

Article

Towards Efficient Federated Learning: Layer-Wise Pruning-Quantization Scheme and Coding Design

Zheqi Zhu ^{1,2,†}, Yuchen Shi ^{1,2,†}, Gangtao Xin ^{1,2} , Chenghui Peng ³, Pingyi Fan ^{1,2,*}  and Khaled B. Letaief ⁴

¹ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

² Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China

³ Wireless Technology Laboratory, Huawei Technologies, Shanghai 200121, China

⁴ Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong

* Correspondence: fpy@tsinghua.edu.cn; Tel.: +86-010-6279-6973

† These authors contributed equally to this work.

Abstract: As a promising distributed learning paradigm, federated learning (FL) faces the challenge of communication–computation bottlenecks in practical deployments. In this work, we mainly focus on the pruning, quantization, and coding of FL. By adopting a layer-wise operation, we propose an explicit and universal scheme: FedLP-Q (federated learning with layer-wise pruning-quantization). Pruning strategies for homogeneity/heterogeneity scenarios, the stochastic quantization rule, and the corresponding coding scheme were developed. Both theoretical and experimental evaluations suggest that FedLP-Q improves the system efficiency of communication and computation with controllable performance degradation. The key novelty of FedLP-Q is that it serves as a joint pruning-quantization FL framework with layer-wise processing and can easily be applied in practical FL systems.

Keywords: federated learning; model pruning; parameter quantization; code design; layer-wise aggregation; communication-computation efficiency



Citation: Zhu, Z.; Shi, Y.; Xin, G.;

Peng, C.; Fan, P.; Letaief, K.B.

Towards Efficient Federated

Learning: Layer-Wise

Pruning-Quantization Scheme and

Coding Design. *Entropy* **2023**, *25*,

1205. [https://doi.org/10.3390/](https://doi.org/10.3390/e25081205)

e25081205

Academic Editor: Syed A. Jafar

Received: 19 June 2023

Revised: 25 July 2023

Accepted: 11 August 2023

Published: 14 August 2023



Copyright: © 2023 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license ([https://](https://creativecommons.org/licenses/by/4.0/)

[creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)

[4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

1.1. Background and Motivation

With the advent of the big data era, data privacy and system efficiency have attracted great attention, which also brings unprecedented challenges to the field of artificial intelligence (AI). As the first attempt to solve the problem of data silos, federated learning (FL) has now emerged as an important and promising branch of distributed machine learning [1]. Without sharing private data with other clients, FL requires only the interaction of local model parameters for cooperative learning, which significantly improves the security and efficiency of intelligent distributed systems.

The employment of AI in communication networks is already considered as a core feature of 6G systems [2]. In particular, FL has shown its strong potential in combination with deep neural networks [3,4]. As a matter of fact, FL not only naturally adapts to different structures of multi-user networks with distributed data [5], but also enables intelligent collaboration for multi-agent systems [6]. In addition, as a privacy-preserving mechanism that permits unified model training on distributed data from multiple clients, FL can be adopted in numerous fields, such as finance, healthcare, AIoT, smart cities, etc. [7].

The development of this distributed learning paradigm faces multiple challenges. In FL scenarios, clients have greater control over distributed data and local devices. Thus, there exists significant heterogeneity in terms of both data and devices. The former causes a non-iid data distribution, which hinders the convergence and the performance of the global model [8]. The heterogeneity of local devices and settings also leads to strict restrictions on the local computation and transmission, resulting in the straggler effects, where the

systems have to wait for the slowest client [9]. In addition, in some mobile applications with numerous clients, the bandwidth resources are more limited and the communication costs for parameter sharing are much higher [10]. As a result, communication–computation efficiency becomes one of the most significant challenges in the practical deployment of FL systems [11]. An intuitive idea is to deploy models of different complexities on each client, which is called model heterogeneity. However, model heterogeneity violates the original assumption of FL, where the global model is supposed to be identical to all clients. Therefore, the designs of efficient heterogeneous FL systems are still in the early stage.

1.2. Related Work

In order to handle these issues related to system efficiency, two basic solutions have been considered: model pruning and parameter quantization. These were first introduced in centralized machine learning, aiming to accelerate the training process and compress the deep learning models [12]. These two techniques are broadly combined to slim the model and compress the data volume of representation. Inspired by the above two concepts, similar mechanisms have been transferred into the context of FL, which are easy to implement based on the existing FL frameworks. Specifically, pruning performed before local training or uploading, and quantization conducted before model aggregation.

The pruning strategies of FL have been broadly studied in the literature. Caldas et al. proposed the first federated pruning framework in [13], where the global model is compressed into sub-models for local clients. Based on this scheme, the pruning settings and the spectrum allocation for wireless FL were investigated in [14,15]. These methods use random width-wise pruning and have to consider different pruning rules for the inner layers with different structures. These schemes were also challenged in [16], where the authors doubted the rationality of randomly dropping the inner layer parameters. Furthermore, several new pruning mechanisms were also developed. For example, Horvath et al. pointed out that the order of the parameters matters in pruning aggregation and should be taken into consideration [17]. Depth-wise schemes based on blocks and layers were also discussed in [18,19], which showed that model compression in FL should affect the functionality of different parts in local models. The FedLP schemes in [19] specifically provided common pruning solutions for model heterogeneity cases, which extends the FL scenarios. These recent studies suggested that the pruning mechanisms should be designed specially due to the characteristic aggregation operation in FL.

As for quantization, it is commonly treated as an independent procedure from pruning. QSGD, a classic quantization scheme proposed in [20], has been integrated into FL systems [21,22]. Some extensions and variants for communication-efficient scenarios were also developed in [23,24]. The theoretical analysis was studied in [25], and some universal coding schemes were adopted in [26]. In addition, Prakash et al. investigated optimized FL by combining pruning and quantization [27]. However, specific coding schemes that fit the aforementioned pruning and quantization are essentially required. As combined technologies for model compression, pruning-quantization-coding can be jointly and compatibly designed to improve the efficiency for practical FL systems.

1.3. Contribution and Organization

Inspired by the above studies, we mainly consider a compressed FL scheme with joint layer-wise model pruning and parameter quantization in this work, which is an extension of [19]. The main contributions of this work can be summarized as follows:

- We propose FedLP-Q (the codes in this work are available at <https://github.com/Zhuzzq/FedLP/tree/main/FedLP-Q>), a joint pruning-quantization FL framework employing the layer-wise approach. FedLP-Q provides a simple but efficient paradigm to mitigate the communication loads and computation complexity in FL systems;
- We develop two distinct FedLP-Q schemes for homogeneous and heterogeneous cases. The corresponding pruning strategies, quantization rules, and the coding designs are

presented in detail. The theoretical analysis also shows the strengths of the coding scheme for FedLP-Q;

- We carry out several experiments to evaluate the performance of FedLP-Q. The outcomes suggest that this layer-wise pruning-quantization mechanism significantly improves the system efficiency with controllable performance loss.

The remainder of this article is organized as follows. In Section 2, we introduce the preliminaries and the basic concept of layer-wise pruning-quantization. In Section 3, we illustrate the specific design of the proposed approach and its theoretical analysis. The detailed experimental results and more discussions are presented in Section 4. Finally, in Section 5, we conclude and show several potential directions for this work.

2. Preliminaries and FedLP-Q Framework

In this section, we introduce some preliminaries and the framework of FedLP-Q. Before that, for explicitness, we list some basic notations in Table 1.

Table 1. Basic Notations.

Notations	Description
N	The number of distributed clients.
E_l	Local training epoch.
$\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$	Distributed local datasets.
P_t	The set of participation clients at t -th global epoch.
K	The number of participation clients.
$\bar{\theta}$	The global model.
$\{\theta_1, \dots, \theta_N\}$	Local models.
$\{\omega_1, \dots, \omega_N\}$	Aggregation weights for each client.
$\mathbf{g}_{k,t}$	The accumulated gradient of client k in t -th global epoch.

2.1. Basic FL

In a typical FL system with N distributed clients, each client k possesses its own dataset \mathcal{D}_k to train its local model θ_k . To preserve the privacy of personal data, FL conducts local training and periodic model aggregation. Clients cooperate to update the global model $\bar{\theta}$ via parameter sharing, which improves the communication efficiency. As a standard rule, FedAvg [1] selects K clients as a temporary active group P_t and aggregates their local gradients to update the global model. At each global round t , participator clients train their local models for E_l epochs and upload the local gradients to the server. The global model is updated by model aggregation as follows:

$$\bar{\theta}_{t+1} \leftarrow \bar{\theta}_t - \sum_{k \in P_t} \frac{\omega_k}{\sum_{m \in P_t} \omega_m} \cdot \eta_k \mathbf{g}_{k,t}, \quad (1)$$

where $\mathbf{g}_{k,t}$ is the accumulated gradients of client k after E_l epoch local training. $\omega_k = \frac{|\mathcal{D}_k|}{\sum |\mathcal{D}_m|}$ is the aggregation weights and η_k is the learning rate of each client.

2.2. An Overview of the FedLP-Q Framework

As discussed above, the aggregation mechanism of FL makes it possible to conduct layer-wise processes for local models. Furthermore, FL with layer-wise pruning has shown its strengths in several aspects including stability, convergence, and communication-computation efficiency [19]. We generalize such schemes by adding the quantization for layer-wise pruning and formulate the FL framework with layer-wise pruning-quantization, abbreviated as FedLP-Q.

Consider a full model with L layers, $\theta := [\theta^1, \theta^2, \dots, \theta^L]$. After a round of local training on client k , its accumulated gradients can be represented by:

$$\tilde{\mathbf{g}}_k = [\mathbf{g}_k^l], \quad l \in \mathcal{L}_k, \quad (2)$$

where \mathcal{L}_k is the index set of the preserved layers and is determined by the specific layer-wise pruning strategies. The pruned accumulated gradients of each layer are further quantized into $Q(g_k^l)$, where $Q(\cdot)$ denotes the layer-wise quantization function. Hence, the layer-wise aggregation rule shall be modified as:

$$\bar{\theta}_{t+1}^l \leftarrow \bar{\theta}_t^l - \sum_{k \in P_t} \frac{\mathbb{1}_k^l \cdot \omega_k \eta_k}{\sum_{m \in P_t} \mathbb{1}_m^l \cdot \omega_m} Q(g_{k,t}^l), \tag{3}$$

where $\mathbb{1}_k^l$ is the probabilistic indicator function with value $\{0, 1\}$ to represent whether the l -th layer is preserved for the local model on client k . The examples of the indicators will be presented in the following section.

Figure 1 shows the basic frameworks of FedLP-Q for homogeneous and heterogeneous local model scenarios. In the homogeneity scheme, all clients possess the same full layers of the global model. In the heterogeneity scheme, clients initialize their local pruned models with different sequential layers. During the aggregation rounds, clients prune their local models according to layer-wise pruning. Then, the sub-models are quantized into lower bit rates and encoded for model uploading. The parameter server decodes and aggregates the quantized gradients. Similarly, the updated global model will be encoded and transmitted to each local client. In FedLP-Q, since each layer of the model is the basic unit, the pruning and quantization processes can be explicit and model-agnostic. For the above reasons, layer-wise pruning-quantization schemes require no previous knowledge of the model and can easily be deployed in FL systems.

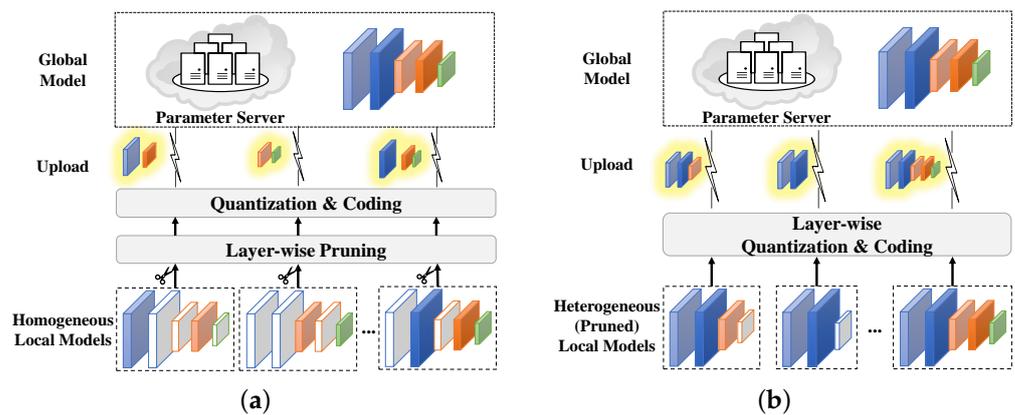


Figure 1. Two typical FedLP-Q schemes for different local model settings. The inactive layers (in gray) are removed from the sub-models for uploading. (a) Homogeneity scheme. (b) Heterogeneity scheme.

3. FedLP-Q: Quantization-Coding Scheme for Layer-Wise Pruning

In this section, we will introduce the proposed layer-wise pruning-quantization approach and the coding scheme in detail. The convergence and the coding performance of FedLP-Q will also be analyzed.

3.1. Pruning Phase

To develop layer-wise pruning for FL, we shall consider the homogeneous and heterogeneous local models. Similar to [19], we adopt the probabilistic layer-wise pruning strategy for homogeneity cases. For heterogeneity cases, we assign the pruned sub-models with different sequential layers to the clients.

3.1.1. Homogeneity Case

As shown in Figure 1a, all clients possess the same full model. While finishing the local training, clients carry out layer-wise pruning for the accumulated gradients. Probabilistic layer-wise pruning proceeds as follows: before uploading the accumulated gradients for

aggregation, each client k preserves the gradients of the l -th layer with probability p_k^l , termed as the layer-preserving rate (LPR). In other words, the indicators in (3) satisfies:

$$\mathbb{1}_k^l = \begin{cases} 1 & \text{with probability } p_k^l, \\ 0 & \text{with probability } 1 - p_k^l. \end{cases} \quad (4)$$

The pruned accumulated gradients \tilde{g}_k in (2) for homogeneity cases can be formulated according to $\{\mathbb{1}_k^l\}$.

3.1.2. Heterogeneity Case

The major challenge of FL with heterogeneous local models is to design the rules for model assignment and aggregation. Fortunately, the layer-wise mechanism of FedLP-Q gives an explicit solution to such problems. That is, each client initializes the local model of different layers. Let L_k denote the layer count (LC). As shown in Figure 1b, its local model shall be initially pruned into $\tilde{\theta}_k = [\theta_k^1, \dots, \theta_k^{L_k}, (\theta_k^O)]$, where θ_k^O (if it exists) is the personalized fully connected (FC) layer to fit the output dimensions and will not be uploaded for aggregation. Then, the pruned accumulated gradients of client k in heterogeneity settings can be expressed as:

$$\tilde{g}_k = [g_k^1, \dots, g_k^{L_k}]. \quad (5)$$

and the layer indicators $\{\mathbb{1}_k^l\}$ can be determined.

In this case, the computation complexity for local training is notably reduced since client k only needs to train a sub-model with L_k layers. As a result, the straggler clients are able to contribute to the global model by training and transmitting small sub-models. This scheme considers the imbalance of the local devices and their computation–communication capabilities, which is more practical in real scenarios.

3.2. Quantization and Coding Phase

Based on layer-wise pruning, we further formulate the corresponding quantization settings and the coding scheme.

3.2.1. Quantization Scheme

For each preserved layer, FedLP-Q conducts the quantization method with equal intervals, which is similar to the quantized-SGD (QSGD) in [20]. Layer-wise quantization after pruning proceeds as follows: for a preserved layer l on client k and the preset quantization bit b , the quantized aggregated gradient $Q(g_k^l; b)$ is defined by:

$$Q(g_k^l; b) = \frac{\|g_k^l\|_2}{2^b} \cdot \text{sgn}(g_k^l) \odot \text{seg}(g_k^l; b), \quad (6)$$

where \odot is the element-wise product, $\text{sgn}(\cdot)$ is the sign function, and $\text{seg}(g_k^l; b)$ is the interval segmentation function. Specifically, for the full gradients g_k^l of the l -th layer, we firstly divide its ℓ_2 -norm into 2^b equal intervals, which takes b bits. Then, each gradient in the layer follows a stochastic quantization: for $g_k^{(l,i)}$, if it locates at the s -th interval, i.e., $|g_k^{(l,i)}| / \|g_k^l\|_2 \in [\frac{s}{2^b}, \frac{s+1}{2^b}]$, the gradient $g_k^{(l,i)}$ will be quantized as the values of two sides according to the distances from the two endpoints of the interval. The quantization can be expressed as:

$$\text{seg}(g_k^{(l,i)}; b) = \begin{cases} s + 1 & \text{with probability } p_Q^{(l,i)}(s; b), \\ s & \text{otherwise.} \end{cases} \quad (7)$$

Therein, the stochastic quantization probability is defined as:

$$p_Q^{(l,i)}(s; b) = \frac{|g_k^{(l,i)}|}{\|g_k^l\|_2} \cdot 2^b - s, \tag{8}$$

which is the ratio between the distance to the left endpoint and the interval length.

3.2.2. Coding Design

Given the quantization bit b and the layer 2-norm $\|g_k^l\|_2$, the quantized value of each gradient can be handled as the index of the interval. Then, the coding of the quantized gradients can be considered as the representation of the integers. Therefore, we adopt a classic solution: recursive Elias coding, which is also known as Elias omega coding [28]. The core idea of Elias omega coding is to recursively encode the prefix of the binary representation. Hence, the code length of any positive integer N follows $|\text{Elias}(N)| = 1 + \log N + \log \log N + \dots = 1 + (1 + o(1)) \log N$. Then, we shall consider the simple encoding scheme for layer-wise pruning-quantization. The full coding scheme for FedLP-Q proceeds as follows: For each preserved layer l , the layer 2-norm will firstly be encoded into 32 bits (float). The quantization setting b (integer) will be attached using Elias omega coding. Then, for each gradient, the index of interval $\text{seg}_k^l(i)$ for stochastic quantization will be sequentially encoded according to the Elias omega coding. The sign of each gradient will additionally take 1 bit (a boolean variable standing for positive/negative). The decoding scheme is also simple: The decoder firstly reads off the 32 bits to obtain the layer 2-norm $\|g_k^l\|_2$ and decodes the Elias codes of b . Then, for each gradient, its interval index will be reconstructed according to Elias omega decoding, while the sign will be decoded by the last bit.

After layer-wise pruning-quantization, the layer gradients can be represented by a 4-element tuple $\{\|g_k^l\|_2, b, \text{seg}(g_k^l; b), \text{sgn}(g_k^l)\}$. Therefore, based on Elias omega coding, the code structure of each layer with M_l parameters is shown in Figure 2. Furthermore, the corresponding encoding and decoding procedures can be developed as shown in Algorithm 1 and Algorithm 2, respectively.

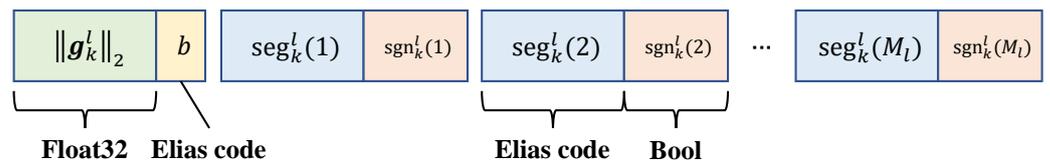


Figure 2. Code structure of each layer’s gradients under the FedLP-Q scheme.

Algorithm 1 Encoder for layer l

Input: $\{\|g_k^l\|_2, b, \text{seg}(g_k^l; b), \text{sgn}(g_k^l)\}, M_l;$

- 1: $C_l \leftarrow \text{Concat}(\text{Float32}(\|g_k^l\|_2), \text{Elias_Encoder}(b));$ ▷ encode the header
- 2: **for** $i \leftarrow 1$ **to** M_l **do**
- 3: $C_l \leftarrow \text{Concat}(C_l, \text{Elias_Encoder}(\text{seg}(g_k^{(l,i)}; b)));$ ▷ encode the interval index
- 4: $C_l \leftarrow \text{Concat}(C_l, \text{Binary}(\text{sgn}(g_k^{(l,i)}; b)));$ ▷ encode the gradient sign
- 5: **end for**

Output: encoded message C_l .

Algorithm 2 Decoder for layer l

Input: received message \mathcal{C}_l ; **Initialization:** $\text{seg}^l \leftarrow [], \text{sgn}^l \leftarrow [], i \leftarrow 0$;

- 1: $G_l \leftarrow \text{Float32_Decoder}(\mathcal{C}_l)$; ▷ decode the gradient norm
- 2: $\mathcal{C}_l \gg 32$;
- 3: $(b, l_E) \leftarrow \text{Elias_Decoder}(\mathcal{C}_l)$; ▷ decode the quantization bit
- 4: $\mathcal{C}_l \gg l_E$;
- 5: **while** \mathcal{C}_l **do**
- 6: $(\text{seg}_i^l, l_E) \leftarrow \text{Elias_Decoder}(\mathcal{C}_l)$; ▷ decode the interval index
- 7: $\mathcal{C}_l \gg l_E$;
- 8: $(\text{sgn}_i^l, l_E) \leftarrow \mathcal{C}_l$; ▷ decode the gradient sign
- 9: $i \leftarrow i + 1; \mathcal{C}_l \gg 1$;
- 10: **end while**

Output: decoded elements $\{G_l, b, \text{seg}^l, \text{sgn}^l\}$.

3.2.3. Algorithm Formulation

Finally, we formulate the procedures of FedLP-Q as in Algorithm 3. The main changes from the original FL include (1) layer-wise pruning (for both the homogeneity and heterogeneity case) for the local gradients is conducted in Line 5; (2) stochastic quantization and the corresponding encoding/decoding are performed in Lines 6 and 7; and (3) Line 9 introduces the layer-wise aggregation for the proposed pruning-quantization schemes.

Algorithm 3 FedLP-Q

Initialization: local models, pruning configures, etc.

- 1: **for** $t \leftarrow 1$ **to** max_epoch **do**
- 2: **for** participator client k in parallel **do** ▷ client side
- 3: Update $\theta_{k,t} \leftarrow \text{Local_Train}(\theta_{k,t-1}; \mathcal{D}_k)$;
- 4: Calculate the accumulated gradients $\mathbf{g}_{k,t}$;
- 5: Conduct layer-wise pruning: $\tilde{\mathbf{g}}_{k,t}$;
- 6: Quantize the local gradients: $Q(\tilde{\mathbf{g}}_{k,t})$;
- 7: Carry out Algorithm 1 to encode the quantized gradients;
- 8: Upload local message to parameter server;
- 9: **end for**
- 10: Carry out Algorithm 2 to decode the received messages; ▷ server side
- 11: Aggregate each layer $\bar{\theta}_t^l$ by (3);
- 12: Download the global model: $\theta_{k,t} \leftarrow \bar{\theta}_t$; ▷ client side
- 13: **end for**

Output: global model: $\bar{\theta}_t$.

3.3. Theoretical Analysis

Then, we theoretically analyzed the convergence and coding performance of FedLP-Q.

3.3.1. Impacts on Convergence

Since FedLP-Q modifies the aggregation process of the original FL, the main impacts on convergence come from layer gradient pruning and quantization. Assuming that local training is independent from the pruning operations, the following proposition indicates the convergence result of FedLP-Q.

Proposition 1. Consider a fairness case where $\omega_k = 1/K$ and $p_k^l = p$ for all $k = 1, \dots, K$. FedLP-Q leads to $(1-p)^K$ convergence rate decay compared to the original FL scheme, i.e., the expectation of the aggregated gradient in FedLP-Q satisfies:

$$\mathbb{E}[\hat{\mathbf{g}}^l | \bar{\mathbf{g}}^l] = [1 - (1-p)^K] \cdot \bar{\mathbf{g}}^l. \quad (9)$$

Therein,

$$\bar{\mathbf{g}}^l = \frac{1}{K} \sum_k \mathbf{g}_k^l, \quad \hat{\mathbf{g}}^l = \sum_k \frac{\mathbb{1}_k^l}{\sum_m \mathbb{1}_m^l} Q(\mathbf{g}_k^l) \tag{10}$$

denote the aggregated gradients in the original FL and FedLP-Q, respectively.

Proof. For simplicity, rewrite the client index in P_t as $\{1, 2, \dots, K\}$. According to the layer-wise aggregation in (3), the expectation of the pruned-quantized gradients can be obtained by:

$$\mathbb{E}[\hat{\mathbf{g}}^l | \bar{\mathbf{g}}^l] = \mathbb{E}\left[\sum_{k=1}^K \frac{\mathbb{1}_k^l}{\sum_m \mathbb{1}_m^l} Q(\mathbf{g}_k^l) \middle| \bar{\mathbf{g}}^l\right] = \sum_{k=1}^K \mathbb{E}\left[\frac{\mathbb{1}_k^l}{\sum_m \mathbb{1}_m^l} Q(\mathbf{g}_k^l) \middle| \bar{\mathbf{g}}^l\right] \tag{11}$$

$$\stackrel{\textcircled{1}}{=} \sum_{k=1}^K \mathbb{E}\left\{\mathbb{E}\left[\frac{\mathbb{1}_k^l}{\sum_m \mathbb{1}_m^l} Q(\mathbf{g}_k^l) \middle| \bar{\mathbf{g}}^l, \mathbb{1}_k^l\right] \middle| \bar{\mathbf{g}}^l\right\} \tag{12}$$

$$= \sum_{k=1}^K p \cdot \mathbb{E}\left[\frac{1}{1 + \sum_{m \neq k} \mathbb{1}_m^l} \cdot Q(\mathbf{g}_k^l) \middle| \bar{\mathbf{g}}^l\right] \tag{13}$$

$$= \sum_{k=1}^K p \left[\sum_{m=1}^K \frac{1}{m} \binom{K-1}{m-1} p^{m-1} (1-p)^{K-m} \right] \mathbb{E}[Q(\mathbf{g}_k^l) | \bar{\mathbf{g}}^l] \tag{14}$$

$$\stackrel{\textcircled{2}}{=} \sum_{k=1}^K \left[\sum_{m=1}^K \frac{1}{K} \binom{K}{m} p^m (1-p)^{K-m} \right] \mathbb{E}[Q(\mathbf{g}_k^l) | \bar{\mathbf{g}}^l] \tag{15}$$

$$= [1 - (1-p)^K] \mathbb{E}\left[\sum_{k=1}^K \frac{Q(\mathbf{g}_k^l)}{K} \middle| \bar{\mathbf{g}}^l\right] \tag{16}$$

$$\stackrel{\textcircled{3}}{=} [1 - (1-p)^K] \cdot \bar{\mathbf{g}}^l, \tag{17}$$

where $\textcircled{1}$ is the result of equation $\mathbb{E}[X|Y] = \mathbb{E}\{\mathbb{E}[X|Y, Z]|Y\}$, $\textcircled{2}$ holds because $\frac{1}{m} \binom{K-1}{m-1} = \frac{(K-1)!}{(K-m)!m!} = \frac{1}{K} \binom{K}{m}$, and $\textcircled{3}$ is the unbiased quantization property of QSGD-based quantizers [22,25]. Then, the proposition is proved. \square

Remark 1. (9) shows that the aggregated gradients of FedLP-Q decreases compared to the original FL schemes, which implies $(1-p)^K$ convergence rate decay. In addition, the conclusion of Proposition 1 can be directly adopted in the assumptions of model pruning/quantization (e.g., [22,24]) and the specific convergence theorems can be further modified.

According to the proposition, such impacts on convergence can be mitigated by increasing either the layer preserving rate p or the number of participant clients K .

3.3.2. Coding Performance

Based on the stochastic quantization method and the corresponding coding scheme, we obtained the following proposition, which highlights the communication costs of FedLP-Q.

Proposition 2 (Code length). *By employing layer-wise pruning-quantization, the average bit number to communicate \mathbf{g}_k^l for client k is upper-bounded by:*

$$\hat{\mathcal{C}}(\mathbf{g}_k^l; b) = 33 + (1 + o(1)) \log b + M_l [2 + (1 + o(1))b]. \tag{18}$$

Proof. The proposition is an intuitive result of the proposed quantization coding scheme. As discussed above, the code length of an integer N using Elias omega coding satisfies $|\text{Elias}(N)| = 1 + \log N + \log \log N + \dots = 1 + (1 + o(1)) \log N$. Consider the code blocks of the l -th layer as shown in Figure 2, the code length can be divided into three parts: (1)

The first block to represent the layer 2-norm $\|g_k^l\|_2$ takes 32 bits; (2) the Elias codes of the quantization level b takes $1 + (1 + o(1)) \log b$ bits; and (3) after stochastic quantization, the interval index of each parameter is an integer less than 2^b , which takes $1 + (1 + o(1))b$ bits, and its sign takes one more bit. Hence, the total code length can be upper bounded as shown in (18). \square

Remark 2. According to [20], the original coding scheme proposed in QSGD for the l -th layer leads to the average code length asymptotically bounded by:

$$32 + \left[3 + (1.5 + o(1)) \log \frac{2(S^2 + M_l)}{S(S + \sqrt{M_l})} \right] S(S + \sqrt{M_l}), \tag{19}$$

where $S = 2^b$ is the count of the quantization intervals. Comparing the two upper bounds of encoding the layer-wise gradients in (18) and (19), one can find that the proposed coding scheme for FedLP-Q has a shorter code length than that in the original QSGD when the number of parameters is not extremely large.

Specifically, the Elias coding approach for QSGD considers the positional coding of the non-zero gradient to fit the cases with sparse and numerous parameters. On the contrary, in FedLP-Q, the gradients of each layer are handled independently. Therefore, the sparsity and the element amounts are not enough to show the advantages of the Elias coding scheme in QSGD. The experimental results in the next section also indicate that the quantization coding for FedLP-Q can be simple but more efficient. Overall, by employing layer-wise pruning and quantization, FedLP-Q improves the communication-computation efficiency in FL, and also simplifies the code design for model uploading.

Furthermore, we discuss the code efficiency. After layer-wise pruning-quantization, the gradients of $Q(g_k^l)$ can be represented by a set of interval indexes. Then, the entropy rate of the layer gradients can be defined as follows:

Definition 1. After the b -bit quantization, the entropy rate of the layer gradients g_k^l is defined as:

$$\mathcal{H}(Q(g_k^l); b) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \mathcal{H}(\text{seg}_k^l(i)); \tag{20}$$

where $\text{seg}(\cdot)$ is the quantization index. Consider the independence of each gradient and the empirical distribution, the above entropy rate can be approximated by:

$$\hat{\mathcal{H}}(Q(g_k^l); b) \approx \hat{\mathcal{H}}(\text{seg}_k^l) := - \sum_{s \in \{\text{seg}_k^l\}} f(s) \log(f(s)), \tag{21}$$

where $f(s) = \frac{\text{count}(s)}{M_l}$ is the proportion of index s in seg_k^l .

Then, the code efficiency can be investigated by comparing the per-gradient code length and the entropy rate. According to Proposition 2, the per-gradient code length can be expressed asymptotically by:

$$\frac{\hat{C}(g_k^l; b)}{M_l} \approx 2 + (1 + o(1))b. \tag{22}$$

Hence, the aim is to make the asymptotic per-gradient code length as close as possible to the entropy rate, i.e., $\frac{\hat{C}(g_k^l; b)}{M_l} \rightarrow \hat{\mathcal{H}}(Q(g_k^l); b)$. Moreover, consider the whole procedure of FedLP-Q, the equivalent per-gradient code length $p_k^l \frac{\hat{C}(g_k^l; b)}{M_l}$ can be even smaller than the entropy rate $\hat{\mathcal{H}}(Q(g_k^l); b)$, where p_k^l is the preserving proportion of the gradients of the

l -th layer. In the next section, we will show that the proposed FedLP-Q can achieve the above goals through experiments.

4. Experimental Results

In this section, we describe several experiments to evaluate the performance of the FedLP-Q schemes.

We carried out the experiments on an image classification FL task on CIFAR-10 with 100 clients and a 0.1 participation rate. Specifically, the parameter server randomly selected 10 clients for aggregation in every global round. The number of the local training epoch was fixed at 5. We adopted the global model as a CNN-based neural network with six Conv layers and two FC layers. Batch normalization and maxpooling were also conducted following each Conv layer. As for the two proposed schemes, we adopted the consistent LPR p for the homogeneous case, abbreviated as FedLP-Q_Homo(p). For the heterogeneous case, we separated the Conv layers into five ordered sequences with LC from 1 to 5. These five sub-models were assigned to clients according to LC distributions. FedLP-Q_Hetero(l) means that the model with l LC was assigned with the highest probability and the parameter ‘u’ represents the uniform assignment.

We evaluated the performance of the proposed FedLP-Q schemes under both iid and Dirichlet non-iid data ($\alpha = 1$) settings. As shown in Figure 3, the FedLP-Q schemes reached a similar accuracy and convergence as the original FedAvg. In particular, the homogeneous case with high quantization bits ($b = 10$) led to nearly the same test accuracy curves as that of FedAvg, but significantly reduced the communication traffic. Small quantization bits and the non-iid data settings led to a lower accuracy and the instability of the global model. Therefore, there exists redundancy in the transmitted gradients and the quantization representation. By setting different FedLP-Q configures, FL systems can reduce the requirement of the communication rate and computation capability, which enables more devices to participate in the FL tasks.

The detailed numerical results for the test accuracy and the communication–computation efficiency are listed in Table 2. The savings of communication and computation are represented by the decay ratios of the parameter count and the million floating point of operations (MFLOPs) per local model, respectively. One can find that, for homogeneity cases, FedLP-Q performs better on both data settings. The randomness caused by the probabilistic layer-wise pruning and stochastic quantization even benefit the model performance. In addition, the number of parameters can be reduced, which relieves the communication loads. For heterogeneity cases, though there exists degradation of the test accuracy, FedLP-Q can significantly reduce both the communication and computation costs with acceptable model performance. This is an important characteristic, especially for mobile scenarios where the clients’ capabilities of transmission and local computation are limited.

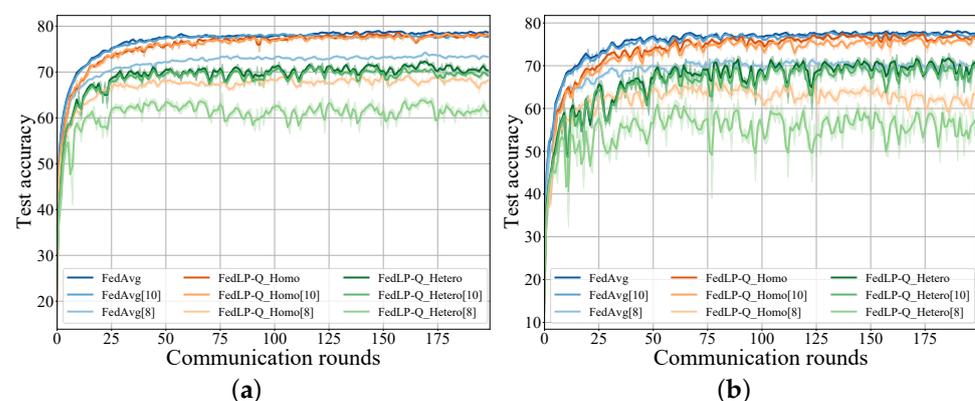
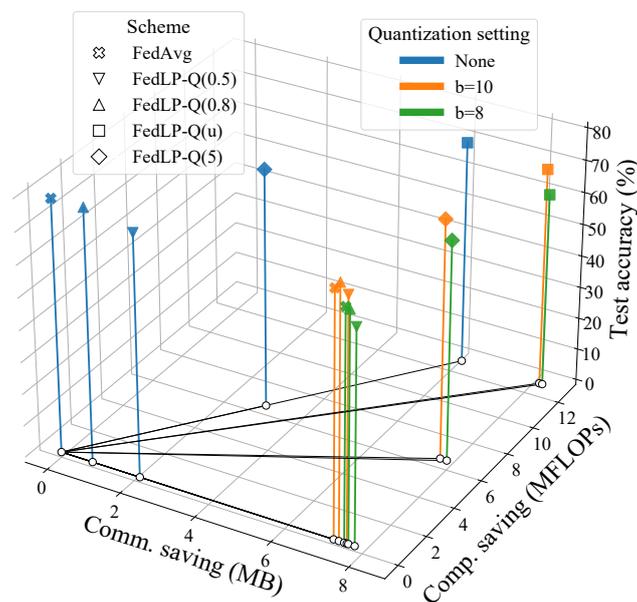


Figure 3. Test on CIFAR-10. FedLP-Q_Homo(0.5) and FedLP-Q_Hetero(u). (a) Under iid data. (b) Under non-iid data.

Table 2. Comparisons on accuracy, communication, and computation saving.

Schemes	Test Accuracy % (iid/Non-iid)			#Param ↘	MFLOPs ↘
	No Quant	b = 10	b = 8		
FedAvg	78.8/75.3	76.2/74.9	71.7/68.5	/	/
FedLP-Q(0.2)	73.4/68.2	71.5/66.8	56.6/50.4	40.0%	/
FedLP-Q(0.5)	75.5/73.5	75.4/72.5	66.7/62.0	25.0%	/
FedLP-Q(0.8)	78.9 / 77.3	78.3/76.2	71.3/66.9	10.0%	/
FedLP-Q(1)	64.4/60.3	64.1/59.2	53.6/49.4	84.6%	51.2%
FedLP-Q(3)	66.8/62.0	66.1/61.7	58.4/53.9	79.6%	32.3%
FedLP-Q(u)	70.1/66.5	68.3/65.4	60.7/55.7	71.1%	34.5%
FedLP-Q(5)	74.6/73.1	74.4/72.1	68.6/64.5	35.6%	17.2%

Figure 4 displays the model performance, communication traffic, and computation load savings of several schemes, including FedAvg, homogeneous FedLP-Q, and heterogeneous FedLP-Q, with different quantization settings. The higher vertical axis represents a higher model accuracy. The x-axis denotes the reduced communication traffic (MB) and the y-axis is the saved local computation complexity (MFLOPs). The horizontal plane implies the communication and computation efficiency. If a projection on the x-y plane lies farther from the zero point, the scheme saves more communication and computation loads. By plotting the 3D figure, the trade-offs between the model performance and system efficiency can be intuitively shown. According to this trade-off, the corresponding system design and the pruning-quantization settings can be well guided.

**Figure 4.** Trade-off: accuracy vs. communication–computation savings (under iid data).

We next evaluated the efficiency of the coding scheme for FedLP-Q. Figure 5a shows how the code length evolves as the number of the parameters increases. We compared the expected code length of the proposed Elias coding for FedLP-Q (as in (18)), the Elias coding for QSGD (as in (19)), and the non-quantization coding (Float32). For a parameter count of less than 600 K, the Elias-based coding scheme proposed in Section 3.2.2 led to the lowest expected code length. The superiority of the QSGD coding scheme in [20] appeared for an extremely large parameter count. Notably, since we adopted a layer-wise approach, the parameter count of each layer commonly fell within the red region marked in Figure 5b. As a result, the proposed coding scheme is more suitable for FedLP-Q. Figure 5b presents the practical code length of the neural network model employed in the experiments. By

implementing the Elias code, FedLP-Q encodes the accumulated gradients into a lower bit rate, which reduces the requirement for the communication rate in real-world systems.

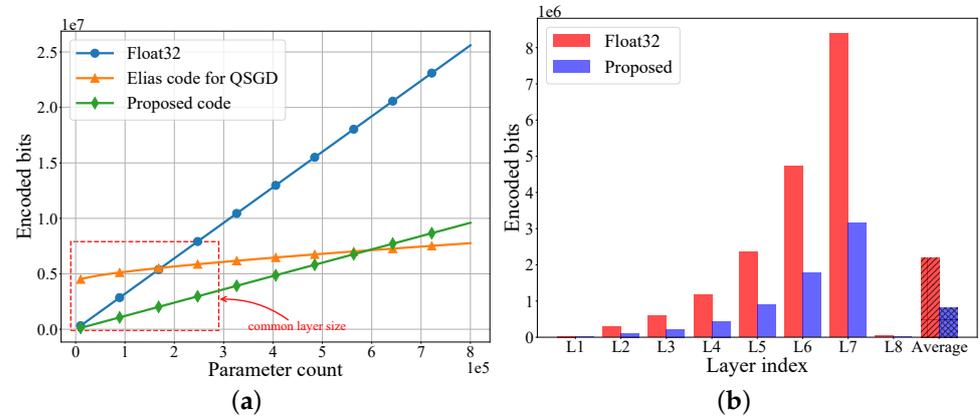


Figure 5. The comparison of coding schemes of model uploading ($b = 10$). (a) Encoded bits vs. parameters. (b) Code length of each layer.

Finally, we investigated the gap between the per-gradient length of the proposed schemes and the entropy rate of the gradients. Figure 6 displays the quantized gradient distribution of the selected clients under iid and non-iid data settings. The quantized gradient distributions of different layers have similar shapes under both iid and non-iid settings. This again demonstrates the potential and future directions of introducing layer-wise operation into FL. That is, the distribution and the entropy of each layer gradients can be modeled. Then, the communication traffic of each client can be better estimated and further coding schemes can be specifically designed.

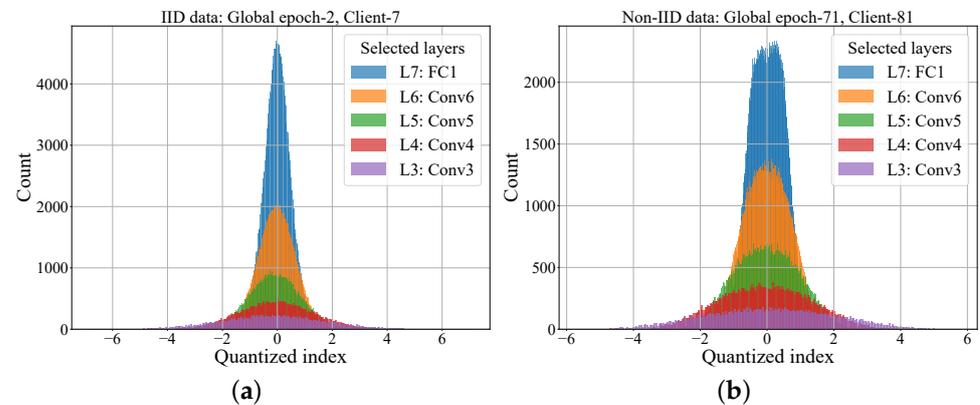


Figure 6. The quantized gradient distribution of selected clients. (a) Under iid data, 2nd global epoch, client 7. (b) Under non-iid data, 71st global epoch, client 81.

Figure 7 shows the entropy rate $\hat{H}(Q(g_k^l); b)$ and the equivalent per-gradient code length $p_k^l \frac{\hat{C}(g_k^l; b)}{M_l}$ of each layer. Herein, the entropy rate of the gradients can be calculated through (21) based on the distributions shown in Figure 6. It is intuitive that the per-gradient length of the proposed schemes is close to the entropy rate of the gradients. In particular, as discussed in Section 3.3.2, with a lower LPR, the expectation of the per-gradient code length can take even fewer bits than the entropy rate, e.g., as shown by the curves of $p_k^l = 0.7$. This indicates that the proposed coding scheme is efficient and can be adjustable to meet different system requirements on compression.

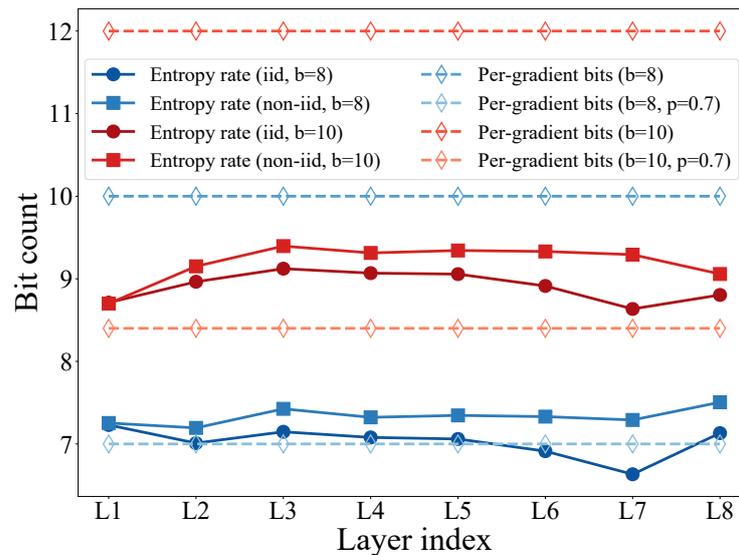


Figure 7. The entropy rate $\hat{\mathcal{H}}(Q(g_k^l); b)$ and the equivalent per-gradient code length $p_k^l \frac{\mathcal{C}(g_k^l; b)}{M_l}$ of each layer.

All the above results suggest that there exists a significant redundancy in the interaction of FL systems. Layer-wise pruning-quantization and the proposed coding design can be applied without damaging the model performance. Therefore, it can be concluded that FedLP-Q is an efficient scheme that improves the system efficiency and facilitates the deployment of FL applications by mitigating the straggler effects.

5. Conclusions

In this article, we proposed FedLP-Q, an efficient pruning-quantization approach for FL by adopting layer-wise operations. The corresponding pruning strategies, stochastic quantization, and coding scheme were formulated. Both the theoretical and experimental analyses indicated that FedLP-Q achieves a better communication–computation efficiency with controllable performance loss. This may be vital for the deployment of FL applications in mobile scenarios.

This work also opens up several potential research directions for future work, including exploring the robustness and security of layer-wise pruning-quantization, combinations with other FL structures, quantization/coding designs based on layer-gradient entropy estimation, and more practical modeling for the communication of FL clients.

Author Contributions: Conceptualization, P.F., Z.Z. and Y.S.; methodology, Z.Z., Y.S., G.X. and P.F.; software, Y.S. and Z.Z.; investigation, Z.Z., Y.S. and G.X.; writing—original draft preparation, Z.Z. and Y.S.; writing—review and editing, P.F., Z.Z., Y.S. and G.X.; visualization, Z.Z. and Y.S.; supervision, P.F., C.P. and K.B.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work of P.F. is supported in part by the National Key Research and Development Program of China (Grant No. 2021YFA1000500(4)). K.B.L. is supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FL	Federated Learning
iid	Independent and Identically Distributed
non-iid	Non-Independent and Non-Identically Distributed
FedAvg	Federated Averaging
QSGD	Quantized Stochastic Gradient Descent
FedLP	Federated Learning with Layer-wise Pruning
FedLP-Q	Federated Learning with Layer-wise Pruning and Quantization
Conv	Convolutional
FC	Fully Connected
FLOPs	Floating Point Operations

References

- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
- Letaief, K.B.; Chen, W.; Shi, Y.; Zhang, J.; Zhang, Y.J.A. The roadmap to 6G: AI empowered wireless networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [\[CrossRef\]](#)
- Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 5–36. [\[CrossRef\]](#)
- Yang, Z.; Chen, M.; Wong, K.K.; Poor, H.V.; Cui, S. Federated learning for 6G: Applications, challenges, and opportunities. *Engineering* **2022**, *8*, 33–41. [\[CrossRef\]](#)
- Wan, S.; Lu, J.; Fan, P.; Shao, Y.; Peng, C.; Chuai, J. How global observation works in Federated Learning: Integrating vertical training into Horizontal Federated Learning. *IEEE Internet Things J.* **2023**, *10*, 9482–9497. [\[CrossRef\]](#)
- Zhu, Z.; Wan, S.; Fan, P.; Letaief, K.B. Federated multiagent actor–critic learning for age sensitive mobile-edge computing. *IEEE Internet Things J.* **2021**, *9*, 1053–1067. [\[CrossRef\]](#)
- Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [\[CrossRef\]](#)
- Zhu, Z.; Fan, P.; Peng, C.; Letaief, K.B. ISFL: Federated Learning for Non-i.i.d. Data with Local Importance Sampling. *arXiv* **2022**, arXiv:2210.02119v2.
- Chen, M.; Mao, B.; Ma, T. Efficient and robust asynchronous federated learning with stragglers. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
- Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60.
- Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [\[CrossRef\]](#)
- Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
- Caldas, S.; Konečný, J.; McMahan, H.B.; Talwalkar, A. Expanding the reach of federated learning by reducing client resource requirements. *arXiv* **2018**, arXiv:1812.07210.
- Liu, S.; Yu, G.; Yin, R.; Yuan, J. Adaptive network pruning for wireless federated learning. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1572–1576. [\[CrossRef\]](#)
- Wen, D.; Jeon, K.J.; Huang, K. Federated Dropout—A Simple Approach for Enabling Federated Learning on Resource Constrained Devices. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 923–927. [\[CrossRef\]](#)
- Cheng, G.; Charles, Z.; Garrett, Z.; Rush, K. Does Federated Dropout Actually Work? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3387–3395.
- Horvath, S.; Laskaridis, S.; Almeida, M.; Leontiadis, I.; Venieris, S.; Lane, N. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12876–12889.
- Chen, Y.; Chen, Z.; Wu, P.; Yu, H. FedOBD: Opportunistic Block Dropout for Efficiently Training Large-scale Neural Networks through Federated Learning. *arXiv* **2022**, arXiv:2208.05174.
- Zhu, Z.; Shi, Y.; Luo, J.; Wang, F.; Peng, C.; Fan, P.; Letaief, K.B. FedLP: Layer-wise Pruning Mechanism for Communication-Computation Efficient Federated Learning. *arXiv* **2023**, arXiv:2303.06360.
- Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1707–1718.
- Hönig, R.; Zhao, Y.; Mullins, R. DAdaQuant: Doubly-adaptive quantization for communication-efficient Federated Learning. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 8852–8866.

22. Liu, L.; Zhang, J.; Song, S.; Letaief, K.B. Hierarchical federated learning with quantization: Convergence analysis and system design. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 2–18. [[CrossRef](#)]
23. Amiri, M.M.; Gunduz, D.; Kulkarni, S.R.; Poor, H.V. Federated learning with quantized global model updates. *arXiv* **2020**, arXiv:2006.10672.
24. Shlezinger, N.; Chen, M.; Eldar, Y.C.; Poor, H.V.; Cui, S. UVeQFed: Universal vector quantization for federated learning. *IEEE Trans. Signal Process.* **2020**, *69*, 500–514. [[CrossRef](#)]
25. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Online, 26–28 August 2020; pp. 2021–2031.
26. Yue, K.; Jin, R.; Wong, C.W.; Dai, H. Communication-efficient federated learning via predictive coding. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 369–380. [[CrossRef](#)]
27. Prakash, P.; Ding, J.; Chen, R.; Qin, X.; Shu, M.; Cui, Q.; Guo, Y.; Pan, M. IoT Device Friendly and Communication-Efficient Federated Learning via Joint Model Pruning and Quantization. *IEEE Internet Things J.* **2022**, *9*, 13638–13650. [[CrossRef](#)]
28. Elias, P. Universal codeword sets and representations of the integers. *IEEE Trans. Inf. Theory* **1975**, *21*, 194–203. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.