Dual Adaptive Compression for Efficient Communication in Heterogeneous Federated Learning

***Abstract*—In federated learning, multiple rounds of communication are involved between clients and the server to train a global model. The extensive model updates transmitted during the training lead to significant communication costs. Previous methods usually employ quantization or sparsification to compress model updates. However, the lossy compression leads to a decline in accuracy, it is challenging to strike a balance between communication efficiency and model accuracy. Meanwhile, due to the data heterogeneity, local updates among different clients are biased towards each other. Employing the same compression ratios for each local updates will further degrade the model accuracy. To achieve the trade-off between communication efficiency and model accuracy, we propose FedDAC, a Dual Adaptive Compression method in heterogeneous federated learning. In the local computation phase, the loss queue is adopted to detect the convergence trends within each client. FedDAC can then dynamically quantify model updates and allow for various compression ratios among heterogeneous clients. In the global aggregation phase, FedDAC can determine the fluctuations in training based on the similarity between clients and the server, thereby adjusting the sparsity ratio flexibly. To alleviate the reduction in model accuracy caused by lossy compression, we introduce residual updates in the local computation and global aggregation phases to maintain model accuracy. Experiment results show that compared with one-way compression methods NAGC and AdaQuantFL, FedDAC can maintain comparable accuracy while the accumulated communication volume is reduced by about 29.6 times, and 22.8 times, respectively. Moreover, the global model accuracy of FedDAC surpasses the two-way compression method T-FedAvg by about 2.4%, and the accumulated communication volume is about 2.5 times lower than T-FedAvg.**

***Index Terms*—federated learning, communication efficiency, data heterogeneity, adaptive compression.**

I. INTRODUCTION

With the rapid development of technologies such as 5G, the Internet of Things (IoT) is gradually being applied in real-world applications [1]. Based on the collected data and computational power of IoT edge devices (smartphones [2], wireless cameras [3], drones [4], IoT sensors [5], etc.), deep neural networks can be trained in a distributed manner. In order to protect data privacy, federated learning [6] is usually utilized to organize edge devices for distributed training. In federated learning, the model updates transmitted between clients and the server lead to significant communication costs, especially when it involves large-scale models. Exchanging model updates with a large number of parameters in each round of communication will put great pressure on the network channel. It also exacerbates the waiting latency of the training process and results in the underutilization of edge devices.

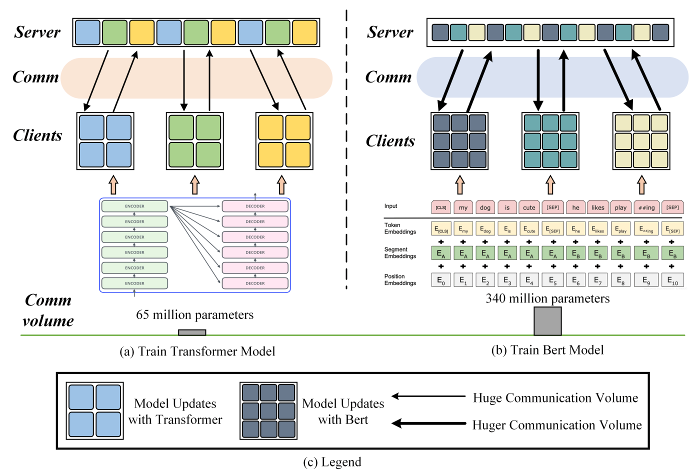

Fig. 1. The extensive single-round communication volume in federated learning

Fig. 1 illustrates the extensive single-round communication volume in federated learning when training the Transformer [7] and Bert [8], respectively. Assuming that each parameter is represented in 64-bit and the network bandwidth is 100 Mbps. The Transformer contains 65 million parameters, the client-server communication volume is $65million{*}64bit{*}2 = 991.8Mb$ in each round of training, and the transmission time is $991.8Mb/100Mbps \approx 10s$. The number of parameters in the Bert is about 5 times more than that of the Transformer, resulting in unacceptable transmission delay. For this reason, it is necessary to reduce the single-round communication volume thus optimizing the communication efficiency in federated learning.

Related studies compress the single-round communication volume to reduce the high communication costs. The communication volume refers to the model updates transmitted between clients and the server. As typical model compression techniques, quantization [9] and sparsification [10] are commonly utilized in distributed machine learning for efficient communication. However, quantization and sparsification cannot be directly employed in federated learning. On the one hand, lossy compression in quantization or sparsification leads to a decline in model accuracy, it is challenging to strike a balance between communication efficiency and model accuracy. On the other hand, in heterogeneous federated learning, employing the same and fixed quantization or sparsification coefficients for all clients with different data distributions will exacerbate gradient conflict [11] and gradient drift [12], leading to degradation of global model accuracy.

In order to deploy suitable compression methods for heterogeneous federated learning, adaptive compression is introduced by related studies. For example, NAGC [13] defines loss queues for all clients to save the historical loss during local training. With the perception of the training state, NAGC can dynamically adjust the compression ratio to achieve adaptive sparsification. To alleviate the high error floor caused by quantization, AdaQuantFL [14] utilizes the initial loss and the current loss to dynamically determine the quantization coefficient. Furthermore, it introduces the variable learning rate to mitigate the excessive change in quantization coefficients. The above methods adjust the compression ratio to improve the global model accuracy while optimizing communication efficiency. However, these methods only consider the compression in the upstream communication phase, and the communication efficiency needs to be further optimized.

To reduce both the upstream and downstream single-round communication volume, Sattler et al. [15] propose a sparse ternary compression method STC. Combined with sparsification and ternary quantization, STC can greatly compress the model updates and reduce communication costs in federated learning. Drawing on Sattler's work, Xu et al. [16] propose a ternary compression method T-FedAvg. T-FedAvg quantizes the local updates and sparsifies global updates during training, achieving bidirectional compression for upstream and downstream communication. However, both STC and T-FedAvg adopt a fixed compression ratio in compressing the single-round communication volume, which does not take into account the difference in data distribution among clients and cannot be directly employed in heterogeneous federated learning.

To this end, this paper considers compressing each of the upstream and downstream model updates, while achieving the trade-off between communication efficiency and model accuracy in heterogeneous federated learning. A Dual Adaptive Compression method (FedDAC) is proposed in this paper. Specifically, in the local computation phase, FedDAC introduces the loss queue to dynamically quantify model updates among heterogeneous clients. In the global aggregation phase, FedDAC adjusts the sparsity ratio based on the similarity between the local and global updates. Furthermore, we employ local and global residual updates to improve the model accuracy impaired by lossy compression. Generally speaking, FedDAC can reduce communication costs while maintaining global model accuracy in heterogeneous federated learning.

The main contributions of this paper include,

- In the local computation phase, the loss queue is adopted to detect the convergence trends within each client. FedDAC can then dynamically quantify model updates and allow for various compression ratios among heterogeneous clients.
- In the global aggregation phase, FedDAC can determine the fluctuations in training based on the similarity between the local and global updates, thereby adjusting the sparsity ratio flexibly.
- To alleviate the reduction in model accuracy caused by lossy compression, we introduce residual updates in the local computation and global aggregation phases to maintain model accuracy.

The remainder of this paper is organized as follows: section II describes the related work of compressing the single-round communication volume in federated learning. Section III presents the system model of this paper. Section IV describes the specific implementation of FedDAC. Experiment results of FedDAC with other methods are analyzed in Section V. In the end, Section VI concludes this paper.

II. Related Work

To reduce the huge communication costs in federated learning, a common way is to compress the single-round communication volume. Specifically, the single-round communication volume consists of the upstream and downstream model updates. Most approaches employ gradient compression methods to reduce the size of model updates, including gradient quantization [9] and sparsification [10], thereby optimizing communication efficiency.

Bernstein et al. [17] propose a binary quantization mechanism to utilize 1 bit to represent each gradient, greatly reducing communication costs. However, the extreme compression impairs the model accuracy severely. To alleviate the problem of low model accuracy caused by quantization, related studies have investigated the variants of quantization, such as ternary quantization [18], variance reduction quantization [19], and gradient difference quantization [20].

The top-k method proposed by Aji et al. [10] is a typical sparsification scheme. Top-k sparsification sorts the absolute values of the gradients to choose the valid ones to be uploaded to the server. To compensate for the degradation of the model accuracy caused by sparsification, Lin [21] et al. sparse the gradients only when the accumulation of the small gradients exceeds a certain threshold. However, the above compression scheme does not apply to heterogeneous federated learning. Various compression ratios should be introduced to fit different data distributions among clients. The unified compression ratio employed in traditional quantization or sparsification exacerbates problems such as gradient conflict [11] and gradient drift [12], resulting in a decline in model accuracy.

In order to determine reasonable compression ratios for heterogeneous clients, researchers propose adaptive compression methods to reduce the single-round communication volume. Related methods employ the loss during local training which reflects the convergence trend of the global model, to determine the compression coefficients dynamically. For example, NAGC [13] proposes an adaptive sparsification method. Employed with the client loss queue which stores historical loss over multiple rounds, NAGC can dynamically adjust the sparsity ratio according to the difference between the current and historical loss. AdaQuantFL [14] utilizes the initial and current local loss to adjust the quantization coefficient, mitigating the problem of high error floor due to quantization.

TABLE I
MAIN SYMBOLIC PARAMETERS IN THIS PAPER

Symbol	Definition
M	Number of clients
P	Number of parameters in the model updates
S	Number of clients participating in training
η	Local learning rate
c_m	The m-th client participating in training
w_m^r	Local model of client c_m in r-th iteration
w^r	Global model in r-th iteration
Δw_m^r	Local model updates of client c_m in r-th iteration
Δw^r	Global model updates in r-th iteration
V_m^r	Single-round communication volume of c_m

Han et al. [22] propose a FAB-top-k sparsification mechanism with online learning. FAB-top-k predicts the fluctuation of the loss function to adjust the sparsity ratio. However, these methods only reduce the upstream communication volume, and the communication efficiency needs to be further optimized.

To reduce the single-round downstream communication volume, Sattler et al. [15] propose the sparse ternary compression method STC. Combined with sparsification and ternary quantization, STC can extremely reduce the communication costs in federated learning. Drawing on Sattler's work, Xu [16] et al. propose a ternary compression method T-FedAvg. T-FedAvg quantizes the local updates and sparsifies global updates during training, achieving bidirectional compression for upstream and downstream communication. However, the above methods utilize the fixed compression ratio, which is inapplicable to heterogeneous federation learning.

In general, existing methods either reduce only upstream or downstream communication volume or ignore the effect of heterogeneity on model accuracy. In this paper, we propose a Dual Adaptive Compression method (FedDAC) to achieve dynamic bidirectional compression while balancing the model accuracy and communication efficiency in heterogeneous federated learning.

III. System model

The federated system consists of a clients set $C = \{c_i,\ c_2,\ ...,\ c_M\}$ and a central server, where M is the number of clients. The clients set $S \subset C$ is selected for the r-th iteration, and each client in S optimizes the local loss function $f_m\left(w^r\right)$ on its private dataset D_m, as shown in equation (1):

$$f_m\left(w^r\right) = \sum_{j=1}^{|D_m|} l\left(w^r\left(x_j^m\right);\ y_j^m\right), \tag{1}$$

where w^r is the initialized model of clients in the r-th iteration, $x_j^m,\ y_j^m$ are the feature and label of the j-th sample in D_m, respectively. $l\left(w^r\left(x_j^m\right);\ y_j^m\right)$ is the loss between the prediction of w^r on x_j^m and y_j^m. The goal of federated learning is to obtain a global model w_g that minimizes the average loss of each client, as shown in equation (2):

$$\min_{w_g} \frac{1}{|S|} \sum_{c_m \in S} f_m\left(w_g\right). \tag{2}$$

In the r-th iteration, each client $c_m \in S$ applies equation (1) to calculate its local loss and then optimizes equation (2) with stochastic gradient descent (SGD). The initialized model w^r of c_m is updated as shown in equation (3):

$$w_m^r = w^r - \eta \nabla f_m\left(w^r\right), \tag{3}$$

where $\nabla f_m\left(w^r\right)$ is the gradient of the loss function $f_m\left(w^r\right)$, η represents the learning rate, and w_m^r is the updated local model of client c_m. Subsequently, the local model updates of client c_m is computed as shown in equation (4):

$$\Delta w_m^r = w_m^r - w^r. \tag{4}$$

Each client $c_m \in S$ uploads its local model updates to the central server for averaging and aggregating. The global model is updated as shown in equation (5):

$$w^{r+1} = w^r + \frac{1}{|S|} \sum_{c_m \in S} \Delta w_m^r, \tag{5}$$

where w^{r+1} is the global model of the $r+1$-th iteration. The central server distributes the global model updates Δw^r to all clients, which is computed as shown in equation (6):

$$\Delta w^r = w^{r+1} - w^r. \tag{6}$$

Each client computes its local initialized model of the $r+1$ -th iteration through $w^{r+1} = w^r + \Delta w^r$. The communication volume between the client c_m and the central server in the r-th iteration is shown in equation (7):

$$V_m^r = sizeof\left(\Delta w_m^r\right) + sizeof\left(\Delta w^r\right), \tag{7}$$

where $sizeof\left(\Delta w\right)$ is used to compute the size of model updates Δw , as shown in equation (8):

$$sizeof\left(\Delta w\right) = P * bits_p, \tag{8}$$

where P is the number of parameters in the model updates Δw and $bits_p$ is the number of bits required to represent each parameter. Our goal is to reduce the single-round communication volume V_m^r while maintaining the global model accuracy. The main symbolic parameters of FedDAC proposed in this paper are shown in Table I.

IV. The design of FedDAC

A. Overall Framework

Fig. 2 is the framework of FedDAC. In each round of iteration, FedDAC consists of three phases: local updates quantization, global updates sparsification, and local and global residual updates. Combined with local updates quantization and global updates sparsification, FedDAC can achieve dual adaptive compression to optimize communication efficiency with the consideration of data heterogeneity in federated learning. Employed with the local and global residual updates, FedDAC can mitigate the degradation of global model accuracy caused by lossy compression.

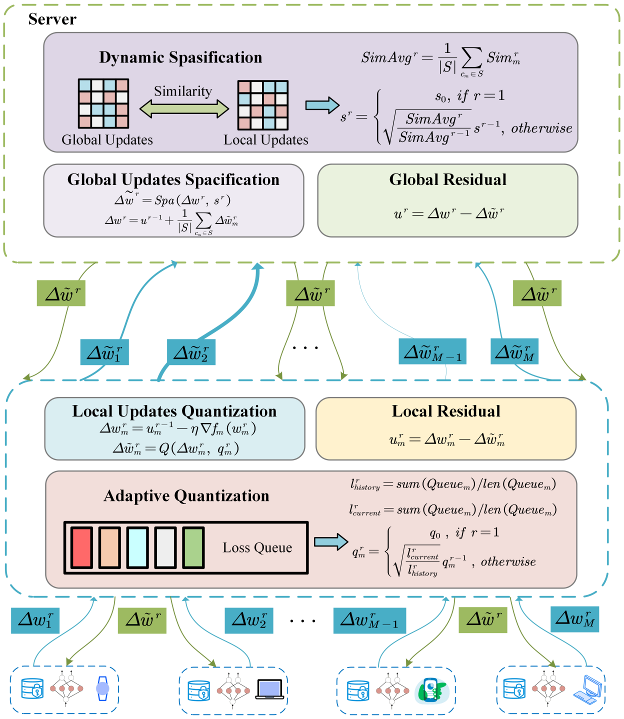

Fig. 2. The framework of FedDAC

B. Adaptive Local Updates Quantization

In the local computation phase, in order to enable each client to determine the various compression ratios with the consideration of its own training state and data distribution, an adaptive local updates quantization scheme based on the loss of clients is proposed. Specifically, a loss queue $Queue_m$ of capacity μ is defined for each client c_m, which is utilized to store the local loss of client c_m. In the r-th iteration, the client c_m computes the historical average loss, which can be formulated as equation (9):

$$l_{history}^{r} = sum\left(Queue_m\right)/len\left(Queue_m\right), \tag{9}$$

where $sum\left(Queue_m\right)$ is the sum of the losses stored in the loss queue, and $len\left(Queue_m\right)$ is the length of the loss queue.

When the client obtains the historical average loss $l_{history}^{r}$, it calculates the local loss in the current iteration according to equation (1) and stores the updated loss in its loss queue. Before depositing the loss into the queue, the first loss in the queue needs to be discharged if $len\left(Queue_m\right) = \mu$, and then the updated loss is inserted. The current average loss of c_m is calculated after the loss queue $Queue_m$ is updated, as shown in equation (10):

$$l_{current}^{r} = sum\left(Queue_m\right)/len\left(Queue_m\right). \tag{10}$$

The quantization coefficient of client c_m in the r-th iteration is determined based on the ratio of the current average loss and historical average loss, as shown in equation (11):

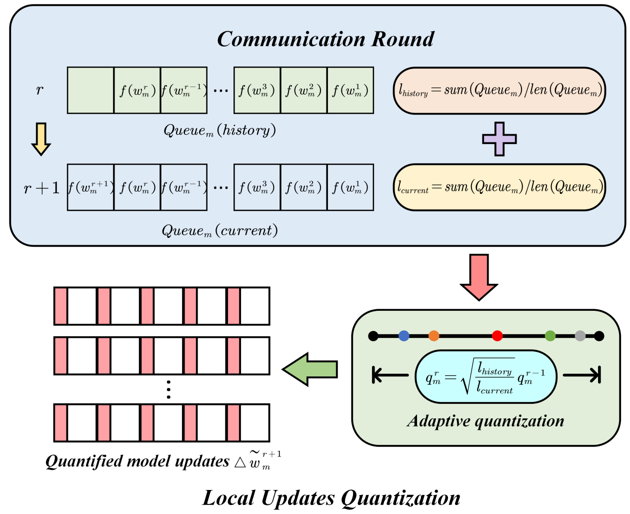

Fig. 3. The illustration of the adaptive local updates quantization

$$q_m^r = \begin{cases} q_0\ , & if\ r = 1 \\ \sqrt{\frac{l_{current}^{r}}{l_{history}^{r}}}\, q_m^{r-1}\ , & otherwise \end{cases}\ , \tag{11}$$

where q_0 is the initial quantization coefficient. The random uniform quantizer [23] is then utilized to compress local model updates Δw_m^r of client c_m, as shown in equation (12):

$$\Delta\widetilde{w}_m^r = Q\left(\Delta w_m^r, q_m^r\right) = \|\gamma_m^r\|_2 sign\left(\Delta w_m^r\right)\zeta_m\left(\gamma_m^r,\ q_m^r\right), \tag{12}$$

where γ_m^r is the vector of local model updates Δw_m^r after flattening, $sign\left(\right)$ is the symbolic function, and $\zeta_m\left(\gamma_m^r,\ q_m^r\right)$ is the random variable determined by the quantization coefficient q_m^r. Fig. 3 is the illustration of the adaptive local updates quantization.

Considering the different data distributions among heterogeneous clients, FedDAC can detect the convergence trends of the local model based on the loss queue, thereby assigning various quantization coefficients to each client. With the adaptive local updates quantization, FedDAC can alleviate the degradation of global model accuracy caused by unreasonable compression.

C. Dynamic Global Updates Sparsification

In the initial stage of training, the data distribution among heterogeneous clients leads to a large difference between local and global updates. A smaller compression ratio should be employed to maintain the integrity of the model updates, thereby improving the training effect in the initial stage. When the global model tends to converge, the difference between local and global updates becomes smaller. The compression ratio can be scaled up to further optimize the communication efficiency with the global model accuracy not being impaired. Based on the above analysis, we propose the dynamic global updates sparsification with the similarity between local and

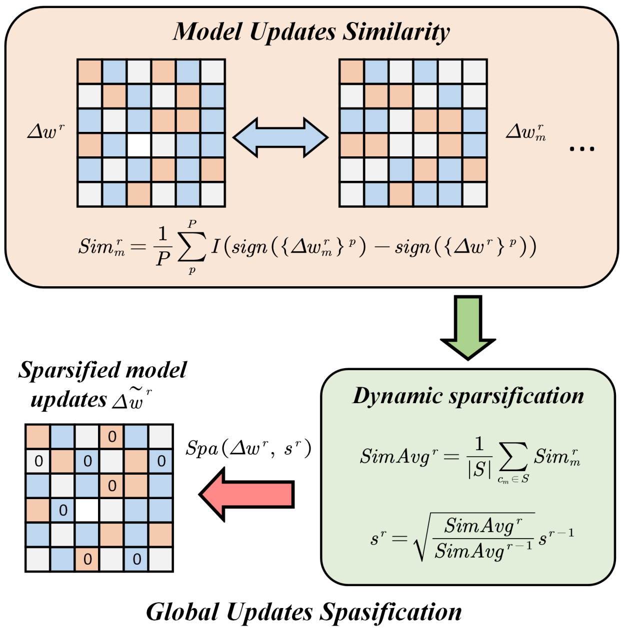

Fig. 4. The illustration of the dynamic global updates sparsification

global updates. The similarity between Δw_m^r and Δw^r in the r-th iteration is defined in equation (13):

$$Sim_m^r = \frac{1}{P}\sum_p^P I\left(sign\left(\{\Delta w_m^r\}^p\right) = sign\left(\{\Delta w^r\}^p\right)\right), \tag{13}$$

where P is the number of parameters in model updates, $\{\Delta w\}^p$ represents the p-th parameter of model updates Δw, and $sign\,()$ is the symbolic function. The update direction of the p-th parameter between Δw_m^r and Δw^r is identical if $I\,()$ equals to 1. In the global aggregation phase, the average local-global similarity is calculated as shown in equation (14):

$$SimAvg^r = \frac{1}{|S|}\sum_{c_m \in S} Sim_m^r, \tag{14}$$

where S is the clients involved in the r-th iteration. The sparsity ratio is defined with the current and previous average local-global similarity, as shown in equation (15):

$$s^r = \begin{cases} s_0, \ if\ r = 1 \\ \sqrt{\frac{SimAvg^r}{SimAvg^{r-1}}}\, s^{r-1}, \ otherwise \end{cases}, \tag{15}$$

where s_0 is the initial sparsity ratio. After determining the sparsity ratio, FedDAC specifies the global model updates to reduce the single-round downstream communication volume, as shown in equation (16):

$$\Delta\widetilde{w}^r = Spa\left(\Delta w^r,\ s^r\right), \tag{16}$$

where $\Delta\widetilde{w}^r$ is the sparsified global model updates. During sparsification, the absolute values of the parameters in the global model updates are sorted in descending order. The smallest $s^r * P$ parameters in the global model updates are set to 0, indicating that these parameters are not distributed to the clients.

Algorithm 1 Dual Adaptive Compression for Efficient Communication in Heterogeneous Federated Learning (FedDAC)

1 **Input:** initial global model w^1, learning rate η, number of iterations R, number of clients M, capacity of loss queue μ
Output: the global model w^{R+1}

Define a $Queue$ of capacity μ for each client
for $r = 1, 2, \cdots, R$ **do**
  Clients:
  for Client $c_m \in S$ **do**
    Download sparsified model updates $\Delta\widetilde{w}^{r-1}$
    $w_m^r = w_m^{r-1} + \Delta\widetilde{w}^{r-1}$
    $l_{history}^r = sum\left(Queue_m\right)/len\left(Queue_m\right)$
    if $len(Queue_m) = \mu$ **then**
      Remove the first loss in $Queue_m$
    end if
    Add $f_m(w^r)$ to $Queue_m$
    $l_{current}^r = sum\left(Queue_m\right)/len\left(Queue_m\right)$
    $q_m^r = \begin{cases} q_0\ ,\ if\ r = 1 \\ \sqrt{\frac{l_{current}^r}{l_{history}^r}}\, q_m^{r-1}\ ,\ otherwise \end{cases}$
    $\Delta w_m^r = u_m^{r-1} - \eta\nabla f_m\left(w^r\right)$
    $\Delta\widetilde{w}_m^r = Q\left(\Delta w_m^r, q_m^r\right)$
    $u_m^r = \Delta w_m^r - \Delta\widetilde{w}_m^r$
    Communicate $\Delta\widetilde{w}_m^r$ **to Server**
  end for
  Server:
  $\Delta w^r = u^{r-1} + \sum_{c_m \in S}\frac{1}{|S|}\Delta\widetilde{w}_m^r$
  for Client $c_m \in S$ **do**
    $Sim_m^r = \frac{1}{P}\sum_p^P I\left(sign\left(\{\Delta\widetilde{w}_m^r\}^p\right) = sign\left(\{\Delta w^r\}^p\right)\right)$
  end for
  $SimAvg^r = \frac{1}{|S|}\sum_{c_m \in S} Sim_m^r$
  $s^r = \begin{cases} s_0,\ if\ r = 1 \\ \sqrt{\frac{SimAvg^r}{SimAvg^{r-1}}}\, s^{r-1},\ otherwise \end{cases}$
  $\Delta\widetilde{w}^r = Spa\left(\Delta w^r,\ s^r\right)$
  $u^r = \Delta w^r - \Delta\widetilde{w}^r$
  Communicate $\Delta\widetilde{w}^r$ **to all the clients**
end for
Return w^{R+1}

D. Local and Global Residual Updates

To alleviate the impairment of global model accuracy caused by quantization and sparsification, FedDAC introduces the residual updates on both local computation and global aggregation phases. The residual refers to the difference between the full precision model and the lossy compressed model.

In the local computation phase, each client $c_m \in S$ calculates the local residual between $\Delta\widetilde{w}_m^r$ and Δw_m^r, as shown in equation (17):

$$u_m^r = \Delta w_m^r - \Delta\widetilde{w}_m^r. \tag{17}$$

TABLE II
EXPERIMENTAL ENVIRONMENT DETAILS

Central Server		
CPU	Intel Xeon Platinum 8369B@2.4 GHz 8 cores	
RAM	16GB	
Client		
Name	Hardware Specifications	Num
Jetson Orin Nano	Ampere CUDA 1024 cores GPU 8G RAM	5
Jetson Orin NX	Ampere CUDA 1024 cores GPU 8G RAM	5
Jetson TX2	Maxwell CUDA 128 cores GPU 4G RAM	10
Jetson Orin Nano	Ampere CUDA 512 cores GPU 4G RAM	10
12500 Docker Node	4 cores Intel i5-12500H@2.5GHz 4G RAM	30
13400 Docker Node	4 cores Intel i5-13400@2.5GHz 4G RAM	40

The local residual in the $r-1$-th iteration is added into equation (4) for obtaining the local updates Δw_m^r, as shown in equation (18):

$$\Delta w_m^r = u_m^{r-1} - \eta \nabla f_m\left(w^r\right). \tag{18}$$

In the global aggregation phase, the server computes the global residual between $\Delta \widetilde{w}^r$ and Δw^r, as shown in equation (19):

$$u^r = \Delta w^r - \Delta \tilde{w}^r. \tag{19}$$

After the server aggregates the local model updates, the global residual of $r-1$-th round of training is introduced as shown in equation (20):

$$\Delta w^r = u^{r-1} + \sum_{c_m \in S} \frac{1}{|S|} \triangle \tilde{w}_m^r. \tag{20}$$

Employed with the local and global residual updates, FedDAC can fill in the missing information, thereby accelerating the convergence speed of the global model and improving the global model accuracy.

E. Algorithm Design

In the local computation phase, the clients calculate the historical and current average loss to determine the quantization coefficient. Then the residual of the local updates is computed and the quantified updates is uploaded to the server. In the global aggregation phase, the server averages the local updates to obtain the global updates. The average similarity between local updates and global updates is computed to determine the sparsity ratio. The sparsified model updates is then distributed to the clients, and the residual of the global updates is computed. Algorithm 1 shows the detailed steps of FedDAC.

V. PERFORMANCE EVALUATION

A. Experiment Setup

1) Experimental environment: In this paper, we adopt an Alibaba cloud server as the central server in federated learning. Clients consist of 30 jetson development boards and 70 docker nodes on different workstations. Clients are in the same LAN and communicate with the central server through the network gateway. The experimental environment details are shown in Table II.

2) Federated datasets and models: The public datasets MNIST and CIFAR-10 are chosen for our experiments. In order to simulate the non-IID data in federated learning, the Dirichlet distribution is employed to generate datasets with different degrees of heterogeneity. The α in Dirichlet distribution determines the degree of heterogeneity. The smaller α is, the more heterogeneous the data distribution among clients is.

Since the CIFAR-10 dataset is more complex compared to the MNIST dataset in terms of categories and RGB channels, models with different architectures are chosen for the two datasets. Specifically, Logistic Regression and AlexNet are employed for the MNIST dataset and CIFAR-10 dataset, respectively, thereby evaluating the performance of FedDAC on both convex and non-convex models.

3) Baselines and parameter settings: We choose NAGC [13], AdaQuantFL [14], and T-FedAvg [16] as the baselines against FedDAC. NAGC and AdaQuantFL are one-way compression methods considering variable compression ratios. T-FedAvg is a two-way compression method with a fixed compression coefficient.

For the MNIST dataset, the number of iterations R is set to 200, the learning rate η is set to 0.1, the number of clients M is set to 100, and randomly select 10 clients to participate in each round of training. The initial quantization coefficient q_0 is set to 64 and the initial sparsity ratio s_0 is set to 0.2.

For the CIFAR-10 dataset, the learning rate η is set to 0.01, the initial quantization coefficient q_0 is set to 128 and the initial sparsity ratio s_0 is set to 0.1. The other parameters remain the same with the MNIST dataset.

The accumulated communication volume for evaluating the methods refers to the total communication volume between clients and the server when the global model reaches specific accuracy.

B. Analysis of hyperparameter selection for FedDAC

The capacity μ of the loss queue in FedDAC determines how close the client quantization coefficients are among each iteration. The larger μ is, the smaller the difference between the historical average loss and the current average loss is, and the smaller the change in quantization coefficients is. In order to choose reasonable μ for the subsequent experiments, we conduct experiments on MNIST and CIFAR-10 datasets under different values of μ with the introduction of different degrees of heterogeneity. The experiment results are shown in Table III to Table IV.

1) Analysis of the communication volume with different μ : Columns 2 to 4 in Table III show the accumulated communication volume of FedDAC on the heterogeneous MNIST dataset. Under the condition of weak heterogeneity ($\alpha = 10$), the minimum accumulated communication volume is 1.4MB when $\mu = 10$, and as μ becomes larger, the accumulated communication volume significantly increases. Under strong heterogeneity ($\alpha = 0.5$, $\alpha = 1$), the accumulated communication volume of FedDAC is minimal when $\mu = 20$.

TABLE III
ACCUMULATED COMMUNICATION VOLUME (MB) WITH μ

μ	Dataset					
	MNIST			CIFAR-10		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
10	12.0	11.3	**1.4**	**215.0**	**163.8**	**153.6**
20	**10.8**	**8.9**	1.5	225.3	184.3	163.8
30	16.9	16.1	3.2	245.8	225.3	204.8
40	32.3	29.0	5.7	358.4	317.4	256.0
50	63.2	59.6	11.9	409.6	389.1	378.9

TABLE IV
GLOBAL MODEL ACCURACY (%) WITH μ

μ	Dataset					
	MNIST			CIFAR-10		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
10	86.84	90.09	90.37	52.86	60.19	**63.33**
20	86.87	90.10	90.36	52.90	60.22	63.30
30	86.86	90.12	90.35	52.84	60.21	63.28
40	86.86	90.13	90.38	52.87	60.23	63.32
50	**86.88**	**90.15**	**90.41**	**52.91**	**60.26**	63.31

Columns 5 to 7 in Table III show the accumulated communication volume on the heterogeneous CIFAR-10 dataset. Under all degrees of data heterogeneity ($\alpha = 0.5$, $\alpha = 1$, $\alpha = 10$), the communication costs of FedDAC is minimal when $\mu = 10$, with accumulated communication volume of 215.0MB, 163.8MB, and 153.6MB, respectively. The accumulated communication volume of FedDAC on the CIFAR-10 dataset significantly increases as the value of μ increases.

The above analysis reveals that FedDAC can achieve the smallest accumulated communication volume when $\mu = 10$ under different degrees of data heterogeneity.

As μ increases, the communication efficiency of FedDAC greatly drops. This is because the value of μ affects the quantization coefficients of the clients. As μ increases, the change in the quantization coefficient becomes smaller, leading to the increment in the single-round upstream communication volume.

2) Analysis of the global model accuracy with different μ: To achieve a trade-off between communication efficiency and model accuracy, we analyze the global model accuracy of FedDAC under various conditions. The results of the global model accuracy on heterogeneous MNIST dataset under different values of μ are shown in columns 2 to 4 in Table IV. FedDAC achieves the best global model accuracy with 86.88%, 90.15%, and 90.41% respectively under all degrees of data heterogeneity ($\alpha = 0.5$, $\alpha = 1$, $\alpha = 10$) when $\mu = 50$.

The results of the global model accuracy on the heterogeneous CIFAR-10 dataset are shown in columns 5 to 7 in Table IV. Under the condition of weak heterogeneity ($\alpha = 10$), FedDAC achieves the highest global model accuracy of 63.33% when $\mu = 10$. Under the degrees of high data heterogeneity ($\alpha = 0.5$, $\alpha = 1$), FedDAC achieves the highest global model accuracy of 52.91% and 60.26% respectively when $\mu = 50$.

The above analysis shows that FedDAC can achieve the smallest accumulated communication volume when $\mu = 10$ and obtains the best global model accuracy when $\mu = 50$. To balance the communication efficiency and global model accuracy, we choose $\mu = 10$ for the subsequent experiments. Therefore, the highest communication efficiency can be obtained with a slight decrease in global model accuracy.

C. Analysis of the accumulated communication volume

1) Experiment results on MNIST: Under the above experimental environment and hyperparameter settings, we compare the accumulated communication volume of FedDAC with NAGC, AdaQuantFL, and T-FedAvg. Fig. 5 shows the experiment results on the heterogeneous MNIST dataset.

As shown in Fig. 5 (a), under the strong data heterogeneity situation ($\alpha = 0.5$), FedDAC achieves the smallest accumulated communication volume with 10.36MB, 10.48MB, and 11.46MB respectively when the global model accuracy reaches 76%, 80%, and 84% Compared with NAGC, AdaQuantFL, and T-FedAvg, the accumulated communication volume of FedDAC is reduced by about 16.7 times, 13.2 times, and 2.8 times, respectively when the global model accuracy reaches 76%. Combined with adaptive local updates quantization and dynamic global updates Sparsification, FedDAC can greatly reduce the single-round upstream and downstream communication volume. T-FedAvg quantizes the local updates and sparsifies global updates during training. However, T-FedAvg adopts a fixed compression ratio, resulting in suboptimal accumulated communication volume. The accumulated communication volume of AdaQuantF is smaller than NAGC due to the decaying learning rate scheme for further adjusting the quantization ratios. However, both AdaQuantFL and NAGC only compress the upstream communication volume and the communication efficiency can be further optimized.

Fig. 5 (b) shows the accumulated communication volume of four methods when $\alpha = 1$. FedDAC achieves the best communication efficiency compared with NAGC, AdaQuantFL, and T-FedAvg. The accumulated communication volume of FedDAC reaches 8.25MB, 9.73MB, and 10.75MB, respectively when different model accuracies are obtained, surpassing the suboptimal method T-FedAvg by about 2.8 times, 2.7 times, and 2.6 times.

The accumulated communication volume of the four methods under the weak data heterogeneity situation ($\alpha = 0.5$) is shown in Fig. 5 (c). The accumulated communication volume of the four methods reduces as the data heterogeneity becomes weak. FedDAC still achieves the smallest accumulated communication volume with 0.99MB, 1.17MB, and 1.32MB, respectively when the global model accuracy reaches 76%, 80%, and 84%.

2) Experiment results on CIFAR-10: Fig. 6 illustrates the accumulated communication volume of the four methods on the heterogeneous CIFAR-10 dataset. As the model adopted for CIFAR-10 becomes complex, the accumulated communication volume greatly increases under all situations. Similar

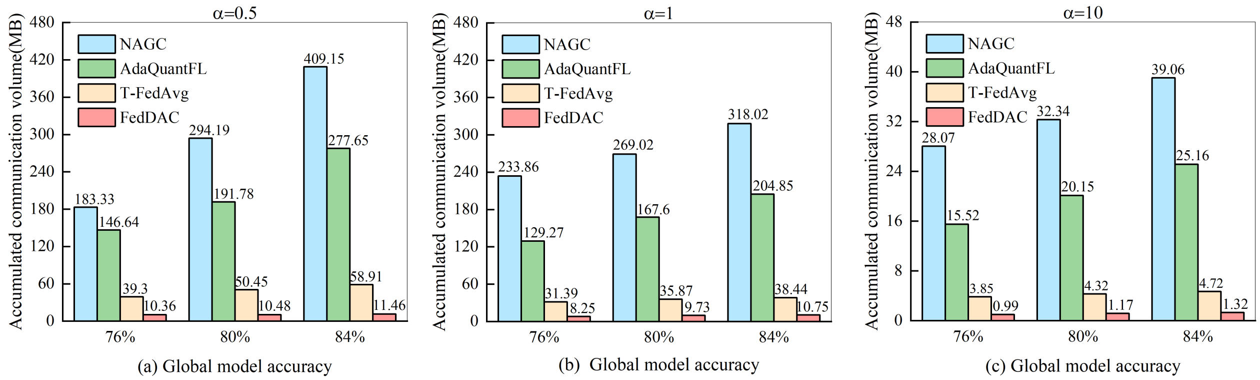

Fig. 5. Accumulated communication volume on the heterogeneous MNIST dataset for NAGC, AdaQuantFL, T-FedAvg, and FedDAC

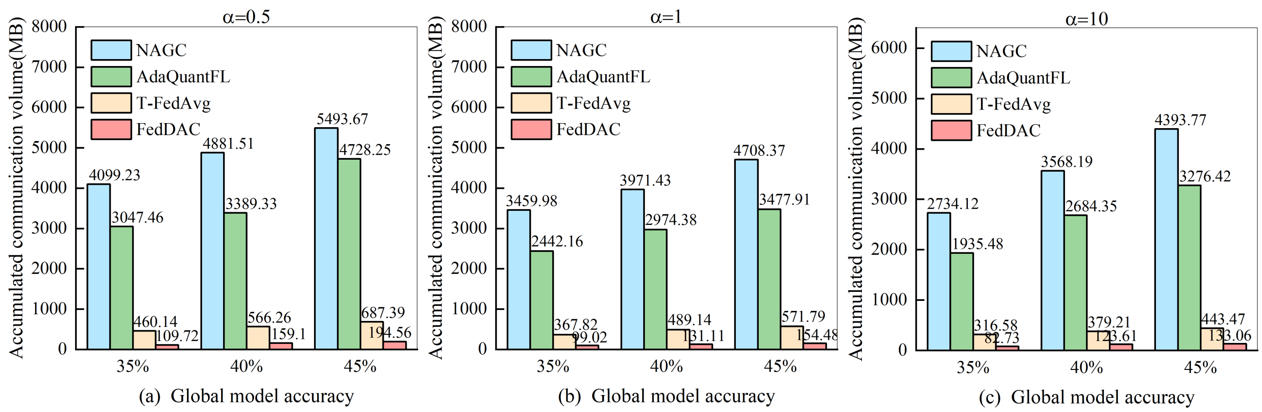

Fig. 6. Accumulated communication volume on the heterogeneous CIFAR-10 dataset for NAGC, AdaQuantFL, T-FedAvg, and FedDAC

to the results on the MNIST dataset, FedDAC outperforms the other three methods on communication efficiency. Under the strong data heterogeneity situation ($\alpha = 0.5$), FedDAC surpasses NAGC, AdaQuantFL, and T-FedAvg by about 27.2 times, 23.3 times, and 2.5 times when the global model accuracy reaches 45%.

The above analysis reveals that compared with the one-way compression methods NAGC and AdaQuantFL, FedDAC can greatly reduce communication costs with the introduction of bidirectional compression. Employed with dual adaptive compression, FedDAC can also perform the two-way compression method T-FedAvg. Generally speaking, FedDAC can significantly optimize the communication efficiency in heterogeneous federated learning.

D. Analysis of the global model accuracy

Table V shows the results of the global model accuracy experiments of NAGC, AdaQuantFL, T-FedAvg, and FedDAC on different heterogeneous datasets under the above experimental environments and hyperparameter settings.

1) Experiment Results on MNIST: The results on the MNIST dataset are shown in columns 2 to 4 of Table V. Under the condition of weak heterogeneity ($\alpha = 10$), the method AdaQuantFL with training loss and decaying learning rate obtains the highest global model accuracy of 91.55%, outperforming FedDAC by about 1.18%. As the degree of data heterogeneity increases ($\alpha = 1$, $\alpha = 0.5$), FedDAC achieves the best global model accuracy of 90.09% and 86.84%, respectively, surpasses the suboptimal AdaQuantFL by about 0.74% and 0.58%, respectively.

2) Experiment Results on CIFAR-10: Columns 5 to 7 of Table V show the experimental results of the four methods on the CIFAR-10 dataset with different degrees of data heterogeneity. The increase in model and dataset complexity leads to large differences in global model accuracy results. Specifically, FedDAC obtains the optimal global model accuracy of 63.33% when $\alpha = 10$. As the degree of data heterogeneity increases ($\alpha = 1$ and $\alpha = 0.5$), the global model accuracy of FedDAC becomes lower than that of AdaQuantFL by approximately 2.1% and 2.2%, respectively. Although FedDAC introduces residual updates to improve the model accuracy impaired by quantization and sparsification, its global model accuracy still needs improvement compared to the one-way compression method AdaQuantFL. Moreover, the global model accuracy of FedDAC is significantly higher than those of T-FedAvg and NAGC, where T-FedAvg obtains the worst global model accuracy due to the fixed bidirectional compression ratio.

The above analysis of the experimental results shows that NAGC and T-FedAvg have worse global model accuracy in heterogeneous federated learning. NAGC adjusts the compres-

TABLE V
GLOBAL MODEL ACCURACY OF DIFFERENT APPROACHES

Comparison approaches	**Dataset**					
	MNIST			**CIFAR-10**		
	α = 0.5	***α = 1***	***α = 10***	***α = 0.5***	***α = 1***	***α = 10***
NAGC	84.20	87.78	88.82	49.41	57.39	61.25
AdaQuantFL	86.34	89.42	**91.55**	**53.02**	**61.47**	62.32
T-FedAvg	84.29	88.17	89.37	46.76	54.29	60.88
FedDAC	**86.84**	**90.09**	90.37	51.86	60.19	**63.33**

sion ratio based on the change of loss without considering the trade-off between the model accuracy and communication efficiency. T-FedAvg adopts the fixed compression coefficient to reduce upstream and downstream communication volume, resulting in serious degradation of global model accuracy under strong heterogeneity. AdaQuantFL introduces the dynamic compression ratio to fit the heterogeneous data, thereby realizing a better global model accuracy.

In general, the global model accuracy of FedDAC is comparable to AdaQuantFL and outperforms the other two methods. Combined with adaptive local updates quantization and dynamic global updates sparsification, FedDAC can adjust the upstream and downstream compression ratio to maintain the global model accuracy with heterogeneous data. With the introduction of local and global residual updates, FedDAC can further improve the global model accuracy impaired by the dual lossy compression.

VI. Conclusions

In order to reduce the significant communication costs in heterogeneous federated learning while achieving the trade-off between communication efficiency and global model accuracy, a Dual Adaptive Compression method (FedDAC) is proposed in this paper. In the local computation phase, FedDAC can detect the convergence trend with the introduction of the local loss queue to justify the upstream quantization ratio. In the global aggregation phase, FedDAC can determine the fluctuations in training based on the similarity between clients and the server, thereby adjusting the downstream sparsity ratio. Employed with local and global residual updates, FedDAC can mitigate the degradation of global model accuracy due to lossy compression. Experiment results show that compared with one-way compression methods NAGC and AdaQuantFL, FedDAC can maintain comparable model accuracy and greatly optimize communication efficiency. Compared with two-way compression methods T-FedAvg, FedDAC can further compress the single-round communication volume while achieving better global model accuracy.

References

[1] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, “URLLC and eMBB in 5G Industrial IoT: A Survey,” *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1134–1163, 2022.

[2] S. Vhaduri, W. Cheung, and S. V. Dibbo, “Bag of On-Phone ANNs to Secure IoT Objects Using Wearable and Smartphone Biometrics,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–12, 2023.

[3] I. Zualkernan, S. Dhou, J. Judas, A. R. Sajun, B. R. Gomez, and L. A. Hussain, “An IoT System Using Deep Learning to Classify Camera Trap Images on the Edge,” *Computers*, vol. 11, no. 1, Jan. 2022.

[4] K. W. Chen, M. R. Xie, Y. M. Chen, T. T. Chu, and Y. B. Lin, “DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15204–15217, Sep. 2022.

[5] M. S. Al-kahtani, F. Khan, and W. Taekeun, “Application of Internet of Things and Sensors in Healthcare,” *Sensors*, vol. 22, no. 15, Art. no. 15, Jan. 2022.

[6] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, ”Federated optimization: Distributed machine learning for on-device intelligence”, *arXiv:1610.02527*, 2016.

[7] A. Vaswani et al., “Attention is All you Need,” *in Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Dec. 07, 2023.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *arXiv*, May 24, 2019.

[9] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, “1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs,” *in Interspeech 2014, ISCA*, pp. 1058–1062, Sep. 2014.

[10] A. F. Aji and K. Heafield, “Sparse Communication for Distributed Gradient Descent,” *in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 440–445, 2017.

[11] H. Yu, S. Yang, and S. Zhu, “Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Art. no. 01, Jul. 2019.

[12] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic Federated Learning,” *in Proceedings of the 36th International Conference on Machine Learning, PMLR*, pp. 4615–4625, May 2019.

[13] W. Yang, Y. Yang, X. Dang, H. Jiang, Y. Zhang, and W. Xiang, “A Novel Adaptive Gradient Compression Approach for Communication-Efficient Federated Learning,” *in 2021 China Automation Congress (CAC)*, pp. 674–678, Oct. 2021.

[14] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar, “Adaptive Quantization of Model Updates for Communication-Efficient Federated Learning,” *in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3110–3114, Jun. 2021.

[15] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[16] J. Xu, W. Du, R. Cheng, W. He, and Y. Jin, “Ternary Compression for Communication-Efficient Federated Learning,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 33, no. 3, pp. 1162–1176, Mar. 2022.

[17] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, “signSGD: Compressed Optimisation for Non-Convex Problems,” *in Proceedings of the 35th International Conference on Machine Learning, PMLR*, pp. 560–569, Jul. 2018.

[18] W. Wen et al., “TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning,” *in Advances in Neural Information Processing Systems, Curran Associates, Inc.*, 2017.

[19] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang, “ZipML: Training Linear Models with End-to-End Low Precision, and a Little Bit of Deep Learning,” *in Proceedings of the 34th International Conference on Machine Learning, PMLR*, pp. 4035–4043, Jul. 2017.

[20] S. Magnússon, H. Shokri-Ghadikolaei, and N. Li, “On Maintaining Linear Convergence of Distributed Learning and Optimization Under Limited Communication,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 6101–6116, 2020.

[21] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, “Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training,” *arXiv.org*, 2023.

[22] P. Han, S. Wang, and K. K. Leung, “Adaptive Gradient Sparsification for Efficient Federated Learning: An Online Learning Approach,” *in 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 300–310, Nov. 2020.

[23] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding,” *in Advances in Neural Information Processing Systems, Curran Associates, Inc.*, 2017.