^t(q_m^\star) \;=\; -\lambda, \quad \forall\, m, \qquad \sum_{m=1}^M q_m^\star = R.$$

Since in practical systems p_m^t is small and q_m^t is large, we apply the approximations $(1-p_m^t)^{\,q_m^t} \approx 1 - p_m^t\, q_m^t$ and $2^{q_m^t} - 1 \approx 2^{q_m^t}$. Substituting into the expression for $\overline{Q}_m^t$, we get

$$\alpha_m^2 \overline{Q}_m^t \;\approx\; \alpha_m^2\, D\Big[\tfrac{1}{4\,2^{2q_m^t}} \;+\; \tfrac{p_m^t\, q_m^t}{3\,2^{q_m^t}}\Big].$$

Taking the derivative with respect to q_m^t, we obtain $\frac{\partial \alpha_m^2 \overline{Q}_m^t}{\partial q_m^t} = \alpha_m^2\, D\Big[-\frac{\ln(2)}{2}\, 2^{-2q_m^t} \;+\; \frac{p_m^t}{3}\left(1 - q_m^t\, \ln(2)\right) 2^{-q_m^t}\Big]$. Setting this equal to $-\lambda$ from the KKT condition gives the equation

$$-\tfrac{\ln(2)}{2}\, 2^{-2q_m^t} \;+\; \tfrac{p_m^t}{3}\left(1 - q_m^t\, \ln(2)\right) 2^{-q_m^t} \;=\; -\,\frac{\lambda}{\alpha_m^2\, D}.$$

Now, let $x_m = 2^{-q_m^t}$. The stationarity condition becomes

$$\tfrac{p_m^t}{3}\, x_m\, \big(1 + \ln(x_m)\big) \;-\; \tfrac{\ln(2)}{2}\, x_m^2 \;=\; -\,\frac{\lambda}{\alpha_m^2\, D}.$$

Solving numerically (or, if the x_m^2 term is small, via the Lambert W function) gives each $x_m(\lambda)$ and thus $q_m^t(\lambda) = -\ln_2\big(x_m(\lambda)\big)$.

A one-dimensional search, e.g. bisection on λ, finds λ^* satisfying $\sum_{m=1}^M q_m(\lambda^*) = R$. Lastly, we map each real solution $q_m(\lambda^*)$ to an integer $\widehat{q}_m^t \in \mathbb{N}$ so that $\sum_{m=1}^M \widehat{q}_m^t = R$. This final discrete allocation $\{\widehat{q}_m^t\}$ is an approximate solution to optimization problem (26).

VI. Experimental Results

In this section, we evaluate the performance of EA-FL through numerical experiments. Throughout this section, we use the following three datasets with the corresponding experimental setups. The code for the project is available at https://anonymous.4open.science/r/INFOCOM2026-7EC2.

• **CIFAR-10:** We partition the CIFAR-10 dataset across $M = 10$ clients using Dirichlet allocation with parameter $\beta = 0.5$. A *ResNet-18* model is trained over $T = 100$ communication rounds, with each client performing $K = 5$ local epochs. The batch size is 64, and the learning rate is fixed at $\eta_t = 0.01$.

• **FEMNIST:** A CNN architecture is employed with two convolutional layers followed by two fully connected layers. The batch size is 32. At each communication round, $M = 500$ clients are randomly sampled from the total pool and perform $K = 5$ local training epochs using their local data.

• **TinyImageNet:** We use ResNet-50 and a batch size of 32 and a learning rate of 0.01. The dataset is divided into $M = 5$ clients with 322, 360, 343, 466, and 509 images, respectively. Each client performs $K = 10$ local epochs.

A. Rate-Accuracy Trade-Off

In this subsection, we focus exclusively on evaluating the effectiveness of the EA-FL's loss function by comparing its rate–accuracy trade-off against the following benchmarks: (i) QSGD [11], (ii) FedPAQ [33], (iii) UVeQFed [34], and (iv) RC-FED [13]. For fair comparison, we apply Huffman coding to compress the quantized gradients in all methods.

For QSGD, FedPAQ, and UVeQFed, we evaluate two fixed quantization levels: $q_m = \{3, 6\}$. For RC-FED, we vary the regularization parameter $\lambda = \{0.2, 0.4, 0.6, 0.8, 1\}$, producing a trade-off curve. Similarly, for EA-FL, we sweep $\lambda = \{0.01, 0.02, 0.03, 0.04, 0.05\}$ to obtain its corresponding trade-off curve using a fixed quantization level of $q_m = 6$. Note that although the quantization level in EA-FL is fixed at $q_m = 6$, larger values of λ induce lower-entropy updates, effectively reducing the communication rate below that of standard 6-bit quantization. To isolate the impact of the loss design, in this subsection, we assume ideal (noise-free) wireless channels and assign the same number of quantization bits to all clients. We note that for noisy channels in practice, error control coding may be employed in addition to quantization and source coding, and our EA-FL design remain applicable.

We plot the test accuracy against the total communication cost incurred during training. The results for all three datasets are presented in Figure 1. As observed, EA-FL consistently achieves superior rate–accuracy trade-off compared with the benchmark methods. In particular, for a fixed target accuracy, EA-FL requires significantly lower communication cost.

B. Evolution of Distribution of Local Updates

Since the entropy term $\mathsf{H}(\boldsymbol{\delta})$ in the optimization objective (12) encourages structure in the local updates, it is instructive to examine how EA-FL alters their distribution to favor higher compressibility. Figure 2 shows how the local update distribution evolves for a random client in the CIFAR-10 setup.

In this experiment, we set $\lambda = 0.03$ in EA-FL and track the distribution of local updates for a randomly selected client across local epochs $\tau \in \{1, 2, \ldots, 5\}$. As shown, the distribution progressively becomes more concentrated around zero, resulting in a notable reduction in entropy. This entropy reduction translates to more compressibility, thereby enabling a higher compression ratio when applying entropy coding.

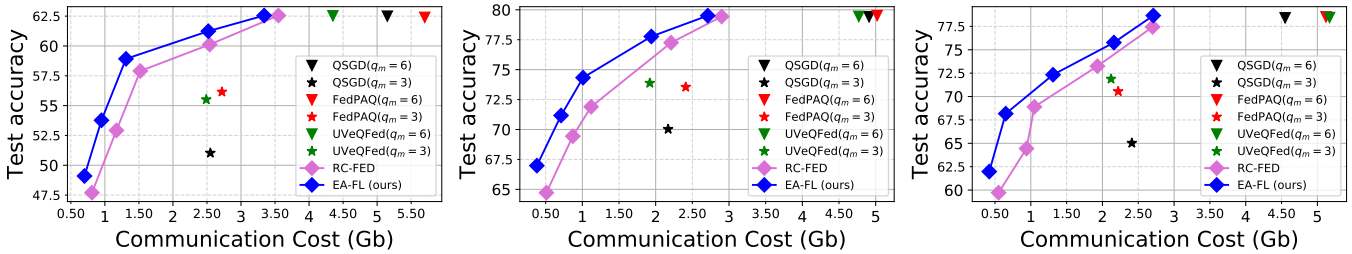

Fig. 1. Test accuracy vs. communication cost for EA-FL and baseline methods on (left) CIFAR-10, (middle) FEMNIST, and (right) Tiny-ImageNet.

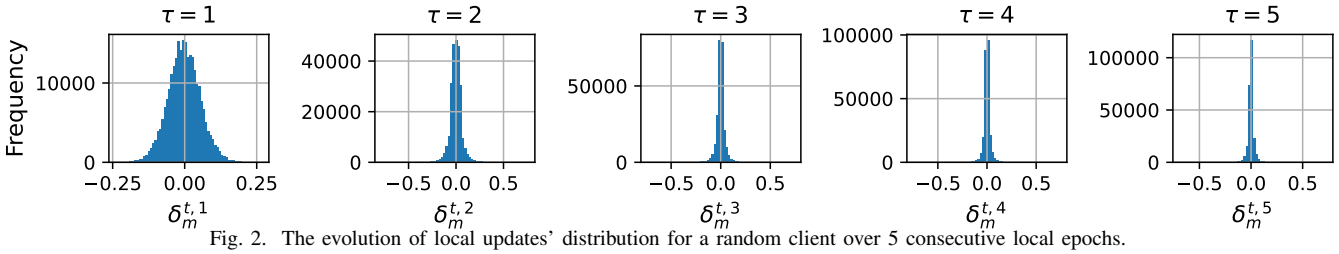

Fig. 2. The evolution of local updates' distribution for a random client over 5 consecutive local epochs.

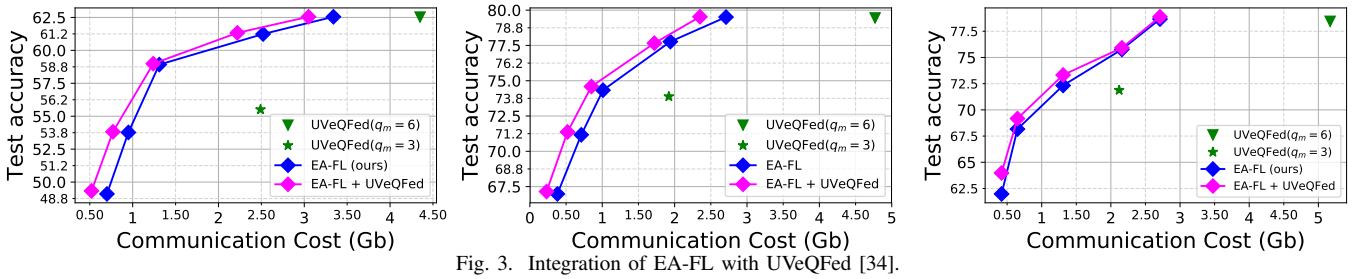

Fig. 3. Integration of EA-FL with UVeQFed [34].

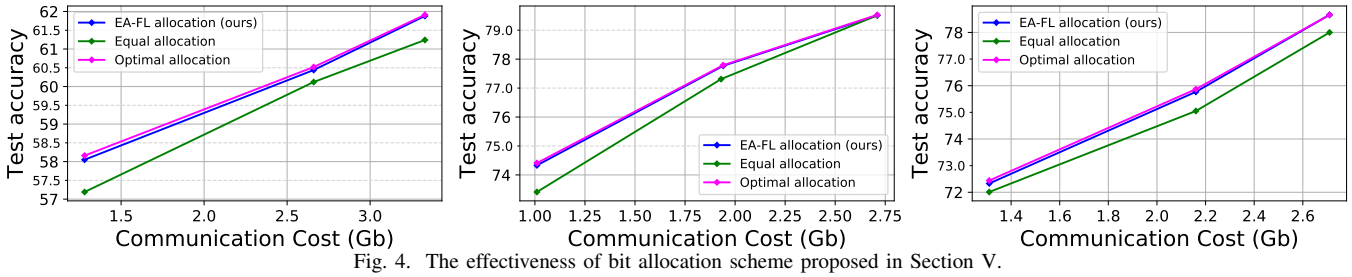

Fig. 4. The effectiveness of bit allocation scheme proposed in Section V.

C. Integrating EA-FL with Other Methods

As discussed in Section I, the key distinction between EA-FL and other encoding methods used in FL lies in the training stage: EA-FL modifies the local training objective to favor compressibility, whereas other methods assume the local models are already trained and focus solely on designing efficient quantization and encoding schemes. In this sense, EA-FL is orthogonal to existing methods and can be integrated with them to further enhance performance.

To demonstrate this, we integrate EA-FL with UVeQFed [34], which applies dithered lattice quantization. The method normalizes model updates, partitions them into sub-vectors, applies dithering before quantization, and uses entropy coding for further compression. The results are presented in Figure 3. As shown, integrating EA-FL with UVeQFed leads to better rate–accuracy trade-off, thanks to the enhanced quantization method, highlighting the orthogonality of the two approaches.

D. Evaluation of Bit Allocation Scheme

In this subsection, we aim to test the effectiveness of our proposed bit allocation scheme in Section V. For the CIFAR-10 setup, we simulate heterogeneous uplink channels by assigning clients to one of 10 channel types, each characterized by a distinct noise standard deviation from the set $\{0.1i : i \in [10]\}$, with an equal number of clients per type. The total quantization bit budget is set to 24 bits. For the FEMNIST setup, we adopt a similar noise model. Specifically, the 500 active clients are partitioned into 10 groups of 50 clients each, with each group assigned a different noise standard deviation from the same set. The total quantization budget is set to 1500 bits. For the Tiny-ImageNet setup, for each of the 5 clients we assign noise standard deviation from the set $\{0.2i : i \in [10]\}$. The total quantization budget is set to 15 bits.

We compare the following three bit allocation strategies: (i) **EA-FL allocation:** Solving the optimization problem in (26) using the method proposed in Section V.

(*ii*) **Equal allocation:** A baseline strategy in which all clients are assigned the same number of quantization bits.
(*iii*) **Optimal allocation:** A heuristic approach that employs a genetic algorithm to obtain a near-optimal solution to the bit allocation problem in (26).

The results are presented in Figure 4. As observed, the EA-FL allocation strategy outperforms the Equal allocation baseline and achieves performance comparable to the Optimal allocation. It is worth noting that, while the genetic algorithm provides near-optimal allocation, it is significantly more computationally intensive. In contrast, EA-FL allocation offers an efficient and scalable alternative with near-optimal performance.

VII. Conclusion

In this paper, we propose EA-FL, a novel federated learning framework that incorporates information-theoretic coding principles into the training process to reduce communication overhead. By imposing entropy constraints on local updates, EA-FL promotes compressibility, enabling more efficient encoding via entropy coding. We provide a convergence analysis under quantization and channel noise, and introduce an optimization-based bit allocation scheme. Experimental results demonstrate that EA-FL achieves better rate–accuracy trade-off and significantly outperforms uniform bit allocation under the same communication budget. Future work could explore advanced compression techniques beyond entropy coding to improve performance.

VIII. Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

Appendix A
Proof of Lemma 1

It is evident that only two entries of the conditional PMF $P\left(\hat{\delta} \mid \boldsymbol{\delta}_m^{t,\tau}[d]\right)$ depend on the value of $\boldsymbol{\delta}_m^{t,\tau}[d]$. As a result, by (13), only two entries of the marginal PMF $P(\hat{\delta})$ depend on $\boldsymbol{\delta}_m^{t,\tau}[d]$. Let the values for these two entries be denoted by P_1 and P_2. Then, the entropy function can be expressed as

$$\mathsf{H}(\boldsymbol{\delta}_m^{t,\tau}) = -D\ [P_1 \log(P_1) + P_2 \log(P_2)] + \Gamma, \tag{27}$$

where Γ is a constant independent of $\boldsymbol{\delta}_m^{t,\tau}[d]$. Furthermore, using (13), P_1 and P_2 take the form:

$$\begin{aligned} P_1 &= \frac{1}{D}\left(A_1 + \frac{c_{u+1} - |\boldsymbol{\delta}_m^{t,\tau}[d]|}{c_{u+1} - c_u}\right) \\ P_2 &= \frac{1}{D}\left(A_2 + \frac{|\boldsymbol{\delta}_m^{t,\tau}[d]| - c_u}{c_{u+1} - c_u}\right), \end{aligned} \tag{28}$$

where A_1 and A_2 are constants that do not depend on $\boldsymbol{\delta}_m^{t,\tau}[d]$.

Without loss of generality, we assume $\boldsymbol{\delta}_m^{t,\tau}[d] > 0$. (The analysis for $\boldsymbol{\delta}_m^{t,\tau}[d] < 0$ follows by symmetry.) Using (27) and (28), the partial derivative of the entropy w.r.t. $\boldsymbol{\delta}_m^{t,\tau}[d]$ becomes

$$\frac{\partial \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]} = -\left\{\frac{\partial P_1}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]}\left[\ln P_1 + 1\right] + \frac{\partial P_2}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]}\left[\ln P_2 + 1\right]\right\}.$$

Now, observe that $\frac{\partial P_1}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]} = \frac{-1}{D(c_{u+1}-c_u)}$ and $\frac{\partial P_2}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]} = \frac{1}{D(c_{u+1}-c_u)}$. Thus,

$$\begin{aligned} \left\| \frac{\partial \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]} \right\|^2 &= \frac{1}{D^2(c_{u+1}-c_u)^2} \| \ln(P_2/P_1)\|^2 \\ &= \frac{(2^{q_m^t}-1)^2}{D^2(\delta_m^{t,\max})^2} \| \ln(P_2/P_1)\|^2 \le \frac{(2^{q_m^t}-1)^2}{D^2(\delta_m^{t,\max})^2} |\ln \mathsf{P}_0|^2, \end{aligned}$$

where we have used $c_{u+1} - c_u = \frac{\delta_m^{t,\max}}{2^{q_m^t}-1}$. Thus, $\left\| \frac{\partial \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial \boldsymbol{\delta}_m^{t,\tau}} \right\|^2 \le \frac{(2^{q_m^t}-1)^2}{D(\delta_m^{t,\max})^2} |\ln \mathsf{P}_0|^2$.

To determine the smoothness constant $L_{m,h}$, we begin by computing the Lipschitz constant of the entropy function $\mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})$ with respect to a single coordinate $\boldsymbol{\delta}_m^{t,\tau}[d]$, rather than the entire vector $\boldsymbol{\delta}_m^{t,\tau}$. Specifically, we find L_d such that $\left|\frac{\partial^2 \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial (\boldsymbol{\delta}_m^{t,\tau}[d])^2}\right| \le L_d$. Once L_d is established for each dimension $d \in [D]$, the overall smoothness constant is given by $L_h = \sqrt{\sum_{d\in[D]} L_d^2}$, as justified in [35].

First, note that $\frac{\partial^2 P_i}{\partial (\boldsymbol{\delta}_m^{t,\tau}[d])^2} = 0, \quad \text{for } i \in \{1,2\}$. Thus,

$$\frac{\partial^2 \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial (\boldsymbol{\delta}_m^{t,\tau}[d])^2} = -\Big[\big(\frac{\partial P_1}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]}\big)^2 \frac{1}{P_1} + \big(\frac{\partial P_2}{\partial \boldsymbol{\delta}_m^{t,\tau}[d]}\big)^2 \frac{1}{P_2}\Big] \tag{29}$$

$$= \frac{-1}{D^2(c_{u+1}-c_u)^2}\left[\frac{1}{P_1} + \frac{1}{P_2}\right]. \tag{30}$$

Now, using the assumption in the lemma that $P(\hat{\delta}) \ge \mathsf{P}_0$: $\frac{1}{P_1} \le \frac{1}{\mathsf{P}_0}$ and $\frac{1}{P_2} \le \frac{1}{\mathsf{P}_0}$. Therefore, according to (30), we have $\left|\frac{\partial^2 \mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})}{\partial (\boldsymbol{\delta}_m^{t,\tau}[d])^2}\right| \le \frac{2}{\mathsf{P}_0 D^2 (c_{u+1}-c_u)^2}$. This gives us the coordinate-wise Lipschitz constant $L_d = \frac{2}{\mathsf{P}_0 D^2 (c_{u+1}-c_u)^2}$. Thus, $\mathsf{H}(\boldsymbol{\delta}_m^{t,\tau})$ is $L_{m,h}$-smooth where $L_{m,h} = \sqrt{\sum_{d\in D} L_d^2}$. Therefore, $L_{m,h} = \frac{2}{\mathsf{P}_0 D^{3/2}(c_{u+1}-c_u)^2}$. Finally, noting that the quantization interval size satisfies $c_{u+1} - c_u = \frac{\delta_m^{t,\max}}{2^{q_m^t}-1}$, the lemma is concluded.

Appendix B
Proof of Lemma 2

$$\mathbb{E}\left[\left\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\right\|^2\right] \tag{31}$$

$$= \mathbb{E}\left[\left\|\left(\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \mathsf{Q}(\boldsymbol{\delta}_m^t)\right) + \left(\mathsf{Q}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\right)\right\|^2\right] \tag{32}$$

$$= \mathbb{E}\left[\left\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \mathsf{Q}(\boldsymbol{\delta}_m^t)\right\|^2\right] + \mathbb{E}\left[\left\|\mathsf{Q}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\right\|^2\right] + 2\mathbb{E}\left[\langle \widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \mathsf{Q}(\boldsymbol{\delta}_m^t)\ ,\ \mathsf{Q}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\rangle\right] \tag{33}$$

$$= \mathbb{E}\left[\left\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t) - \mathsf{Q}(\boldsymbol{\delta}_m^t)\right\|^2\right] + \mathbb{E}\left[\left\|\mathsf{Q}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\right\|^2\right] \tag{34}$$

where (34) comes from the unbiasedness of the quantizer.

For the first term in (34), the probability of no bit errors is $(1-p_m^t)^{q_m^t}$, and that of at least one bit error is $1-(1-p_m^t)^{q_m^t}$. Let $\mathsf{Q}(\boldsymbol{\delta}_m^t)[d] = c_u$. Then, the decoded symbol is

$$\widetilde{\mathsf{Q}}\big(\boldsymbol{\delta}_m^t[d]\big) = \begin{cases} c_u, & \text{w. p.} \quad (1-p_m^t)^{q_m^t}, \\ c_v \neq c_u, & \text{w. p.} \quad 1-(1-p_m^t)^{q_m^t}. \end{cases} \tag{35}$$

Therefore,

$$\mathbb{E}\Big[\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t[d]) - \mathsf{Q}(\boldsymbol{\delta}_m^t[d])\|^2\Big] = \Big[1-(1-p_m^t)^{q_m^t}\Big] \times \Gamma,$$

where Γ is the conditional MSE given an error. Note that since each q_m^t bit is sent through a BSC with independent bit-flip probability p_m^t, then if a symbol-level error occurs, the decoded bin index $v \neq u$ is equally likely among the other $2^{q_m^t}-1$ bins. To get the average over all possible $\boldsymbol{\delta}_m^t[d]$ values and all code-indices u, note that u itself is uniformly distributed over $\{0,1,\ldots,2^{q_m^t}-1\}$. Hence, $\Gamma = \frac{1}{2^{q_m^t}}\sum_{u=0}^{2^{q_m^t}-1} \frac{1}{2^{q_m^t}-1}\sum_{\substack{v=0\\v\neq i}}^{2^{q_m^t}-1}(c_v-c_u)^2$.

Now, define $\Delta \triangleq \frac{\delta_m^{t,\max}}{2^{q_m^t}-1}$. It then follows that $c_v - c_u = (v-u)\,\Delta$, and that $(c_v-c_u)^2 = (v-u)^2\,\Delta^2$. Hence

$$\Gamma = \Delta^2 \frac{1}{2^{q_m^t}} \sum_{u=0}^{2^{q_m^t}-1} \frac{1}{2^{q_m^t}-1}\sum_{v\neq u}(v-u)^2 = \Delta^2\,\frac{2^{q_m^t}+1}{3},$$

where we have used a standard combinatorial identity. Now, putting back the definition of Δ we have

$$\begin{aligned}\mathbb{E}\Big[\big\|\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t[d]) - \mathsf{Q}(\boldsymbol{\delta}_m^t[d])\big\|^2\Big] &= \big[1-(1-p_m^t)^{\,q_m^t}\big]\times\Gamma \\ &= \Big[1-(1-p_m^t)^{q_m^t}\Big]\,\frac{(\delta_m^{t,\max})^2}{(2^{q_m^t}-1)^2}\,\frac{2^{q_m^t}+1}{3}. \end{aligned} \tag{36}$$

Next, for the second term in (34), we invoke results of [14]:

$$\mathbb{E}\left[\big\|\mathsf{Q}(\boldsymbol{\delta}_m^t) - \boldsymbol{\delta}_m^t\big\|^2\right] \le \frac{D(\delta_m^{t,\max})^2}{4(2^{q_m^t}-1)^2}. \tag{37}$$

Using (36) and (37), the lemma is concluded.

Appendix C
Proof of Lemmas 3, 4, 5, 6 and 7

A. Proof of Lemma 3

$$\mathbb{E}\left[\boldsymbol{w}^{t+1}\right] = \mathbb{E}\left[\boldsymbol{w}^t + \sum_{m\in[M]}\alpha_m\widetilde{\mathsf{Q}}(\boldsymbol{\delta}_m^t)\right] \tag{38}$$

$$= \mathbb{E}\left[\boldsymbol{w}^t + \sum_{m\in[M]}\alpha_m\boldsymbol{\delta}_m^t\right] \tag{39}$$

$$= \mathbb{E}\left[\boldsymbol{w}^t + \sum_{m\in[M]}\alpha_m(\boldsymbol{w}_m^{t+1}-\boldsymbol{w}^t)\right] \tag{40}$$

$$= \mathbb{E}\left[\sum_{m\in[M]}\alpha_m\boldsymbol{w}_m^{t+1}\right] = \mathbb{E}\left[\tilde{\boldsymbol{w}}^{t+1}\right]. \tag{41}$$

where (39) is due to Lemma 2.

B. Proof of Lemma 4

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{w}^{t+1}\right) &= \mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1} + \boldsymbol{w}^{t+1} - \tilde{\boldsymbol{w}}^{t+1}\right) \\ &\le \mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right) + \left\langle\nabla\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right), \boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\rangle \\ &\quad + \frac{L}{2}\left\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\|^2,\end{aligned}$$

where the inequality holds due to the L-smoothness of $\mathcal{L}$. We take expectation on both sides. We also note that $\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}$ only depends on the quantization and channel error, so it is independent of $\nabla\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right)$. Then, using Lemma 3 we obtain

$$\mathbb{E}\mathcal{L}\left(\boldsymbol{w}^{t+1}\right) \le \mathbb{E}\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right) + \frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}^{t+1}\right\|^2. \tag{42}$$

C. Proof of Lemma 5

Note the entropy function does not depend on $\boldsymbol{\xi}$. Therefore, $\mathbb{E}\mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{w}-\boldsymbol{w}^t\right)\right) = \mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{w}-\boldsymbol{w}^t\right)\right)$. As per Assumption 2, $\mathbb{E}[\nabla f_m(\boldsymbol{w},\boldsymbol{\xi})] = \nabla f_m(\boldsymbol{w})$, so: $\mathbb{E}[\nabla\mathcal{L}_m(\boldsymbol{w},\boldsymbol{\xi})] = \nabla\mathcal{L}_m(\boldsymbol{w})$. To bound the second moment of the gradient, we have

$$\begin{aligned}\mathbb{E}\|\nabla\mathcal{L}_m(\boldsymbol{w})\|^2 &= \mathbb{E}\|\nabla f_m(\boldsymbol{w}) + \lambda\nabla\mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{w}-\boldsymbol{w}^t\right)\right)\|^2 \\ &\le 2\mathbb{E}\|\nabla f_m(\boldsymbol{w})\|^2 + 2\mathbb{E}\|\lambda\nabla\mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{w}-\boldsymbol{w}^t\right)\right)\|^2 \quad (43)\\ &\le 2\sigma_f^2 + 2\mathbb{E}\|\lambda\nabla\mathsf{H}\left(\mathsf{Q}\left(\boldsymbol{w}-\boldsymbol{w}^t\right)\right)\|^2 \quad (44)\\ &\le 2\sigma_f^2 + 2\lambda^2(\kappa^2/D)|\ln\mathsf{P}_0|^2, \quad (45)\end{aligned}$$

where (43) is obtained due to the parallelogram identity, (44) is due to Assumption 2, and (45) is due to the definition of κ.

D. Proof of Lemma 6

$$\mathbb{E}\mathcal{L}\left(\tilde{\boldsymbol{w}}^{t+1}\right) = \mathbb{E}\mathcal{L}\left(\sum_m\alpha_m\left(\boldsymbol{w}^t - \eta_t\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right)\right) \tag{46}$$

$$= \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t - \eta_t\sum_m\alpha_m\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right) \tag{47}$$

$$\le \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) - \mathbb{E}\left\langle\nabla\mathcal{L}\left(\boldsymbol{w}^t\right), \eta_t\sum_m\alpha_m\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right\rangle + \frac{L\eta_t^2}{2}\mathbb{E}\left\|\sum_m\alpha_m\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right\|^2 \tag{48}$$

$$= \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) - \mathbb{E}\left\langle\nabla\mathcal{L}\left(\boldsymbol{w}^t\right), \eta_t\sum_m\alpha_m\nabla\mathcal{L}_m(\boldsymbol{w}^t)\right\rangle + \frac{L\eta_t^2}{2}\mathbb{E}\left\|\sum_m\alpha_m\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right\|^2 \tag{49}$$

$$= \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) - \eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2 + \frac{L\eta_t^2}{2}\mathbb{E}\left\|\sum_m\alpha_m\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right\|^2$$

$$\le \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) - \eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2 + \sum_m\frac{M\alpha_m^2L\eta_t^2}{2}\mathbb{E}\left\|\nabla\tilde{\mathcal{L}}_m(\boldsymbol{w}^t)\right\|^2 \tag{50}$$

$$\le \mathbb{E}\mathcal{L}\left(\boldsymbol{w}^t\right) - \eta_t\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^t\right)\right\|^2 + \frac{ML\eta_t^2\sigma^2}{2}\sum_m\alpha_m^2, \tag{51}$$

where the inequality (48) is due to L-smoothness of $\mathcal{L}$, (49) is due to Lemma 5, (50) is due to the Cauchy-Schwarz inequality, and (51) is from Lemma 5.

Lastly, due to space limitation, the proof of Lemma 7 is differed to the full version of the paper.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS '17)*. PMLR, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[4] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2019.

[5] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 323–341, 2021.

[6] Z. Zhao, Y. Mao, Y. Liu, L. Song, Y. Ouyang, X. Chen, and W. Ding, "Towards efficient communications in federated learning: A contemporary survey," *Journal of the Franklin Institute*, vol. 360, no. 12, pp. 8669–8703, 2023.

[7] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS '18)*, vol. 31. Curran Associates Inc., 2018.

[8] A. Xu and H. Huang, "Detached error feedback for distributed sgd with random sparsification," in *Proceedings of the 39th International Conference on Machine Learning (ICML '22)*. PMLR, 2022, pp. 24 550–24 575.

[9] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 374–10 386, 2022.

[10] K. Sayood, *Introduction to Data Compression (5th ed.)*. San Francisco, CA: Morgan Kaufmann, 2017.

[11] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient SGD via gradient quantization and encoding," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS '17)*. Long Beach, CA, USA: Curran Associates Inc., 2017, pp. 1709–1720.

[12] Z. Zhu, Y. Shi, G. Xin, C. Peng, P. Fan, and K. B. Letaief, "Towards efficient federated learning: Layer-wise pruning-quantization scheme and coding design," *Entropy*, vol. 25, no. 8, p. 1205, 2023.

[13] S. M. Hamidi and A. Bereyhi, "Rate-constrained quantization for communication-efficient federated learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '25)*. IEEE, 2025, pp. 1–5.

[14] X. Han, W. Chen, J. Li, M. Ding, Q. Wu, K. Wei, X. Deng, and Z. Mei, "Energy-efficient wireless federated learning via doubly adaptive quantization," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.

[15] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," arXiv preprint, 2020, arXiv:2006.10672.

[16] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," arXiv preprint, 2018, arXiv:1812.01097.

[17] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Canada, Technical Report TR-2009, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[18] Y. Le and X. Wu, "Tiny imagenet visual recognition challenge," https://tiny-imagenet.herokuapp.com/, 2015, accessed: 2025-07-27.

[19] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 194–203, 1975.

[20] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[21] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[22] ——, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN '19)*. IEEE, 2019, pp. 1–8.

[23] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proceedings of the International Conference on Learning Representations (ICLR '18)*, 2018.

[24] K. Yue, R. Jin, C.-W. Wong, and H. Dai, "Communication-efficient federated learning via predictive coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 369–380, 2022.

[25] X. Su, Y. Zhou, L. Cui, J. C. S. Lui, and J. Liu, "Fed-cvlc: Compressing federated learning communications with variable-length codes," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '24)*. IEEE, 2024, pp. 601–610.

[26] K. Deng, Z. Chen, S. Zhang, C. Gong, and J. Zhu, "Content compression coding for federated learning," in *Proceedings of the 11th International Conference on Wireless Communications and Signal Processing (WCSP '19)*. Xi'an, China: IEEE, 2019, pp. 1–6.

[27] J. Wang, M. Dong, B. Liang, G. Boudreau, and A. Afana, "Exploring temporal similarity for joint computation and communication in online distributed optimization," *IEEE/ACM Transactions on Networking*, pp. 1–17, 2025.

[28] J. Wang, B. Liang, M. Dong, G. Boudreau, and A. Afana, "Online distributed optimization with efficient communication via temporal similarity," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[29] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 5204–5219, 2022.

[30] M. F. Sabir, H. R. Sheikh, R. W. Heath, and A. C. Bovik, "A joint source-channel distortion model for jpeg compressed images," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1349–1364, 2006.

[31] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[32] S. U. Stich, "Local SGD converges fast and communicates little," in *International Conference on Learning Representations*, 2019.

[33] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS '20)*. PMLR, 2020, pp. 2021–2031.

[34] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Uveqfed: Universal vector quantization for federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 500–514, 2021.

[35] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004, ch. 9, pp. 413–468.