Vector Quantized Compressed Sensing for Communication-Efficient Federated Learning

Yongjeong Oh*, Yo-Seb Jeon*, Mingzhe Chen†, and Walid Saad‡
*Department of Electrical Engineering, POSTECH, Pohang, Gyeongbuk 37673, South Korea
†Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA
‡Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060, USA
Email: {yongjeongoh,yoseb.jeon}@postech.ac.kr, mingzhec@princeton.edu, walids@vt.edu

***Abstract*—In this paper, a communication-efficient federated learning (FL) framework is proposed, which leverages ideas from vector quantized compressed sensing, for the first time, to compress the local model updates at wireless devices in FL. For the compression, each local model update is projected onto a lower dimensional space; then, the projected local model update is quantized by using a vector quantizer. The global model update at a parameter server is reconstructed by using a sparse signal recovery algorithm on the aggregation of the compressed local model updates. A key feature of our compression strategy is that the local model update after the projection is effectively modeled as a Gaussian random vector by the central limit theorem. Inspired by this feature, the optimal vector quantizer is derived for minimizing the compression error of the local model update. Simulation results on the MNIST dataset demonstrate that the proposed framework that uses 0.5 bit to represent each local model update entry shows less than a 1% decrease in classification accuracy compared to FL without local update compression.**

I. Introduction

Federated learning (FL) is a distributed machine learning technique that trains a *global* learning model on a parameter server (PS) by enabling distributed wireless devices to collaborate and leverage their own local training datasets [1], [2]. Typically, in FL, each device updates its *local* model based on the local training dataset and then sends a local model update to the PS [1]. After the transmission from the devices, the PS updates its global model by aggregating the local model updates sent by the devices and then distributes the updated global model to the devices. This two-step training process continues until the global model at the PS converges. The above FL framework can help preserve the privacy of the data generated by the devices, but also faces several challenges in real-world applications. One of the major challenges is the significant communication overhead required for transmitting the local model updates from the wireless devices to the PS because the wireless links connecting them have limited capacity in practical communication systems. This challenge becomes critical when the dimensionality of the local model updates is much higher than the capacity of the wireless links.

To address the above challenge, communication-efficient FL via local model update compression has been extensively studied in [3]–[10]. The common idea in these prior works is to apply *lossy* compression to the local model updates, in order to reduce the communication overhead required for transmitting these updates. A representative approach in this direction is the quantization approach in which the entries of the local model update are quantized by a scalar quantizer [3] or by a vector quantizer [4], [5]. A more recent approach is the quantized compressed sensing (QCS) approach motivated by the sparsity of the local model update, obtained either naturally or by applying sparsification [6]. The basic idea of the QCS-based compression is to project the local model updates onto a lower dimensional space as in compressed sensing (CS), before they are quantized [8]–[10]. Because of the ensuing dimensionality reduction, this approach provides a better reduction in the communication overhead compared with the quantization-only approach. The existing QCS-based methods, however, only consider scalar quantization which is generally inferior to vector quantization in terms of quantization error; thereby, the compression error of these methods becomes problematic as the level of the compression increases. To our best knowledge, a vector QCS approach for communication-efficient FL has never been studied before, despite its potential for not only improving the communication efficiency of FL but also mitigating the compression error of the local model updates.

In this paper, we propose a novel communication-efficient FL framework that leverages, for the first time, both vector quantization and CS-based dimensionality reduction to compress local model updates at the wireless devices. In this framework, we compress each local model update by reducing its dimensionality based on CS and, then, quantizing a projected local model update by using a vector quantizer. For accurate but efficient reconstruction of the global model update at the PS, we aggregate a group of the compressed local model updates and estimate the aggregated model update by applying a sparse signal recovery algorithm. Our key observation is that the local model update after the projection is modeled as an independent and identically distributed (IID) Gaussian random vector by the central limit theorem. Motivated by this observation, we employ a shape-gain quantizer in [11] and design the optimal shape and gain quantizers for minimizing the compression error of the local model update. derive the optimal bit allocation between

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1C1C1010074), and in part by the U.S. National Science Foundation under Grant CNS-2114267.

the shape and gain quantizers by characterizing the mean-squared-error (MSE) performance of these quantizers. Using simulations, we demonstrate the superiority of the proposed FL framework over the existing FL frameworks for an image classification task using the MNIST dataset.

II. System Model

We consider a wireless FL system in which a PS trains a global model by collaborating with a set $\mathcal{K}$ of K wireless devices over wireless links with limited capacity. The data samples for training the global model are assumed to be distributed over the wireless devices only, while the PS does not have direct access to them. Let $\mathcal{D}_k$ be the *local training dataset* available at device $k \in \mathcal{K}$. The global model at the PS is assumed to be represented by a parameter vector $\boldsymbol{w} \in \mathbb{R}^{\bar{N}}$, where $\bar{N}$ is the number of global model parameters. The goal of FL is to find the best parameter vector that minimizes the *global* loss function, defined as

$$F(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{u} \in \mathcal{D}} f(\boldsymbol{w}; \boldsymbol{u}), \tag{1}$$

where $f(\boldsymbol{w}; \boldsymbol{u})$ is a loss function that measures how well the global model with the parameter vector $\boldsymbol{w}$ fits one particular data sample $\boldsymbol{u} \in \mathcal{D}_k$, and $\mathcal{D} = \cup_k \mathcal{D}_k$. The global loss function in (1) can be rewritten as

$$F(\boldsymbol{w}) = \frac{1}{\sum_{j=1}^{K} |\mathcal{D}_j|} \sum_{k=1}^{K} |\mathcal{D}_k| F_k(\boldsymbol{w}), \tag{2}$$

where $F_k(\boldsymbol{w}) = \frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{u} \in \mathcal{D}_k} f(\boldsymbol{w}; \boldsymbol{u})$ is a *local* loss function at device k. A typical FL framework for minimizing the global loss function in (2) involves alternating between *local* model update at wireless devices and *global* model update at the PS in each communication round, as explained in [1].

1) Local model update at wireless devices: In the local model update process, each wireless device updates a local parameter vector based on its own local training dataset. Then each device sends a local model update (i.e., the difference between the parameter vectors before and after the local update) to the PS. Let $\boldsymbol{w}^{(t)} \in \mathbb{R}^{\bar{N}}$ be the parameter vector shared by the devices at communication round $t \in \{1, \ldots, T\}$, where T is the total number of communication rounds. Assume that every device employs a mini-batch stochastic gradient descent (SGD) algorithm with $E \geq 1$ local iterations for updating the parameter vector $\boldsymbol{w}^{(t)}$. Then the updated parameter vector at device k after $E \geq 1$ local iterations is given by

$$\boldsymbol{w}_k^{(t,e+1)} = \boldsymbol{w}_k^{(t,e)} - \eta^{(t)} \nabla F_k^{(t,e)}\big(\boldsymbol{w}_k^{(t,e)}\big), \ \forall e \in \{1, \ldots, E\}, \tag{3}$$

where $\boldsymbol{w}_k^{(t,1)} = \boldsymbol{w}^{(t)}$, $\eta^{(t)}$ is a local learning rate,

$$\nabla F_k^{(t,e)}\big(\boldsymbol{w}_k^{(t,e)}\big) = \frac{1}{|\mathcal{D}_k^{(t,e)}|} \sum_{\boldsymbol{u} \in \mathcal{D}_k^{(t,e)}} \nabla f(\boldsymbol{w}_k^{(t,e)}; \boldsymbol{u}), \tag{4}$$

and $\mathcal{D}_k^{(t,e)}$ is a mini-batch randomly drawn from $\mathcal{D}_k$ at the e-th local iteration of round t. As a result, the local model update sent by device k in round t is determined as

$$\boldsymbol{g}_k^{(t)} = \frac{1}{\eta^{(t)} E} (\boldsymbol{w}^{(t)} - \boldsymbol{w}_k^{(t,E+1)}) \in \mathbb{R}^{\bar{N}}. \tag{5}$$

2) Global model update at the PS: In the global model update process, the PS updates a global parameter vector by aggregating the local model updates sent by K devices. Then the PS broadcasts the updated parameter vector to these devices. Under the assumption of perfect reception of K local model updates, the PS can reconstruct the *global* model update, defined as

$$\boldsymbol{g}_{\mathcal{K}}^{(t)} = \sum_{k=1}^{K} \rho_k \boldsymbol{g}_k^{(t)}, \tag{6}$$

where $\rho_k \triangleq \frac{\sum_{e=1}^{E} |\mathcal{D}_k^{(t,e)}|}{\sum_{j=1}^{K} \sum_{e=1}^{E} |\mathcal{D}_j^{(t,e)}|}$ is invariant in each communication round. If the PS employs the SGD algorithm for updating the parameter vector $\boldsymbol{w}^{(t)}$, then the updated parameter vector in round t will be given by $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \gamma^{(t)} \boldsymbol{g}_{\mathcal{K}}^{(t)}$, where $\gamma^{(t)}$ is a global learning rate. Then the PS broadcasts $\boldsymbol{w}^{(t+1)}$ to the wireless devices, which triggers the start of the local model update process in round $t+1$.

A major challenge in realizing the FL framework discussed above is the significant communication overhead required for transmitting the local model updates from the wireless devices to the PS because the wireless links connecting them may have limited capacity in practical communication systems. This challenge can be mitigated by applying lossy compression to the local model updates, but such a compression approach leads to an inevitable error in the global model update reconstructed at the PS. Moreover, in general, the higher the level of compression at the wireless devices, the larger the reconstruction error at the PS. Therefore, it is essential to develop a proper compression strategy for local model updates that not only provides a considerable reduction in the communication overhead, but also minimizes the reconstruction error at the PS.

III. Proposed Federated Learning Framework

In this section, we present a communication-efficient FL framework that can reduce the communication overhead for transmitting local model updates at the wireless devices while also enabling an accurate reconstruction of the global model update at the PS.

A. Compression of Local Model Updates

In the proposed FL framework, each device compresses its local model update by sequentially performing (i) block sparsification, (ii) dimensionality reduction, and (iii) vector quantization. The details of each step performed by the device k for the block b in round t are elaborated below.

1) Block sparsification: In the block sparsification step, each device divides its local model update into B blocks, each of which has a length of $N = \frac{\bar{N}}{B}$. Let $\boldsymbol{g}_k^{(t,b)} \in \mathbb{R}^N$ be the b-th block of $\boldsymbol{g}_k^{(t)}$, and we refer to this vector as the b-th local *block* update of device k in round t. Then, each local block update is sparsified by nullifying all but the most significant S entries in terms of their magnitudes, where $S < N$ is a sparsity level. To compensate for the information loss during the sparsification, the nullified entries are added to the local block update in the next communication round, as done in [6]. With this compensation strategy, the local block update that needs to be sparsified at device k is expressed as

$$\bar{\boldsymbol{g}}_k^{(t,b)} = \boldsymbol{g}_k^{(t,b)} + \boldsymbol{\Delta}_k^{(t-1,b)}, \tag{7}$$

where $\boldsymbol{\Delta}_k^{(t,b)} \in \mathbb{R}^N$ is a residual vector stored by device k for block b in round t. As a result, the local block update after the sparsification is obtained as $\tilde{\boldsymbol{g}}_k^{(t,b)} = \mathsf{Sparse}(\bar{\boldsymbol{g}}_k^{(t,b)})$ which has only S_k nonzero entries. Then, the residual vector is updated as

$$\boldsymbol{\Delta}_k^{(t,b)} = \bar{\boldsymbol{g}}_k^{(t,b)} - \mathsf{Sparse}(\bar{\boldsymbol{g}}_k^{(t,b)}). \tag{8}$$

2) Dimensionality Reduction: In the dimensionality reduction step, each local block update after the sparsification is projected onto a lower dimensional space as in CS [12]. Let $R_k = \frac{N}{M_k} \geq 1$ be a dimensionality reduction ratio chosen by device k, where $M_k \leq N$ is the dimension after projection. Then, the compressed local block update $\boldsymbol{x}_k^{(t,b)} \in \mathbb{R}^{M_k}$ is obtained as

$$\boldsymbol{x}_k^{(t,b)} = \alpha_k^{(t,b)} \boldsymbol{A}_{R_k} \tilde{\boldsymbol{g}}_k^{(t,b)}, \tag{9}$$

where $\alpha_k^{(t,b)} \in \mathbb{R}$ is a scaling factor, and $\boldsymbol{A}_{R_k} \in \mathbb{R}^{M_k \times N}$ is a projection matrix. We set the scaling factor as $\alpha_k^{(t,b)} = 1/\|\tilde{\boldsymbol{g}}_k^{(t,b)}\|$ and the projection matrix as an IID Gaussian random matrix with zero mean and unit variance, i.e., $(\boldsymbol{A}_{R_k})_{m,n} \sim \mathcal{N}(0,1)$, $\forall m, n$.

3) Vector quantization: In the vector quantization step, each compressed local block update is quantized using a $Q_k M_k$-bit vector quantizer, where Q_k is the number of quantization bits per entry chosen by device k. To avoid the computational complexity for quantizing a high-dimensional vector, the compressed local block update $\boldsymbol{x}_k^{(t,b)}$ is first partitioned into P_k subvectors of dimension $L = \frac{M_k}{P_k}$, i.e., $\boldsymbol{x}_k^{(t,b)} = \left[(\boldsymbol{v}_{k,1}^{(t,b)})^\mathsf{T}, \ldots, (\boldsymbol{v}_{k,P_k}^{(t,b)})^\mathsf{T}\right]^\mathsf{T}$, as in [5]. Then, these subvectors are quantized in parallel using a $Q_k L$-bit vector quantizer. The resulting quantized subvector is given by

$$\hat{\boldsymbol{v}}_{k,p}^{(t,b)} = \mathsf{Q}_\mathcal{C}(\boldsymbol{v}_{k,p}^{(t,b)}),\ p \in \{1, \ldots, P_k\}, \tag{10}$$

where $\mathsf{Q}_\mathcal{C} : \mathbb{R}^L \to \mathcal{C}$ is a vector quantizer with a codebook $\mathcal{C}$ such that $|\mathcal{C}| \leq 2^{Q_k L}$. After vector quantization, the quantized subvectors $\{\hat{\boldsymbol{v}}_{k,p}^{(t,b)}\}_{p=1}^P$ and the scaling factor $\alpha_k^{(t,b)}$ are encoded into digital bits, denoted by $\Omega_{k,b}^{(t)}$, for digital communications.

Remark (Communication overhead of the proposed FL framework): In the proposed FL framework, each device k transmits the digital bits representing the quantized subvectors $\{\hat{\boldsymbol{v}}_{k,p}^{(t,b)}\}_{p=1}^{P_k}$ and the scaling factor $\alpha_k^{(t,b)}$ for every block $b \in \{1, \ldots, B\}$ in round t. Because every quantized subvector is represented by $Q_k L$ bits, the total number of digital bits required for transmitting the local model update at each device is given by $(Q_k L P_k + 32)B = (Q_k M_k + 32)B$ for every round. If $Q_k M_k \gg 32$, the communication overhead for device k is $\frac{Q_k M_k B}{NB} = \frac{Q_k}{R_k}$ bits per local model entry. Owing to this feature, the proposed framework allows each device to not only reduce the communication overhead for transmitting its local model update but also control this overhead by adjusting the values of Q_k and R_k based on the capacity of the wireless link.

B. Reconstruction Strategy

In the proposed FL framework, the PS reconstructs a global model update from the compressed local model updates by sequentially performing (i) group-wise aggregation and (ii) sparse signal recovery. The details of each step performed by the PS for block b in round t are elaborated below.

1) Group-wise aggregation: In the group-wise aggregation step, the PS groups the compressed local block updates with the same dimension and, then, aggregates only the compressed updates in the same group. Meanwhile, to facilitate an accurate reconstruction of the aggregated block update, the PS limits the number of the compressed updates in each group to be less than or equal to K'. By adjusting K', the PS can control the tradeoff between the complexity and accuracy of the global update reconstruction.

Let $\mathcal{K}_g$ be the indices of the compressed local block updates in group $g \in \{1, \ldots, G\}$, where G is the number of groups, and $\mathcal{K}_1, \ldots, \mathcal{K}_G$ are mutually exclusive subsets of $\mathcal{K}$ such that $\mathcal{K} = \bigcup_{g=1}^G \mathcal{K}_g$. For ease of exposition, we assume that $|\mathcal{K}_g| = K'$, $\forall g$. Under error-free reception of the transmitted data $\{\Omega_k^{(t,b)}\}_{k \in \mathcal{K}}$, the PS obtains the quantized subvectors $\{\hat{\boldsymbol{v}}_{k,p}^{(t,b)}\}_{p=1}^{P_k}$ and the scaling factor $\alpha_k^{(t,b)}$ by decoding $\Omega_k^{(t,b)}$. Let $\boldsymbol{d}_{k,p}^{(t,b)} \triangleq \hat{\boldsymbol{v}}_{k,p}^{(t,b)} - \boldsymbol{v}_{k,p}^{(t,b)}$ be a quantization error vector. Then, from $\hat{\boldsymbol{v}}_{k,p}^{(t,b)} = \boldsymbol{v}_{k,p}^{(t,b)} + \boldsymbol{d}_{k,p}^{(t,b)}$, the compressed local block update $\hat{\boldsymbol{x}}_k^{(t,b)}$ is expressed as

$$\begin{aligned}\hat{\boldsymbol{x}}_k^{(t,b)} &= \left[(\hat{\boldsymbol{v}}_{k,1}^{(t,b)})^\mathsf{T}, \ldots, (\hat{\boldsymbol{v}}_{k,P}^{(t,b)})^\mathsf{T}\right]^\mathsf{T} = \boldsymbol{x}_k^{(t,b)} + \boldsymbol{d}_k^{(t,b)} \\ &= \alpha_k^{(t,b)} \boldsymbol{A}_{R_k} \tilde{\boldsymbol{g}}_k^{(t,b)} + \boldsymbol{d}_k^{(t,b)},\end{aligned} \tag{11}$$

where $\boldsymbol{d}_k^{(t,b)} = \left[(\boldsymbol{d}_{k,1}^{(t,b)})^\mathsf{T}, \ldots, (\boldsymbol{d}_{k,P}^{(t,b)})^\mathsf{T}\right]^\mathsf{T}$. Because $R_k = R_g$, $\forall k \in \mathcal{K}_g$, the PS can aggregate the compressed local block updates in group g as

$$\boldsymbol{y}_{\mathcal{K}_g}^{(t,b)} = \sum_{k \in \mathcal{K}_g} \frac{\rho_k}{\alpha_k^{(t,b)}} \hat{\boldsymbol{x}}_k^{(t,b)} = \boldsymbol{A}_{R_g} \tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)} + \boldsymbol{d}_{\mathcal{K}_g}^{(t,b)}, \tag{12}$$

where $\boldsymbol{d}_{\mathcal{K}_g}^{(t,b)} = \sum_{k \in \mathcal{K}_g} (\rho_k / \alpha_k^{(t,b)}) \boldsymbol{d}_k^{(t,b)}$ and $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)} = \sum_{k \in \mathcal{K}_g} \rho_k \tilde{\boldsymbol{g}}_k^{(t,b)}$ is the aggregated block update of group g for block b in round t.

2) Sparse signal recovery: In the sparse signal recovery step, the PS estimates the aggregated block update $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$ from its noisy linear observation $\boldsymbol{y}_{\mathcal{K}_g}^{(t,b)}$ in (12) for every group g. As can be seen in (12), the estimation of $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$ from $\boldsymbol{y}_{\mathcal{K}_g}^{(t,b)}$ is a well-known sparse signal recovery problem [12]. Motivated by this fact, the PS employs a sparse signal recovery algorithm (e.g., [13]) to estimate $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$ from $\boldsymbol{y}_{\mathcal{K}_g}^{(t,b)}$ for every group g. If the sparsity level of $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$ is sufficiently smaller than the dimension of $\boldsymbol{y}_{\mathcal{K}_g}^{(t,b)}$, the PS is able to attain an accurate estimate of $\tilde{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$. After the recovery of the aggregated block updates for all groups, the PS reconstructs the global block update as $\hat{\boldsymbol{g}}_{\mathcal{K}}^{(t,b)} = \sum_{g=1}^{G} \hat{\boldsymbol{g}}_{\mathcal{K}_g}^{(t,b)}$. Finally, the PS obtains the global model update $\hat{\boldsymbol{g}}_{\mathcal{K}}^{(t)}$ by concatenating all the global block updates $\{\hat{\boldsymbol{g}}_{\mathcal{K}}^{(t,b)}\}_{b=1}^{B}$.

IV. Parameter Optimization of Vector Quantizer

In the proposed FL framework, a proper design of the vector quantizer in (10) is critical for reducing the reconstruction error of the global model update at the PS. In this section, we optimize the design of the vector quantizer for the local update compression of the proposed FL framework.

A. Quantizer design problem

The underlying challenge of the vector quantizer design is that the exact distribution of the quantizer input depends on many factors such as the global model choice, the local training data distribution, and the loss function type. Moreover, the optimal vector quantizer may differ across partitions, blocks, devices, and communication rounds, which can lead to additional communication overhead for transmitting the information of the quantizer between the PS and the devices. Fortunately, the above challenges can be readily addressed in the proposed framework. First of all, it is reported in [10] that a sparsified local block update $\tilde{\boldsymbol{g}}_{k}^{(t,b)}$ can be modeled as an IID random vector whose entry follows a Bernoulli Gaussian-mixture distribution. Hence, for a large N, the projected local model update $\boldsymbol{x}_{k}^{(t,b)} = \alpha_{k}^{(t,b)} \boldsymbol{A}_{R_k} \tilde{\boldsymbol{g}}_{k}^{(t,b)}$ can be modeled as an M_k-dimensional IID Gaussian random vector with zero mean and unit variance by the central limit theorem, provided that $\boldsymbol{A}_{R_k}$ is an IID Gaussian random matrix with $\alpha_{k}^{(t,b)} = 1/\|\boldsymbol{g}_{k}^{(t,b)}\|$. This implies that every subvector of $\boldsymbol{x}_{k}^{(t,b)}$ can also be modeled as an L-dimensional IID Gaussian random vector whose distribution is same for all partitions, blocks, devices, and communication rounds, i.e., $\boldsymbol{v}_{k,p}^{(t,b)} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$, $\forall k, p, t, b$. Motivated by this observation, we use the vector quantizer optimized for the distribution of $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$ for a given number of quantization bits. Owing to this feature, both the PS and devices can utilize the same optimal quantizer by sharing the number of quantization bits.

To determine the optimal vector quantizer for the distribution of $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$, we consider a *shape-gain* quantizer, which is effective in quantizing an IID Gaussian random vector while enabling the efficient construction of the optimal codebook [11]. When employing this quantizer, the shape of $\boldsymbol{v}$, defined as $\boldsymbol{s} = \boldsymbol{v}/\|\boldsymbol{v}\|$, and its gain defined as $h = \|\boldsymbol{v}\|$ are independently quantized. Let $\mathsf{Q}_{\mathcal{C}_\mathrm{s}}(\boldsymbol{s}) = \operatorname*{argmin}_{\hat{\boldsymbol{s}} \in \mathcal{C}_\mathrm{s}} \|\boldsymbol{s} - \hat{\boldsymbol{s}}\|^2$ and $\mathsf{Q}_{\mathcal{C}_\mathrm{h}}(h) = \operatorname*{argmin}_{\hat{h} \in \mathcal{C}_\mathrm{h}} |h - \hat{h}|^2$ be the shape and gain quantizers using the minimum Euclidean distance criterion, respectively. Then the output of the shape-gain quantizer is given by $\hat{\boldsymbol{v}} = \hat{h}\hat{\boldsymbol{s}}$, where $\hat{\boldsymbol{s}} = \mathsf{Q}_{\mathcal{C}_\mathrm{s}}(\boldsymbol{s})$ and $\hat{h} = \mathsf{Q}_{\mathcal{C}_\mathrm{h}}(h)$. Given this fact, the MSE of the shape-gain quantizer can be approximated by [11]

$$\mathbb{E}\big[\|\boldsymbol{v} - \hat{h}\hat{\boldsymbol{s}}\|^2\big] \approx L \cdot \mathsf{MSE}(\boldsymbol{s}; \mathcal{C}_\mathrm{s}) + \mathsf{MSE}(h; \mathcal{C}_\mathrm{h}), \tag{13}$$

where $\mathsf{MSE}(h; \mathcal{C}_\mathrm{h}) = \mathbb{E}\big[|h - \mathsf{Q}_{\mathcal{C}_\mathrm{h}}(h)|^2\big]$ and $\mathsf{MSE}(\boldsymbol{s}; \mathcal{C}_\mathrm{s}) = \mathbb{E}\big[\|\boldsymbol{s} - \mathsf{Q}_{\mathcal{C}_\mathrm{s}}(\boldsymbol{s})\|^2\big]$. By utilizing this approximation, we formulate the shape-gain quantizer design problem for a given number of quantization bits QL as follows:

$$\begin{aligned} &\operatorname*{argmin}_{\mathcal{C}_\mathrm{s}, \mathcal{C}_\mathrm{h}} L \cdot \mathsf{MSE}(\boldsymbol{v}/\|\boldsymbol{v}\|; \mathcal{C}_\mathrm{s}) + \mathsf{MSE}(\|\boldsymbol{v}\|; \mathcal{C}_\mathrm{h}), \\ &\quad \text{s.t.} \quad |\mathcal{C}_\mathrm{s}| \le 2^{Q_\mathrm{s}},\ |\mathcal{C}_\mathrm{h}| \le 2^{Q_\mathrm{h}},\ Q_\mathrm{s} + Q_\mathrm{h} \le QL, \end{aligned} \tag{14}$$

where $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$.

B. Design of shape quantizer

To solve (14), as a first step, we determine the MSE-optimal shape quantizer for a given number of shape quantization bits Q_s, by solving the following problem:

$$\mathcal{C}_\mathrm{s}^\star = \operatorname*{argmin}_{|\mathcal{C}_\mathrm{s}| \le 2^{Q_\mathrm{s}}} \mathsf{MSE}(\boldsymbol{s}; \mathcal{C}_\mathrm{s}), \tag{15}$$

where $\boldsymbol{s} = \boldsymbol{v}/\|\boldsymbol{v}\|$ with $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$. Unfortunately, it is hard to characterize the exact solution of this problem; thus, as an alternative, we consider an approximate solution by replacing the Euclidean distance with the squared chordal distance, as done in [5]. Then the approximate solution of (15) is obtained as an even Grassmannian codebook $\mathcal{C}_\mathrm{s}^\star$ which is constructed using the following strategy [5]. First, solve the Grassmannian line packing problem formulated as

$$\max_{\mathcal{C}_\mathrm{s}^+} \min_{\hat{\boldsymbol{s}} \neq \hat{\boldsymbol{s}}', \hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}' \in \mathcal{C}_\mathrm{s}^+} d(\hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}'), \quad \text{s.t.} \ \ |\mathcal{C}_\mathrm{s}^+| = 2^{Q_\mathrm{s}-1}, \tag{16}$$

where $d(\hat{\boldsymbol{s}}, \hat{\boldsymbol{s}}') = \sqrt{1 - |\hat{\boldsymbol{s}}^\mathsf{T} \hat{\boldsymbol{s}}'|^2}$. Next, construct the even Grassmannian codebook as $\mathcal{C}_\mathrm{s}^\star = \mathcal{C}_\mathrm{s}^+ \bigcup \mathcal{C}_\mathrm{s}^-$ where $\mathcal{C}_\mathrm{s}^- = \{\boldsymbol{s} : -\boldsymbol{s} \in \mathcal{C}_\mathrm{s}^+\}$. We determine the optimal shape quantizer by using the even Grassmannian codebook $\mathcal{C}_\mathrm{s}^\star$ constructed above. For large L, the upper bound of the MSE of this quantizer is characterized as [14]

$$\mathsf{MSE}(\boldsymbol{s}; \mathcal{C}_\mathrm{s}^\star) \le \mathbb{E}[2d^2(\boldsymbol{s}, \hat{\boldsymbol{s}})] \le 2^{-\frac{2(Q_\mathrm{s}-1)}{L-1}+1}. \tag{17}$$

C. Design of gain quantizer

As a next step, we characterize the MSE-optimal gain quantizer for a given number of gain quantization bits Q_h, by solving the following problem:

$$\mathcal{C}_\mathrm{h}^\star = \operatorname*{argmin}_{|\mathcal{C}_\mathrm{h}| \le 2^{Q_\mathrm{h}}} \mathsf{MSE}(h; \mathcal{C}_\mathrm{h}), \tag{18}$$

where $h = \|\boldsymbol{v}\|$. Because the distribution of $h = \|\boldsymbol{v}\|$ for $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L)$ is strictly log-concave, the Lloyd–Max

algorithm converges to a globally optimal quantizer in terms of minimizing the MSE [15]. Hence, when $Q_{\rm h} > 0$, we determine the optimal gain quantizerby applying the Lloyd–Max algorithm for the distribution of $h = \|\boldsymbol{v}\|$. When $Q_{\rm h} = 0$, our strategy is to approximate the gain $h = \|\boldsymbol{v}\|$ as its expected value $\mathbb{E}[h] = \sqrt{2}\frac{\Gamma((L+1)/2)}{\Gamma(L/2)}$ based on its distribution. Considering these two cases, the MSE of the above gain quantizer is approximately characterized as [16]

$$\mathsf{MSE}(h; \mathcal{C}_{\rm h}^{\star}) \approx \begin{cases} \chi_L 2^{-2(Q_{\rm h}+1)}, & Q_{\rm h} > 0, \\ L - \frac{2\pi}{\beta^2\left(\frac{L}{2}, \frac{1}{2}\right)}, & Q_{\rm h} = 0, \end{cases} \tag{19}$$

where $\chi_L = \frac{3^{\frac{L}{2}}\Gamma^3(\frac{L+2}{6})}{2\Gamma(\frac{L}{2})}$.

D. Optimal bit allocation

Now, we derive the optimal bit allocation for the optimal shape and gain quantizers based on the MSE characterizations in (17) and (19). This result is given in the following theorem:

Theorem 1: If both the upper bound in (17) and the approximation in (19) hold with equality, the optimal bit allocation $\left(Q_{\rm s}^{\star}, Q_{\rm h}^{\star}\right)$ in (14) is given by

$$\left(Q_{\rm s}^{\star}, Q_{\rm h}^{\star}\right) = \begin{cases} \left(QL - H_{L,Q}, H_{L,Q}\right), & \text{if } F_{L,Q} \leq 0, \\ \left(QL, 0\right), & \text{if } F_{L,Q} > 0, \end{cases} \tag{20}$$

where $H_{L,Q} = \frac{L-1}{2L}\log_2\left(\frac{L-1}{2L}\chi_L\right) + Q - 1$ and

$$F_{L,Q} = L2^{-\frac{2(QL-1)}{L-1}+1}\left(2^{\frac{2H_{L,Q}}{L-1}+1} - 1\right) + \chi_L 2^{-2(H_{L,Q}+1)} - L + \frac{2\pi}{\beta^2\left(\frac{L}{2}, \frac{1}{2}\right)}.$$

Proof: From (17) and (19), the bit allocation problem with $Q_{\rm h} > 0$ is formulated as

$$(\mathbf{P}_1) \quad \underset{Q_{\rm s}, Q_{\rm h}}{\operatorname{argmin}} \; L2^{-\frac{2(Q_{\rm s}-1)}{L-1}+1} + \chi_L 2^{-2(Q_{\rm h}+1)}, \quad \text{s.t.} \quad Q_{\rm s} + Q_{\rm h} \leq QL, \; Q_{\rm h} > 0. \tag{21}$$

From the Karush-Kuhn-Tucker (KKT) conditions, the solution of $(\mathbf{P}_1)$ is given by $\left(Q_{\rm s}, Q_{\rm h}\right) = (QL - H_{L,Q}, H_{L,Q})$, where $H_{L,Q} = \frac{L-1}{2L}\log_2\left(\frac{L-1}{2L}\chi_L\right) + Q - 1$. Similarly, the bit allocation problem with $Q_{\rm h} = 0$ is formulated as

$$(\mathbf{P}_2) \quad \underset{Q_{\rm s} \leq QL}{\operatorname{argmin}} \; L2^{-\frac{2(Q_{\rm s}-1)}{L-1}+1} + L - \frac{2\pi}{\beta^2\left(\frac{L}{2}, \frac{1}{2}\right)}. \tag{22}$$

Because the objective function of $(\mathbf{P}_2)$ is a decreasing function of $Q_{\rm s}$, the solution is given by $Q_{\rm s} = QL$. The optimal bit allocation in (20) can be obtained by comparing $F_{L,1}(QL - H_{L,Q}, H_{L,Q})$ and $F_{L,2}(QL, 0)$. This completes the proof. ■

V. Simulation Results

In this section, we demonstrate the superiority of the proposed FL framework over existing FL frameworks, using simulations. We assume that the communication overhead allowed for transmitting the local model update at device k is given by C_k bits per local model entry. Under this assumption, we consider two communication scenarios: (i) *homogeneous* and (ii) *heterogeneous*. In the homogeneous scenario, we set $C_k = \bar{C}$, $\forall k \in \mathcal{K}$, to model the wireless links with the same capacity; whereas, in the heterogeneous scenario, we uniformly draw C_k from a pre-defined set $\mathcal{C}$ to model the wireless links with different capacities.

In this simulation, an image classification task using the publicly accessible MNIST dataset is considered with $K = 75$ and $T = 50$. A global model is set to be a fully-connected neural network that consists of 784 input nodes, a single hidden layer with 20 hidden nodes, and 10 output nodes. The activation functions of the hidden layer and the output layer are set to the rectified linear unit and the softmax function, respectively. For the global model training at the PS, the ADAM optimizer with an initial learning rate 0.01 is adopted. Each local training dataset is determined by randomly selecting 500 training data samples from two classes. For the local model training at each device, the mini-batch SGD algorithm with a learning rate 0.01 is adopted with $|\mathcal{D}_k^{(t,e)}| = 10$ and $E = 3$. For both the ADAM optimizer and the mini-batch SGD algorithm, the cross-entropy loss function is used.

For performance comparisons, we consider the following FL frameworks:

- *FedAvg:* This framework assumes *lossless* transmission of the local model updates from the wireless devices to the PS without compression, as done in [1].
- *Proposed FL framework:* This is the proposed FL framework when $R_k = 1.5$, $Q_k = C_k R_k$, $G = 25$, and $B = 10$. The EM-GAMP algorithm in [10] is employed as a sparse signal recovery algorithm during the global block update reconstruction. The dimension of a subvector for vector quantization is set to be the largest integer, L, such that $L2^{Q_{\rm s}^{\star}} \leq 2^{15}$, where $Q_{\rm s}^{\star}$ is the optimal number of shape quantization bits.
- *ScalarQCS:* ScalarQCS is the communication-efficient FL framework developed in [10] when employing the aggregate-and-estimate strategy with $Q_k = 1$, $R_k = 1/C_k$, $G = 25$, and $B = 10$.
- *HighVQ:* HighVQ is the communication-efficient FL framework developed in [5] when $Q_k = C_k$. The dimension of each partition for vector quantization is set to the largest integer, L, such that $L2^{C_k L} \leq 2^{15}$ for device k.
- *D-DSGD:* D-DSGD is the communication-efficient FL framework developed in [6]. The number of the nonzero entries for device k is set to be S_k such that $C_k\bar{N} = \log_2\left(\frac{\bar{N}}{S_k}\right) + 33$.

For the proposed FL framework and ScalarQCS, we set $S_k = \underset{S}{\operatorname{argmax}}\left\{S : R_k < \frac{N}{2K'S\log(N/(K'S))}\right\}$, motivated by a recovery condition in the CS theory [12]. Except for D-DSGD, we normalize the input data values of the training and test data samples using the mean and standard deviation of the MNIST dataset. For D-DSGD, we adjust the input data values of the training and test data samples so that these values are between 0 and 1.

In Fig. 1, we compare the classification accuracy of different FL frameworks in the homogeneous scenario. Fig. 1

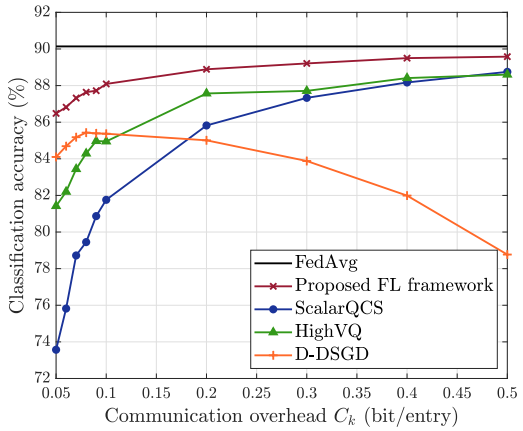

Fig. 1. Classification accuracy vs. communication overhead for different FL frameworks in the homogeneous scenario.

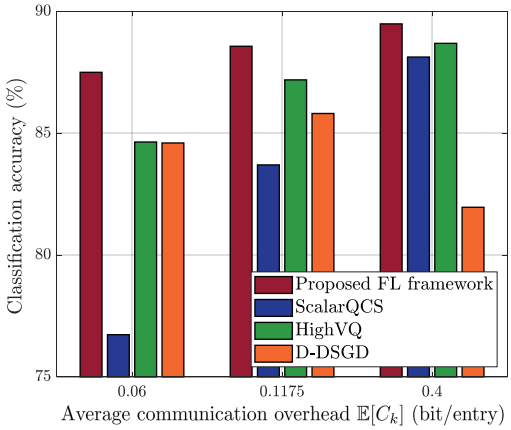

Fig. 2. Classification accuracy of different FL frameworks in the heterogeneous scenario with various average communication overheads.

shows that the proposed FL framework outperforms the existing communication-efficient FL frameworks regardless of the communication overhead allowed for the devices. Further, the proposed FL framework with a 0.5-bit overhead per local model entry shows less than a one percent decrease in classification accuracy compared to FedAvg which may require a 32-bit overhead per local model entry for perfect transmission of the local model updates. This result demonstrates that the proposed FL framework enables an accurate reconstruction of the global model update, while significantly reducing the communication overhead of FL.

In Fig. 2, we compare the classification accuracy of different FL frameworks in the heterogeneous scenario with various average communication overheads. In this simulation, we set $\mathcal{C} = \{0.05, 0.06, 0.07\}$ for $\mathbb{E}[C_k] = 0.06$, $\mathcal{C} = \{0.08, 0.09, 0.1, 0.2\}$ for $\mathbb{E}[C_k] = 0.1175$, and $\mathcal{C} = \{0.3, 0.4, 0.5\}$ for $\mathbb{E}[C_k] = 0.4$. Fig. 2 shows that the proposed FL framework converges to the highest classification accuracy among the communication-efficient FL frameworks, irrespective of the average communication overhead allowed for the devices. In particular, the performance gain of the proposed FL framework over the existing FL frameworks increases as the average communication overhead decreases. This results implies that the proposed FL framework is effective for enabling wireless FL particularly when the dimensionality of the local model updates is much higher than the wireless link capacity.

VI. Conclusion

In this paper, we have presented a novel framework for communication-efficient wireless FL. We have shown that our framework significantly reduces the communication overhead of FL by leveraging both vector quantization and CS-based dimensionality reduction for compressing the local model updates at the wireless devices. Further, we have optimized the design of vector quantizer to minimize the compression error of the local model update. Using the MNIST dataset, we have demonstrated the superiority of the proposed framework over the existing FL frameworks in terms of both communication efficiency and learning accuracy.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Int. Conf. Artif. Intell. Statist.,* Ft. Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[2] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an Intelligent Edge: Wireless Communication Meets Machine Learning," *IEEE Commun. Magazine,* vol. 58, no. 1, pp. 19–25, Jan. 2020.

[3] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed Optimisation for Non-Convex Problems," in *Proc. Int. Conf. Mach. Learn. (ICML),* Stockholm, Sweden, July 2018, pp. 560–569.

[4] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal Vector Quantization for Federated Learning," *IEEE Trans. Signal Process.,* vol. 69, pp. 500–514, Dec. 2021.

[5] Y. Du, S. Yang, and K. Huang, "High-Dimensional Stochastic Gradient Quantization for Communication-Efficient Edge Learning," *IEEE Trans. Signal Process.,* vol. 68, pp. 2128–2142, Mar. 2020.

[6] M. M. Amiri and D. Gündüz, "Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air," *IEEE Trans. Signal Process.,* vol. 68, pp. 2155–2169, Mar. 2020.

[7] Y.-S. Jeon, M. M. Amiri, and N. Lee "Communication-Efficient Federated Learning over MIMO Multiple Access Channels," to be appeared in *IEEE Trans. Commun.,* 2022. [Online] Available: https://arxiv.org/abs/arXiv:2206.05723

[8] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "Communication-Efficient Federated Learning Through 1-bit Compressive Sensing and Analog Aggregation," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops),* Montreal, QC, Canada, June 2021, pp. 1–6.

[9] C. Li, G. Li, and P. K. Varshney, "Communication-Efficient Federated Learning Based on Compressed Sensing," *IEEE Internet Things J.,* vol. 8, no. 20, pp. 15531–15541, Oct. 2021.

[10] Y. Oh, N. Lee, Y.-S. Jeon, and H. V. Poor, "Communication-Efficient Federated Learning via Quantized Compressed Sensing," arXiv:2111.15071 [cs.DC], Nov. 2021. [Online] Available: https://arxiv.org/abs/2111.15071

[11] J. Hamkins and K. Zeger, "Gaussian Source Coding With Spherical Codes," *IEEE Trans. Inf. Theory,* vol. 48, no. 11, pp. 2980–2989, Nov. 2002.

[12] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing,* Boston, MA, USA: Birkhäuser, 2013.

[13] J. P. Vila and P. Schniter, "Expectation-Maximization Gaussian-Mixture Approximate Message Passing," *IEEE Trans. Signal Process.,* vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[14] W. Dai, Y. Liu, and B. Rider, "Quantization Bounds on Grassmann Manifolds and Applications to MIMO Communications," *IEEE Trans. Inf. Theory,* vol. 54, no. 3, pp. 1108–1123, Mar. 2008.

[15] J. Kieffer, "Uniqueness of Locally Optimal Quantizer for Log-Concave Density and Convex Error Weighting Function," *IEEE Trans. Inf. Theory,* vol. 29, no. 1, pp. 42–47, Jan. 1983.

[16] P. F. Panter and W. Dite, "Quantization Distortion in Pulse-Count Modulation With Nonuniform Spacing of Levels," in *Proc. IRE,* vol. 39, no. 1, pp. 44–48, Jan. 1951.