Bandwidth-Aware Adaptive Gradient Quantization for Cross-Organization Federated Learning

Su Liu[†], Hong Shen[‡,†], Chan-Tong Lam[†], Eddie K. L. Law[†]
[†]Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, China
[‡]School of Engineering and Technology, Central Queensland University, Australia

***Abstract*—Gradient quantization is an effective way of compressing gradient data to reduce communication overhead in federated learning (FL) across a wide-area network (WAN). Existing gradient quantization methods are either static adopting a uniform quantization strategy across all nodes or dynamic according to the training demands without taking into account link bandwidths available at different nodes, which hinges the performance of FL due to the communication bottleneck. This bottleneck becomes severe for cross-organizational FL in a WAN due to the dynamic heterogeneity across different nodes.**

This paper proposes a bandwidth-aware adaptive gradient quantization method to tackle the communication bottleneck caused by bandwidth heterogeneity for FL in WANs. It performs gradient quantization by adaptively adjusting the bit-width of gradient quantization based on the bandwidth condition of each node — large on low-speed links to accelerate transmission and small on high-speed links to improve global model accuracy. We present the FL algorithm under our framework and show how to compute the bit-width of each node by determining the appropriate number of discretization levels for gradient quantization. The experimental results demonstrate that our method reduces communication overhead by approximately 87.5% compared to non-quantized gradient methods, while achieving significantly higher gradient precision compared to fixed quantization methods such as 2-bit and 4-bit approaches. Furthermore, experimental evaluations on multiple federated learning datasets show that our method achieves convergence time acceleration ranging from 1.57× to 4.80× compared to four existing schemes. These findings highlight the important application value of our method for cross-organizational distributed federated learning.

***Index Terms*—Hierarchical Federated Learning; Communication Efficiency; Gradient Compression; Bandwidth Awareness; Adaptive Gradient Quantization**

I. Introduction

Cross-organization federated learning (FL), as a form of distributed machine learning architecture, is typically used to collaboratively train models across multiple geographically distributed data centers (DCs) with a central parameter server (PS). In this setting, multiple DCs, connected via a WAN, each store data collected from devices of specific organizations and work together to train a shared global model. This system architecture inherently faces the challenge of heterogeneous and dynamic communication resources in WANs.

Operating over WANs, cross-organization federated learning presents several unique challenges compared to traditional distributed parameter server architectures due to link bandwidth limitation and node heterogeneity across DCs. As shown in recent studies, communication time in large-scale distributed machine learning systems with 32 client servers can account for up to 90% of the total training time [1], and training some state-of-the-art distributed machine learning systems across data centers in WAN environments can lead to a slowdown by factors ranging from 1.8× to 53.7× [2]. As different client severs may have varying bandwidths when connecting to the PS, the node with the least bandwidth may become the bottleneck for the entire training process, causing the *so-called straggler problem* [3].

Existing gradient quantization methods are either static with fixed discritization intervals or adaptive only to training requirements from indidual nodes, therefore unable to resolve the communication bottleneck caused by link bandwidth heterogeniety in cross-organization federated learning. To address this issue, this paper proposes a **Bandwidth-Aware Dynamic Adaptive Gradient Quantization (BA-DAGQ)** method that dynamically adjusts the gradient quantization strategy based on the actual bandwidth between client servers and the PS. As the main contribution, BA-DAGQ is based on the following strategies:

1) **Bandwidth-aware gradient quantization**: Determine appropriate quantization bit-widths (intervals) for individual client severs based on their communication bandwidths and the PS. Low-bandwidth nodes use large bit-width quantization to reduce data transmission, and high-bandwidth nodes use small bit-width quantization to preserve gradient precision, achieving a balance between communication efficiency and model accuracy.
2) **Adaptive bit-width adjustment**: Dynamically sense real-time changes in node bandwidths during training and adjust the quantization bit-widths of individual nodes. This ensures that low-bandwidth nodes do not become the communication bottleneck of the system when increasing the contribution of high-bandwidth nodes, thus alleviating the *straggler problem.*

ISBN 978-3-903176-72-0 ©2025 IFIP

II. Related Work

Significant research effort has been made on gradient compression for communication optimization in Federated Learning in recent years. Gradient compression techniques can be broadly categorized into two classes: gradient sparsification and gradient quantization including fixed quantization and adaptive quantization. This section reviews recent studies on gradient quantization that are closely related to this work.

A. Fixed Gradient Quantization

Fixed gradient quantization reduces communication overhead by representing gradients using fewer bits. Gupta et al. [4] introduced a stochastic rounding approach to quantize gradients to 16 bits, demonstrating strong performance on neural network training tasks using the MNIST and CIFAR-10 datasets. Alistarh et al. [5] introduced a randomized quantization mechanism that dynamically adjusts the quantization levels of gradient values, achieving a better balance between communication efficiency and model performance. Similarly, TernGrad method [6] leverages gradient sparsity by ignoring smaller gradient values, thereby further reducing the transmission volume while significantly decreasing communication overhead without compromising model accuracy. Bernstein et al. [7] proposed a method that transmits only the gradient signs (SignSGD) and aggregates gradient information across distributed nodes using a majority vote mechanism. However, in heterogeneous networks, variations in bandwidth and latency among nodes may lead to synchronization issues, potentially undermining the effectiveness of the voting mechanism.

Fixed quantization methods are widely used in distributed training but encounter significant challenges in WAN environments. Due to the heterogeneous network conditions of federated learning nodes, such as varying bandwidth, latency and packet loss rates, they often lead to inefficient utilization of computational and communication resources.

B. Adaptive Gradient Quantization

Adaptive Gradient Quantization (AGQ) dynamically adjusts quantization strategies according to the varying demands of different training phases, thereby achieving more efficient communication optimization. Samuel et al. [8] introduced a stochastic rounding technique that effectively enhances communication efficiency by rounding both positive and negative entries of the gradient vector to the nearest power of two. Sun et al. [9] proposed the Layer-wise Adaptive Quantization (LAQ) method, which not only quantizes gradients but also compresses the transmitted gradient data by ignoring gradients with low information content during each iteration, enabling adaptive optimization of communication efficiency. Yu et al. [10] proposed a heuristic gradient clipping strategy that incorporates the concept of dual sampling, ingeniously combining full-precision and low-precision gradients to minimize communication costs. Ivkin et al. [11] developed a ternary gradient method that represents each element of the gradient using only three values -1, 0, 1, thereby substantially improving communication efficiency in distributed deep learning.

TABLE I
Symbols and Definitions

Symbol	Definition
N_i	Number of DCs participating in training
M_i	Number of clients within data center i.
T	Upper limit of the number of training rounds
k	Minimum batch-size during training process
D_i	Dataset used by DC i
D	Global dataset, where $D_i \in D$
n_g	Number of gradients DCs send to the sever
w	Model parameters, $w \in \mathbb{R}^d$
w^t	Model parameters in t-th training round
w_i^t	Model parameters of DC i in t-th training round
$l(w, z)$	Loss function with parameters w on sample z
b_i	Number of bits used for quantization in device i
b_{min}	Min number of quantization bits used in training
thr	Prediction threshold value of $F(w)$
η^t	Learning rate applied in the t-th round
δ	Constant used to enforce global parameter synchronization in fixed training intervals
g	Gradient calculated as $g = \nabla l(w, z)$
g_i^t	Original gradient in DC i in t-th training round
$\hat{g}_i^t$	Quantized gradient in DC i in t-th training round
$\hat{g}^t$	Globally aggregated quantized gradient across all DCs in t-th training round
$Q_s(g)$	Quantization function of gradient g at level s
B_i	Bandwidth between DC i and PS
B_{min}	Minimum bandwidth between DCs and PS
q_i	Quantization ratio used by DC i
q_{min}	Minimum quantization ratio in the training process
Δ	Gradient step size
M^2	Variance upper bound of the gradient
τ	Total communication time in one round

In cross-organizational federated learning in WANs, dynamic network heterogeneity exacerbates the straggler problem, where certain nodes become system bottlenecks due to network limitations, negatively impacting overall FL performance [12], [13], [14]. Existing methods fail to adequately account for the need for personalized quantization strategies in dynamic and heterogeneous network environments and utilize available bandwidth of individual nodes when making quantization adjustment.

As shown above, existing studies exhibit limitations in addressing the challenges posed by WAN heterogeneity, particularly in balancing communication efficiency and model accuracy in dynamic environments. This motivates our work to take account the node heterogeneity in dynamic network environments and adaptively adjust gradient transmission strategies based on network bandwidth to optimize FL performance by balancing communication efficiency and model accuracy.

III. Preliminaries

A. Problem Formulation

In this section, we provide a detailed definition of the problem to be addressed in the context of cross-organizational federated learning and present the corresponding system model. To facilitate understanding, key notations used throughout this study are summarized in Table I.

In a cross-organizational federated learning system, all participating DCs use the same machine learning model for training[15], [16]. These models share the same loss function $l(w, z)$, where $w \in \mathbb{R}^d$ represents the model parameters (variables to be optimized), z is a sample randomly drawn from the dataset D.

Assume the system contains N DCs participating in training, and DC$_i$ has a local dataset D_i. These datasets satisfy the following conditions:D_i is a partition of the global dataset D, that is: $\bigcup_{i=1}^{N} D_i = D$. This indicates that the data held by all participants together covers the entire global dataset. Different data partitions D_i and D_j are mutually exclusive. Specifically, for $i \neq j$, we have $D_i \cap D_j = \emptyset$.

Under this context, all DCs use their local datasets D_i to participate in training. However, the goal is to collaboratively optimize a unified global model w. To train this global model, the practical problem to solve is as follows [17]:

$$\min_{w} F(w) = \mathbb{E}_{z \sim D}[l(w, z)] \tag{1}$$

where $F(w)$ is the global objective function, representing the expected loss over all samples z. The goal is to minimize this expected loss by adjusting the model parameters w.

In this work, since the data is stored in a distributed manner and each data center holds an independent subset D_i, the PS cannot directly access the complete global dataset D. Consequently, a distributed optimization algorithm is required to minimize the objective function. In this paper, we employ Distributed Stochastic Gradient Descent (SGD) [6] to find the optimal parameters w^* that minimize the objective function $F(w)$ presented in Eq. (1).

In the t-th round of training, each data center DC_i computes the local gradient g_t^i using its local dataset D_i by the following formula [18]:

$$g_t^i = \frac{1}{|D_i|} \sum_{z \in D_i} \nabla_w l(w_t^i, z) \tag{2}$$

where $|D_i|$ represents the number of samples in the local dataset at data center i, and $\nabla_w l(w_t^i, z)$ denotes the gradient of the loss function with respect to the current model parameter w_t^i for sample z. The computation of local gradients is performed entirely within each respective node to ensure that data privacy is not compromised.

Due to the high communication cost between nodes in distributed systems, the locally computed gradient g_t^i is quantized to reduce communication bandwidth consumption. We design the random quantization method in this work based on the approach presented in [19], which is an advanced quantization technique. The quantization process is represented by the function $Q_s(g_t^i)$, and the quantized gradient is denoted as $\tilde{g}_t^i$. The specific implementation of the quantization method will be discussed in subsequent sections.

After aggregating the quantized gradients $\tilde{g}_t^i$ from all data centers, the global aggregated gradient $\bar{g}_t = \frac{1}{N} \sum_{i=1}^{N} \tilde{g}_t^i$. Through the aggregation operation, the computational contributions of all data centers in the distributed system are integrated into the global gradient $\bar{g}_t$. The cross-organizational federated learning system utilizes the globally aggregated gradient $\bar{g}_t$ received by the parameter server to update the global model parameter. The update formula for the model parameters is as follows:

$$w_{t+1} = w_t - \eta_t \cdot \bar{g}_t \tag{3}$$

In federated learning systems, the training process typically concludes when the loss function value $F(w_T)$ of the model reaches a predefined threshold thr. This threshold is used to measure whether the performance of the global model meets expectations. Let τ_t^i represent the communication time between data center DC_i and PS during the t-th training round. To improve communication efficiency during training and ensure model performance, we need to minimize the total communication time of the data center with the longest communication time across all training rounds, thereby reducing the impact of communication bottlenecks. Mathematically the problem can formulated as follows:

$$\min \sum_{t=1}^{T} \max_{i \in [1,N]} \tau_t^i, \quad \text{s.t.} \quad F(w_T) \leq \text{thr} \tag{4}$$

Because data centers typically possess substantial computational power, local computation time can be deemed negligible compared to communication time.

Given that communication delays constitute the primary bottleneck in cross-organizational federated learning systems, the method proposed in this paper focuses on minimizing communication delays. This ensures that the federated learning system can achieve efficient model training in distributed environments.

B. Hierarchical Federated Learning

Considering that data in cross-organization federated learning is not only distributed across different DCs but may also follow a distributed learning structure within each organization, the combination of such federated learning with edge computing systems represents a practical application scenario of hierarchical federated learning [20], [21]. In this context, different layers of the hierarchical structure correspond to various levels of nodes within the edge computing system.

Lumin et al. [22] proposed an efficient distributed training framework for FL. Based on the framework of [22], we adopt the hierarchical training architecture depicted in Figure 1. The upper layer consists of a Global Parameter Server (GPS) and distributed DCs, which are responsible for managing global model parameters across organizations and coordinating the training process. In the lower layer, each DC is further divided into a Local Parameter Server (LPS) and multiple Clients. The LPS acts as the coordinator within the data center, aggregating computation results from the clients and communicating with the GPS.

As is shown in Figure 1,the training process of cross-organizational federated learning involves the

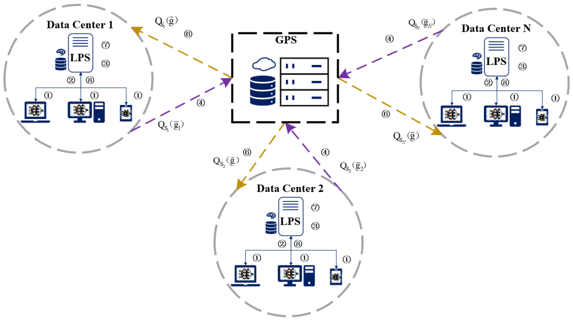


Fig. 1. Training Architecture of Hierarchical Cross-Organizational FL

following steps: 1) Client servers calculate the gradients based on their local datasets, ensuring that the computation reflects the data available;

2) Once local gradients are computed, client servers transmit them to the LPS;

3) Upon receiving the gradients, the LPS aggregates them from all client servers;

4) The LPS sends the quantized gradients, derived from local aggregation, to the GPS;

5) Upon reception, the GPS aggregates and applies quantization to the gradients;

6) Following this, the LPS updates its local model accordingly;

7) Client servers update their local models for subsequent operations.

IV. Gradient Quantization Level Dedermination

This section describes the core process of stochastic gradient quantization andbandwidth-aware adjustment mechanism used in our algorithm.

A. Stochastic Gradient Quantization Process

Assume that $Q_s(\cdot)$ is the operator for the quantization process, where s represents the quantization level, indicating the discretized intervals of the ladder values. For a non-zero vector $\mathbf{g}$ with quantization level s, $\forall \mathbf{g} \in \mathbb{R}n, \mathbf{g} \neq \mathbf{0}$.

The quantization process $Q_s(\mathbf{g})$ is defined as [23]:

$$Q_s(g_i) = \|\mathbf{g}\|_p \cdot \text{sign}(g_i) \cdot \xi(g_i, s) \tag{5}$$

where g_i ris the i-th component of the vector $\mathbf{g}$, $\|\mathbf{g}\|_p$ is the l_p norm of the gradient $\mathbf{g}$ used to normalize the magnitude of the gradient, $\text{sign}(g_i)$ is the sign function of the gradient element g_i used to retain directional information of the gradient (positive or negative), $\xi(g_i, s)$ is an independent random variable used to perform stochastic quantization of the normalized unsigned gradient value.

$\xi(g_i, s)$ maps the normalized gradient magnitude $\frac{|g_i|}{\|\mathbf{g}\|_p}$ into the discrete quantization interval defined as:

$$\xi(g_i, s) = \begin{cases} \frac{l+1}{s}, & \text{with probability } s \cdot \frac{|g_i|}{\|\mathbf{g}\|_p} - l \\ \frac{l}{s}, & \text{otherwise} \end{cases} \tag{6}$$

where l is the lower bound of the quantization interval, satisfying $0 \leq l < s$ and $\frac{|g_i|}{\|\mathbf{g}\|_p} \in \left[\frac{l}{s}, \frac{l+1}{s}\right]$. s represents the quantization level (number of intervals), which determines the precision of the quantized values. $\frac{|g_i|}{\|\mathbf{g}\|_p}$ is the normalized gradient magnitude, reflecting the relative importance of the gradient component g_i within the overall gradient $\mathbf{g}$.

From a theoretical perspective, the introduction of quantization methods serves as a compression technique aimed at reducing communication costs. However, without proving its fairness, the validity and correctness of optimization algorithms under such compression cannot be guaranteed.

B. Quantization Level Determination

To effectively control communication data volume in a federated learning system, we need to dynamically determines an appropriate quantization level s_i for each DCi. s_i determines the discretization degree of the gradients and the precision of data representation, which directly impacts the quantization bit-width b_i of each gradient component during communication. BA-DAGQ adjusts s_i in real-time to accommodate bandwidth variations across various DCs and the GPS, with the objective of ensuring uniform data transmission times between the GPS and each DC, all while preserving the overall training accuracy at a global level. Let n_g be the number of gradients.

To achieve the same communication time T across all DCs, it is sufficient to set b_i of DC$_i$ to satisfy $T_i = \frac{b_i \times n_g}{B_i}$. That is $b_i = \frac{T_i B_i}{n_g}$. Because $T_i \geq 1$, we set $T_i = \alpha$ for all i, which is determined by system requirement. So we have

$$b_i = \lceil \frac{\alpha B_i}{n_g} \rceil, \ 1 \leq i \leq N. \tag{7}$$

In gradient quantization, an unsigned gradient can be discretized into $2s_i + 1$ signed values. The required bit-width b_i satisfies:

$$b_i = \lceil \frac{\alpha B_i}{n_g} \rceil = \lceil \log_2(2s_i + 1) \rceil. \tag{8}$$

which demonstrates a direct connection between the gradient discretization precision and the communication cost. The quantization level s_i can be determined by Eq.(10):

$$s_i = 2^{\lceil \frac{\alpha B_i}{n_g} \rceil - 1} - 1. \tag{9}$$

In a WAN environment, because bandwidth B_i varies dynamically and communication links are asymmetric, each DC measures its uplink and downlink band-width separately.

To accurately capture bandwidth variation trends and adjust the quantization strategy accordingly, the BA-DAGQ system incorporates **a lightweight bandwidth measurement model called the Probe Gap Model (PGM).** The probing packet mechanism of PGM can be periodically triggered to capture real-time bandwidth changes.

V. Algorithm Description

A. Framework of BA-DAGQ

The BA-DAGQ framework employs a three-tier parameter server architecture, where the upper layer consists of the GPS and multiple DCs. The lower layer includes the LPS and Client Servers within each DC, taking a single LPS as an example, the framework of BA-DAGQ is illustrated in Figure 2. Client Servers compute local gradients and upload them to the respective LPS in their DC. The LPS then aggregates the received gradients and applies the quantization method described in Chapter 4 to the aggregated gradients. The quantized gradients are subsequently sent to the GPS, which performs further global aggregation and quantization of the gradients, generating the updated global model parameters and transmitting them back to the LPS. The LPS retrieves the updated global gradients from the GPS, utilizes them to update the local model parameters, and subsequently broadcasts the updated model parameters to the Client Servers, which then employ these parameters to update their respective local models.

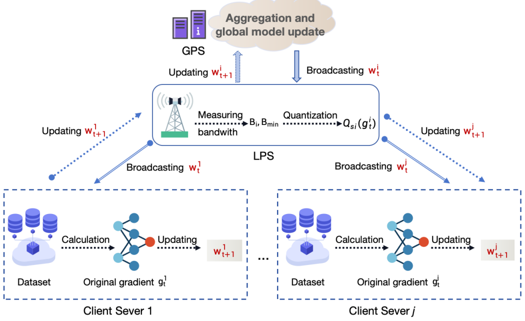


Fig. 2. Framework of BA-DAGQ

Communication is carried out over an LAN within each DC and the WAN environment between DC and GPS. Given the higher bidirectional bandwidth within DC, the communication cost between the LPS and Client Servers is not within the scope of optimization in this work. The primary focus of BA-DAGQ is to optimize the communication efficiency by dynamically adjusting the quantization bit-width to accommodate varying bandwidth conditions.

B. Algorithm of Client Servers, LPS and GPS

Algorithm 1 illustrates the workflow of client servers in the BA-DAGQ framework. Client servers retrieve initial model parameters from their LPS and partition the local dataset into mini-batches. During training, they compute gradients locally, upload them to the LPS, and update their model parameters using the aggregated results from the LPS for the next training round. This process ensures decentralized gradient computation and model updates in a hierarchical structure.

Algorithm 1 BA-DAGQ (Client Server j in DCi)

```
Input: $T, k, D_j$
Output: Updated local model parameters $w^{ij}_{t+1}$
Pull parameters $w^i_t$ from LPS of DC $i$;
$\mathcal{D} \leftarrow$ Divide $D_j$ into mini-batches of size $k$;
for $t$ in upper limit do
    for $d \in \mathcal{D}$ do
        Compute local gradient: $g^{ij}_t \leftarrow g^{ij}_t + \frac{1}{|d|}\sum_{z \in d} \nabla_{w^i_t} l(w^i_t, z)$;
    end for
    Upload $g^{ij}_t$ to LPS;
    Pull updated parameters $w^i_{t+1}$ from LPS;
    Update local parameters: $w^{ij}_{t+1} \leftarrow w^i_{t+1}$;
end for
```

Algorithm 2 BA-DAGQ (LPS i)

```
Input: $T, b_{\text{min}}$
Output: Updated parameters $w^i_{t+1}$ broadcast to client servers
Measure $B_i$ between DC$i$ and GPS using a lightweight probing algorithm (e.g., IGI);
Transmit $B_i$ to the GPS for further processing;
Retrieve the minimum quantization bits $B_{\text{min}}$ and the initial parameters $w_t$ from the GPS;
Broadcast $w_t$ to all client servers in DC $i$;
for $t = 1$ to $T$ do
    for $j = 1$ to $M_i$ do
        Collect $g^{ij}_t$ from lient server $j$;
    end for
    Aggregate gradients at LPS:$g^i_t \leftarrow \frac{1}{M_i}\sum_{j=1}^{M_i} g^{ij}_t$;
    Calculate quantization level $s_i$ using: $s_i \leftarrow 2^{b_i - 1} - 1, \quad b_i \leftarrow \left\lceil b_{\text{min}} \cdot \frac{B_i}{B_{\text{min}}} \right\rceil$;
    Quantize the aggregated gradient: $\tilde{g}^i_t \leftarrow Q_s(g^i_t)$;
    Upload $\tilde{g}^i_t$ and $B_i$ to GPS;
    if $t \mod \delta = 0$ then
        Pull $w_{t+1}$ from GPS;
        Update parameters at LPS: $w^i_{t+1} \leftarrow w_{t+1}$;
    else
        Retrieve the updated parameters $w_{t+1}$ and the minimum quantization bits from the GPS;
        Subsequently, update the parameters at the LPS using the formula: $w^i_{t+1} \leftarrow w^i_t - \eta_t \cdot \tilde{g}^i_t$;
    end if
    Broadcast $w^i_{t+1}$ to all client servers in DC $i$;
end for
```

Algorithm 2 shows how the LPS operates within a DC to manage bandwidth measurements between the DC and GPS and dynamically adjust quantization levels based on the available bandwidth: It collects gradients from client servers, aggregates them, and performs gradient quantization before sending the data to the GPS. It also updates local model parameters based on the GPS feedback and broadcasts them back to the client servers. This process balances local computation and global synchronization while adapting to network heterogeneity.

Algorithm 3 provides a detailed description of the training process carried out by the GPS in the BA-DAGQ framework. At the beginning of training, the GPS employs the IGI algorithm to assess the available bandwidth between itself and each DC, and it continuously updates this bandwidth information during the entire training process. The GPS initializes model parameters and broadcasts them to all LPSs, while continuously measuring bandwidth across all DCs. During training, it collects quantized gradients from LPS, computes the global gradient, and updates the global model parameters. Quantized updates are then sent back to LPSs. Periodic synchronization of global parameters ensures consistency across all data centers while adapting to bandwidth constraints to optimize communication and computation.

Algorithm 3 BA-DAGQ (GPS)

```
Input: $T$ (total iterations), $N$ (number of LPS), $b_{\min}$ (minimum quantization bits)
Output: Updated global parameters $w_{t+1}$
Measure bandwidth $B'_i$ (from GPS to each DC $i$) using a lightweight probing algorithm;
Collect $B_i$ from all LPS (bandwidth from LPS $i$ to GPS);
Compute: $B_{\min} \leftarrow \min(B_i), \quad B'_{\min} \leftarrow \min(B'_i)$;
Initialize $w_t$ with initial parameters;
Broadcast $B_{\min}, w_t$ to all LPS;
for $t = 1$ to $T$ do
    for $i = 1$ to $N$ do
        Pull quantized gradients $\tilde{g}_t^i$ and $B_i$ from LPS $i$;
    end for
    Update: $B_{\min} \leftarrow \min(B_i)$;
    Aggregate global gradients: $\bar{g}_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} \tilde{g}_t^i$;
    Retrieve the global parameters as Eq.(3);
    if $t \mod \delta = 0$ then
        Broadcast $w_{t+1}, B_{\min}$ to all LPS;
    else
        for $i = 1$ to $N$ do
            Calculate quantization level $s_i$ ;
            Quantize global gradients for LPS $i$: $\tilde{g}_t^i \leftarrow Q_s(\bar{g}_t)$;
            Push $\tilde{g}_t^i$ and $B_{\min}$ to LPS $i$;
        end for
    end if
    Update: $B'_{\min} \leftarrow \min(B'_i)$;
end for
```

C. Performance Analysis

Our three-layer BA-DAGQ adaptively adjusts the number of quantization bits for gradients, striking for a balance between communication cost and gradient transmission accuracy.

Specifically, let the total communication cost without quantization be denoted as $C_{\textit{no-q}}$. When each gradient component is quantized with b-bit precision, the gradient accuracy is denoted as P_b. The total communication cost and gradient accuracy of the BA-DAGQ method are represented by $C_{\textit{BA-DAGQ}}$ and $P_{\textit{BA-DAGQ}}$, respectively. Additionally, let N_g represent the total number of global gradient components transmitted per training round, and $Q_b(g)$ denote the b-bit quantization operation on gradient g.

- **Communication Cost Analysis:** In the non-quantized case, each gradient component is quantized with a bit-width of b_f, typically 32-bit floating-point. The total communication cost is $C_{\textit{no-q}} = N_g \cdot b_f$. The communication time is inversely proportional to the communication bandwidth B_i:$T_{\text{comm,no-q}} = \frac{C_{\text{no-q}}}{B_i} = \frac{N_g \cdot b_f}{B_i}$.

The BA-DAGQ method dynamically adjusts the quantization bit-width b_i for different data centers. The total communication cost is $C_{\textit{BA-DAGQ}} = \sum_{i=1}^{N} N_g \cdot b_i$. Here, b_i is the quantization bit-width for the i-th data center. The communication time is calculated as:$T_{\text{comm,BA-DAGQ}} = \max_i \left(\frac{N_g \cdot b_i}{B_i}\right)$. In this context, BA-DAGQ dynamically adjusts b_i such that all $T_{\text{comm,BA-DAGQ}}$ values remain consistent, thereby eliminating communication bottlenecks.

The communication cost reduction of BA-DAGQ compared to the non-quantized case is expressed as:

$$\textit{Reduction} = \frac{C_{\textit{no-q}} - C_{\textit{BA-DAGQ}}}{C_{\textit{no-q}}} = 1 - \frac{\sum_{i=1}^{N} N_g \cdot b_i}{N_g \cdot b_f} \tag{10}$$

If we assume that the average quantization bit-width of BA-DAGQ is b_{avg}, then *Reduction Ratio* will be $1 - \frac{b_{\text{avg}}}{b_f}$.

- **Gradient accuracy analysis:** In the BA-DAGQ method, the quantization bit-width b_i used by different DCs determines their precision levels. The overall precision level is the weighted average of the quantization precision across all DCs: $P_{\textit{BA-DAGQ}} = \frac{\sum_{i=1}^{N} b_i \cdot Q_{b_i}(g)}{\sum_{i=1}^{N} b_i}$.

For a fixed gradient quantization method, if $P_{\textit{fixed}}$ denotes the gradient transmission precision under fixed quantization, the precision improvement ratio of the BA-DAGQ method compared to fixed quantization can be expressed as:

$$Improvement = \frac{P_{\textit{BA-DAGQ}} - P_{\textit{fixed}}}{P_{\textit{fixed}}} \tag{11}$$

Therefore, the adaptive quantization bit adjustment mechanism of BA-DAGQ addresses the communication bottleneck in heterogeneous network environments, improving the accuracy of gradient transmission while ensuring communication efficiency.

VI. Simulation Experiment and Results Analysis

A. Experimental Setup

In the experiment, we simulate the hierarchical PS architecture mentioned in Section 3.2 using multiple processes. By configuring different path bandwidths between DCs and the GPS, we replicate a heterogeneous and dynamic wide-area network environment.

1) Simulation Platform Setup: The platform used for experimental evaluation is a server workstation running the Windows 11 operating system, equipped with an Intel Core i9 14900K processor and an NVIDIA® GeForce® GTX 4060 Ti graphics card, with 128GB of memory. The operating system is the Windows 11 Workstation version, and the simulation code is implemented using the Pytorch 2.3.0 framework.

In the experiment, we designed a system consisting of five DCs, each equipped with two client servers, one LPS and GPS to simulate the environment of cross-organizational federated learning training. To more accurately reflect the heterogeneity and dynamic characteristics of the wide-area network, the bandwidth between the DCs and the GPS was constrained to range from 60Mbps to 960Mbps. The initial bandwidth for each DC was set to 60, 120, 240, 480, and 960Mbps. The bandwidth varied within a specific range, with the differences implemented through corresponding CPU waiting times.

2) Dataset and Model: This study conducted experiments using three commonly used datasets: Fashion-MNIST[24], CIFAR-100[25], and Mini-ImageNet[26]. The Mini-ImageNet dataset was preprocessed following the method described in [23] to adapt it for classification tasks, dividing the data into training, validation, and test sets. For the Fashion-MNIST dataset, a custom-designed CNN consisting of two 5x5 convolution layers, each followed by a 2x2 max pooling layer, followed by a fully connected layer with 512 units and ReLU activation, and a final softmax output layer. For the CIFAR-100 and Mini-ImageNet datasets, the widely used ResNet[27] architecture was selected for experiments, including ResNet18 and ResNet101 versions.

For the experiments, the mini-batch size was initially set to 132, and the weights and biases were initialized randomly using a Gaussian distribution with a mean of 0 and a variance of 0.1. The learning rate was set to 1×10^{-3}. For the Fashion-MNIST dataset, the training process was conducted for a total of 100 epochs, while for the CIFAR-100 and Mini-ImageNet datasets. The constant δ was configured to 10, it represents the interval of global parameter synchronization.

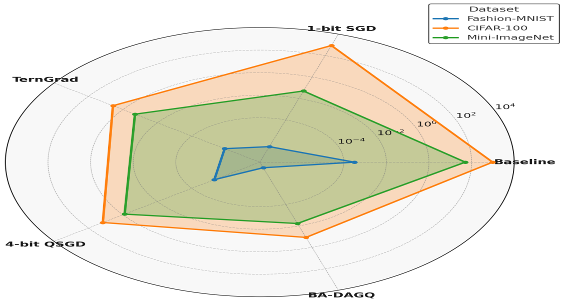


Fig. 3. Radar chart of communication times of different quantization schemes

3) Comparative Schemes and Performance Metrics: This section introduces three comparative algorithms to evaluate the effectiveness of the BA-DAGQ method: ①Baseline method: A standard distributed system without gradient quantization or any optimization strategies. ② [28]: Utilizes 1-bit quantization for each gradient to minimize transmission bandwidth consumption. ③ [6]: Employs a 2-bit quantization scheme for gradients transmitted from DCs to the GPS, while keeping the 32-bit parameters returned by the GPS uncompressed. ④ [5]: Reduces gradient precision to 4-bit representations for efficient communication.

We use three metrics to measure the performance of the BA-DAGQ method in cross-organizational federated learning: ①communication time variance; ② model Top-1 accuracy; ③convergence time. The variance in communication time between different DCs and the GPS indicates the varying communication costs across DCs, providing insights into the extent of the straggler problem. Model test accuracy represents the ratio of correctly classified samples after inference to the total number of samples in the test dataset, and is typically used to measure the model's generalization ability. For convergence time, the convergence criterion is set as follows: if the objective value changes by less than 1% over 20 consecutive training epochs, the model is considered converged. To ensure consistent convergence results, the same initial model is used for each algorithm in different experiments.

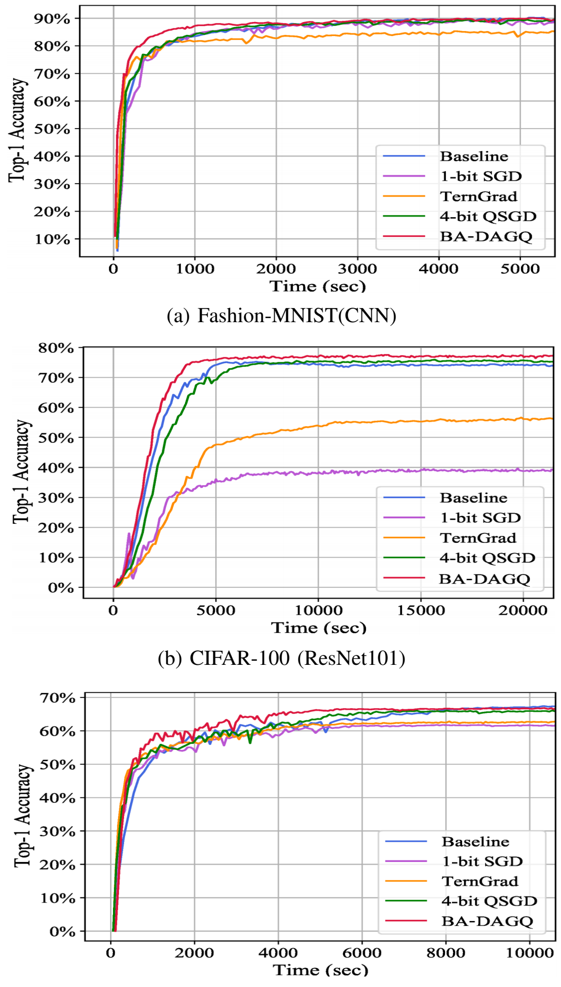
B. Experimental Results and Analysis

This section validates the BA-DAGQ method based on the experimental settings described above. The communication time variance between DCs and the GPS during one training cycle was analyzed for distributed training across different datasets. As shown in Figure VI-A2, it illustrates the logarithmic-scale radar chart of the communication time variance.

The experimental results demonstrate that BA-DAGQ achieves significantly lower communication time variance on the Fashion-MNIST dataset. Compared to the other four baseline methods, BA-DAGQ reduces the communication time variance. Similar performance improvements are observed on the CIFAR-100 and Mini-ImageNet datasets.

During local model training using the BA-DAGQ method, the communication times among DCs remain highly consistent within each training cycle. This demonstrates that BA-DAGQ's bandwidth-adaptive gradient quantization strategy effectively mitigates the straggler issue. Moreover, the method exhibits efficient utilization of bandwidth resources. Unlike other approaches, where faster DCs experience idle time waiting for stragglers, BA-DAGQ leverages the surplus bandwidth on high-speed paths by transmitting higher-precision gradients. Consequently, BA-DAGQ method proves to be particularly suitable for cross-organizational federated learning tasks in bandwidth-constrained wide-area network environments.

Figure 4 illustrates the curves of model Top-1 accuracy over time, with the dataset and training networks divided into three subplots. The experimental results



(a) Fashion-MNIST(CNN)

(b) CIFAR-100 (ResNet101)

(c) Mini-ImageNet (ResNet18)

Fig. 4. Comparison of Model Top-1 Accuracy Over Training Time

show that all curves exhibit an increase in model Top-1 accuracy over time during training. However, in each subplot, the red curve consistently lies above the others, indicating that BA-DAGQ converges the fastest among all quantization methods. Although in Figure 3(c), the growth rate of the 4-bit QSGD curve initially resembles that of BA-DAGQ, ourmethod achieves the same Top-1 accuracy more quickly than the 4-bit QSGD and surpasses it in accuracy by the end of the training. Furthermore, because BA-DAGQ transmits less traffic over bandwidth-limited links, it places less burden on the network compared to the other four methods. Overall, the BA-DAGQ method significantly reduces the communication time per epoch and minimizes gradient precision loss.

The convergence time speedup of BA-DAGQ compared to other quantization methods across three datasets is shown in Figure 5. BA-DAGQ achieves a 4.80× speedup over the baseline on the Fashion-MNIST dataset, a 1.77× speedup on CIFAR-100 (ResNet101), and a 1.57× speedup on Mini-ImageNet (ResNet18). When compared to other quantization methods, BA-DAGQ demonstrates acceleration factors ranging from 1.18× to 21.31× over 1-bit SGD, 1.19× to 7.10× over TernGrad, and 1.13× to 3.17× over 4-bit QSGD. This remarkable performance improvement is attributed to BA-DAGQ's ability to reduce communication time and efficiently utilize bandwidth resources, thereby achieving superior training performance and faster convergence within fewer training cycles.

Table II summarizes the numerical results Table 1 summarizes the numerical results from training various models on their respective datasets. The table primarily compares the differences in Top-1 accuracy and convergence time speedup between BA-DAGQ and other quantization methods.

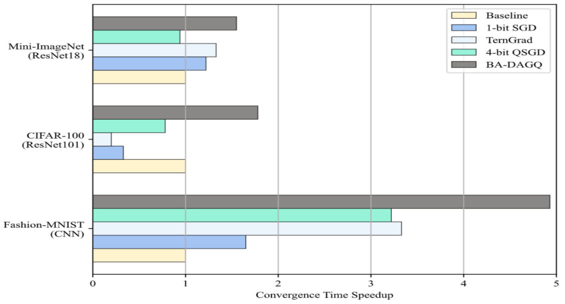


Fig. 5. Results of convergence time speedup comparison

The experimental results demonstrate that BA-DAGQ exhibits superior performance across all three datasets. By optimizing communication efficiency and bandwidth utilization, it successfully reduces communication time while maintaining model performance, providing a viable solution for achieving more efficient cross-organizational federated learning in distributed training environments.

VII. Conclusion

In this paper, we proposed a bandwith-aware adaptive gradient quantization algorithm (BA-DAGQ) to address the communication bottlenecks and heterogeneous bandwidth challenges in cross-organization federated learning (FL) networks. BA-DAGQ enables flexible gradient quantization across data centers according to real-time available network bandwidth and dynamic adjustment of discretization bit-width to synchronize communication times to achieve a desirable balance between communication cost and gradient accuracy. We demonstrate the supriority of our BA-DAGQ method to existing work through extensive empirical evaluations on FL datasets and show that BA-DAGQ can significantly reduce communication overhead while maintaining high model accuracy.

Acknowledgments

This work is supported by Science and Technology Development Fund of Macao (FDCT) (Grant No. #0033/2023/RIA1), Department of Environment and Science of Queensland State Government under Quantum Challenge 2032 Program (Project #Q2032001), and Macao Polytechnic University (File

TABLE II
COMPARISON OF TOP-1 ACCURACY AND CONVERGENCE TIME ACROSS DIFFERENT METHODS

Training Dataset and Model	Quantization Method	Top-1 Accuracy	Convergence Time Speedup
Fashion-MNIST (CNN)	Baseline	91.33%	1.00×
	1-bit SGD	90.76%(-0.57%)	1.56×
	TernGrad	86.95%(-4.38%)	3.29×
	4-bit QSGD	91.12%(-0.21%)	3.17×
	BA-DAGQ	**91.48%(+0.15%)**	**4.80**×
CIFAR-100 (ResNet101)	Baseline	75.41%	1.00×
	1-bit SGD	40.32%(-35.09%)	0.37×
	TernGrad	56.75%(-18.66%)	0.28×
	4-bit QSGD	75.90%(+0.49%)	0.76×
	BA-DAGQ	**77.86%(+2.45%)**	**1.77**×
Mini-ImageNet (ResNet18)	Baseline	67.95%	1.00×
	1-bit SGD	61.83% (-6.12%)	1.18×
	TernGrad	62.76%(-5.19%)	1.32×
	4-bit QSGD	65.13%(-2.82%)	0.91×
	BA-DAGQ	**67.60%(-0.35%)**	**1.57**×

No. fca.732d.c58c.6). The correcponding author is Hong Shen.

REFERENCES

[1] D. Narayanan, A. Harlap, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, P. B. Gibbons, and M. Zaharia, "Pipedream: Generalized pipeline parallelism for dnn training," in *Proceedings of the 27th ACM symposium on operating systems principles*, pp. 1–15, 2019.

[2] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons, and O. Mutlu, "Gaia:{Geo-Distributed} machine learning approaching {LAN} speeds," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pp. 629–647, 2017.

[3] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[4] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*, pp. 1737–1746, PMLR, 2015.

[5] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, pp. 560–569, PMLR, 2018.

[6] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in neural information processing systems*, vol. 30, 2017.

[7] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] S. Horvóth, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtárik, "Natural compression for distributed deep learning," in *Mathematical and Scientific Machine Learning*, pp. 129–141, PMLR, 2022.

[9] J. Sun, T. Chen, G. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[10] Y. Yu, J. Wu, and J. Huang, "Exploring fast and communication-efficient algorithms in large-scale distributed networks," *arXiv preprint arXiv:1901.08924*, 2019.

[11] N. Ivkin, D. Rothchild, E. Ullah, I. Stoica, R. Arora, *et al.*, "Communication-efficient distributed sgd with sketching," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] L. Li, Y. Fan, and K.-Y. Lin, "A survey on federated learning," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, pp. 791–796, IEEE, 2020.

[13] Q. Cheng and G. Long, "Federated learning operations (flops): Challenges, lifecycle and approaches," in *2022 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 12–17, IEEE, 2022.

[14] W. Yin, Y. Sun, X. Li, and R. Song, "A novel cross-organization privacy protection method based on federated learning," in *Proceedings of the 2023 6th International Conference on Information Management and Management Science*, pp. 69–73, 2023.

[15] Y. Zheng, Z. Cheng, Y. Liu, B. Wang, and C. Zhu, "Collaborative learning for cross-organizational data sharing using hyperledger fabric," in *2023 8th International Conference on Data Science in Cyberspace (DSC)*, pp. 285–292, IEEE, 2023.

[16] W. M. Van der Aalst, "Federated process mining: exploiting event data across organizational boundaries," in *2021 IEEE International Conference on Smart Data Services (SMDS)*, pp. 1–7, IEEE, 2021.

[17] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[18] T. Murata and T. Suzuki, "Bias-variance reduced local sgd for less heterogeneous federated learning," *arXiv preprint arXiv:2102.03198*, 2021.

[19] N. Lang, E. Sofer, T. Shaked, and N. Shlezinger, "Joint privacy enhancement and quantization in federated learning," *IEEE Transactions on Signal Processing*, vol. 71, pp. 295–310, 2023.

[20] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12806–12825, 2021.

[21] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[22] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, 2020.

[23] X. Cao, "Mlclf: The project machine learning classification for utilizing mini-imagenet and tiny-imagenet," 2022.

[24] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[25] R. Moradi, R. Berangi, and B. Minaei, "Sparsemaps: convolutional networks with sparse feature maps for tiny image classification," *Expert Systems with Applications*, vol. 119, pp. 142–154, 2019.

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[27] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022.

[28] 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*, vol. 2014, pp. 1058–1062, Singapore, 2014.