Adaptive Federated Optimization via Dynamic Gradient Quantization for Heterogeneous Edge Networks

Alex Rivera*, Jordan Chen†, Morgan Vance‡, and Taylor Reed*
*Department of Distributed Intelligence, Nexa AI Research, San Francisco, CA, USA
Email: {a.rivera, t.reed}@nexa.ai
†School of Computer Science, Meridian Institute of Technology, Boston, MA, USA
Email: j.chen@meridian.edu
‡Edge Systems Division, Vertex Laboratories, Seattle, WA, USA
Email: m.vance@vertexlabs.io

***Abstract*—Federated Learning (FL) enables geographically distributed edge devices to collaboratively train machine learning models while retaining raw training data locally. However, communication overhead remains a primary bottleneck when deploying FL models over low-bandwidth, unreliable wireless edge networks. In this paper, we propose DYNQUANT-FL, a dynamic gradient quantization framework tailored for resource-constrained edge computing environments. By dynamically adjusting quantization precision on a per-layer basis according to local gradient variance, network latency metrics, and available channel capacity, DYNQUANT-FL achieves up to an 84.2% reduction in total uplink data transfer without sacrificing final model accuracy. We validate our framework across a 32-node hardware testbed comprising Nexa Edge-V1 modules and Apex embedded units under non-IID data distributions and fluctuating channel profiles. Experimental results demonstrate that DYNQUANT-FL accelerates global convergence by $2.35\times$ compared to standard FedAvg and outperforms static quantization baselines by up to 14.8% in top-1 accuracy under severe bandwidth constraints (< 1 Mbps).**

***Index Terms*—Federated learning, gradient quantization, edge computing, distributed optimization, adaptive compression, heterogeneous networks.**

I. INTRODUCTION

The proliferation of Internet of Things (IoT) sensors, autonomous robotics, and mobile devices has led to exponential growth in telemetry data generated at the network edge. Traditional centralized machine learning paradigms require transferring massive datasets from edge nodes to centralized cloud datacenters. This centralized model poses severe scalability challenges due to backhaul network congestion, strict data privacy regulations, and unacceptable communication latencies [1], [2].

Federated Learning (FL) has emerged as a compelling privacy-preserving paradigm by shifting model training from central servers to distributed edge clients [3]. In FL, edge devices compute local model updates on their native datasets and periodically transmit parameter vectors to a central coordinator. The server aggregates these updates—typically via Federated Averaging (FedAvg)—to produce a refined global model, which is subsequently redistributed to the participating clients.

Despite its advantages, deploying FL in real-world wireless edge networks encounters fundamental operational obstacles:

1) **High Communication Overhead:** Deep neural network models frequently contain tens to hundreds of millions of parameters. Transmitting full-precision 32-bit floating-point ($fp32$) gradient updates across hundreds of communication rounds severely degrades throughput on bandwidth-constrained links.
2) **System and Channel Heterogeneity:** Edge nodes possess highly diverse compute capabilities, memory limits, and uplink bandwidth profiles. Devices on congested cellular networks or low-power wireless links act as stragglers, delaying global synchronization rounds.
3) **Statistical Data Heterogeneity:** Real-world edge data distributions are inherently Non-Independent and Identically Distributed (non-IID). Statistical skew causes local gradient vectors to diverge substantially across clients, rendering static quantization schemes prone to severe error accumulation and training instability.

To address these interconnected challenges, we propose **DYNQUANT-FL** (Dynamic Quantization for Federated Learning), an adaptive compression framework designed specifically for heterogeneous edge networks. Unlike conventional quantization approaches that impose static bit-widths across all network layers and training epochs, DYNQUANT-FL continuously monitors empirical gradient variance and channel state information (CSI) to allocate optimal quantization precision dynamically.

A. Key Contributions

The main contributions of this paper are summarized as follows:

- **Adaptive Rate-Distortion Formulation:** We model gradient compression as a joint optimization problem balancing

This research was supported in part by the Meridian Science Foundation under Grant No. FX-2025-8810, the Apex Autonomous Systems Research Initiative, and Vertex Systems Lab.

quantization noise and link transmission delay under time-varying network conditions.

- **Variance-Driven Quantization Scheme:** We introduce a low-overhead, layer-wise quantization algorithm executed locally on edge clients with minimal computational complexity ($< 1.2\%$ CPU overhead).
- **Realistic Edge Testbed Validation:** We construct a physical testbed comprising 32 edge devices (Nexa Edge-V1 boards and Apex embedded modules) connected via real-time network emulation.
- **Extensive Empirical Evaluation:** We demonstrate that DynQuant-FL reduces cumulative uplink communication by 84.2% on CIFAR-10, FEMNIST, and Shakespeare benchmarks while matching uncompressed baseline accuracy.

II. Related Work

A. Communication-Efficient Federated Learning

Gradient compression methods in distributed machine learning broadly fall into two main categories: sparsification and quantization [4], [5]. Sparsification algorithms (e.g., Top-k and Random-k selection) transmit only a subset of gradient components exceeding a defined magnitude threshold [6]. Although sparsification substantially reduces vector size, it requires local error-feedback accumulators to track un-transmitted updates, incurring additional memory overhead on RAM-constrained edge hardware.

Quantization approaches map continuous floating-point gradient values into low-precision discrete levels. Quantized SGD (QSGD) [4] established stochastic k-level quantization with theoretical convergence bounds under IID settings. SignSGD [5] compressed updates to 1-bit signs, but exhibits severe accuracy degradation when applied to complex non-IID edge datasets.

B. Adaptive Quantization in Edge Systems

Recent literature has explored adaptive compression to mitigate straggler effects in distributed systems [7], [8]. Systems like FedPAQ [2] incorporate static 8-bit quantization with periodic client sampling. However, existing schemes generally assume uniform communication channels across all participating clients and ignore real-time network variations. In contrast, DynQuant-FL unifies gradient variance metrics with physical network telemetry to achieve fine-grained, per-layer bit-width adaptation.

III. System Architecture and Methodology

A. Problem Formulation

Consider an edge learning system comprising a central aggregation server and K distributed edge clients. Each client $k \in \{1, \dots, K\}$ retains a local dataset $\mathcal{D}_k$ of size $N_k = |\mathcal{D}_k|$. The objective of federated learning is to solve the global empirical risk minimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) \equiv \sum_{k=1}^{K} \frac{N_k}{N} f_k(\mathbf{w}), \tag{1}$$

where $f_k(\mathbf{w}) = \frac{1}{N_k} \sum_{i \in \mathcal{D}_k} \ell(\mathbf{w}; x_i, y_i)$ represents the empirical loss over client k's dataset for parameter vector $\mathbf{w} \in \mathbb{R}^d$.

At global round t, the central server selects a subset of active clients $\mathcal{K}_t \subseteq \mathcal{K}$ and broadcasts the current model weights $\mathbf{w}^{(t)}$. Each selected client performs E local training epochs using stochastic gradient descent (SGD), yielding updated local weights $\mathbf{w}_k^{(t,E)}$. The net parameter update vector is defined as:

$$\mathbf{\Delta}_k^{(t)} = \mathbf{w}^{(t)} - \mathbf{w}_k^{(t,E)}. \tag{2}$$

B. Stochastic Layer-Wise Quantization

Instead of transmitting full-precision 32-bit representations of $\mathbf{\Delta}_k^{(t)}$, client k applies a stochastic quantization operator $Q_b(\cdot)$ parameterized by allocated bit-depth b. For a scalar update value $v \in \mathbf{\Delta}_{k,l}^{(t)}$ belonging to model layer $l \in \{1, \dots, L\}$, the quantization function is defined as:

$$Q_b(v) = \text{sign}(v) \cdot \|\mathbf{\Delta}_{k,l}^{(t)}\|_2 \cdot \frac{1}{s} \left\lfloor s \cdot \frac{|v|}{\|\mathbf{\Delta}_{k,l}^{(t)}\|_2} + \xi \right\rfloor, \tag{3}$$

where $s = 2^{b-1} - 1$ represents the discrete scale levels, and $\xi \sim U(0,1)$ is an independent uniform random variable. Equation (3) ensures that the quantization operator is strictly unbiased:

$$\mathbb{E}\left[Q_b(\mathbf{\Delta}_{k,l}^{(t)})\right] = \mathbf{\Delta}_{k,l}^{(t)}. \tag{4}$$

The expected quantization variance introduced into layer l of dimension d_l scales inverse-proportionally with the number of quantization levels:

$$\mathbb{E}\left[\|Q_b(\mathbf{\Delta}_{k,l}^{(t)}) - \mathbf{\Delta}_{k,l}^{(t)}\|_2^2\right] \leq \frac{d_l \cdot \sigma_{k,l}^2}{2^{2b-2}}, \tag{5}$$

where $\sigma_{k,l}^2 = \frac{1}{d_l}\|\mathbf{\Delta}_{k,l}^{(t)}\|_2^2$ is the empirical variance of the layer's update elements.

C. Joint Optimization of Bit Allocation

To minimize quantization variance while adhering to client-specific communication latency constraints, client k solves a localized resource allocation problem at round t:

$$\min_{\{b_{k,l}\}} \sum_{l=1}^{L} \frac{d_l \cdot \sigma_{k,l}^2}{2^{2b_{k,l}}} + \gamma \cdot \frac{\sum_{l=1}^{L} d_l \cdot b_{k,l}}{C_k^{(t)}}, \tag{6}$$

$$\text{subject to } b_{k,l} \in \{2, 4, 8, 16\}, \quad \forall l \in \{1, \dots, L\}, \tag{7}$$

where $C_k^{(t)}$ is the real-time uplink bandwidth (Mbps) measured at client k, and $\gamma > 0$ is a scalar hyperparameter tuning the trade-off between quantization error and transmission latency.

Using a continuous relaxation of Eq. (6), the optimal unconstrained bit-width $b_{k,l}^*$ for layer l is derived analytically as:

$$b_{k,l}^* = \frac{1}{2} \log_2 \left(\frac{2 \ln(2) \cdot \sigma_{k,l}^2 \cdot C_k^{(t)}}{\gamma} \right). \tag{8}$$

Client k then discretizes $b_{k,l}^*$ to the nearest valid level in $\{2, 4, 8, 16\}$, ensuring low computational overhead during local bit allocation.

Parameter / Notation	System Description
$K, \mathcal{K}$	Total number of edge clients and set of active participants, $\mathcal{K} = \{1, 2, \ldots, K\}$
$\mathcal{D}_k, N_k$	Local dataset and sample count held by client k, where $N = \sum_{k=1}^{K} N_k$
$\mathbf{w}^{(t)} \in \mathbb{R}^d$	Global model weight vector at communication round t
$\mathbf{\Delta}_k^{(t)} \in \mathbb{R}^d$	Accumulated uncompressed update vector computed by client k after E local epochs
$b_{k,l}^{(t)} \in \{2, 4, 8, 16\}$	Dynamically allocated quantization bit-width for layer l at client k during round t
$\sigma_{k,l}^2$	Empirical sample variance of gradient elements in layer l at client k
$C_k^{(t)}$	Real-time estimated uplink channel bandwidth (in Mbps) for client k
$\gamma > 0$	Penalty hyperparameter tuning compression aggressiveness versus transmission latency

Fig. 1. System Notation and Formulation Parameters for DYNQUANT-FL.

IV. EXPERIMENTAL EVALUATION

A. Hardware and System Setup

We evaluate DYNQUANT-FL on an empirical edge hardware testbed consisting of 32 physical edge nodes:

- 16× **Nexa Edge-V1 Modules**: Quad-core ARM Cortex-A72 processors with 8GB LPDDR4 memory.
- 16× **Apex Embedded Controllers**: Dual-core ARM Cortex-A53 devices with 4GB RAM simulating low-power IoT gateways.

The central server is hosted on a workstation with an Lumina RTX 4090 GPU and 64GB RAM. Bandwidth variability is simulated using Linux `tc` (traffic control) and `NetEm`, imposing link speeds ranging from 0.5 Mbps to 10.0 Mbps with random packet delay jitters (15 ± 5 ms).

B. Benchmarks and Datasets

We perform evaluations on three standard federated learning datasets representing vision and language domain workloads:

1) **CIFAR-10 / ResNet-18**: 50,000 training images partitioned across 32 clients using a Dirichlet distribution Dir(α) with concentration parameter $\alpha = 0.1$ (severe non-IID).
2) **FEMNIST / MobileNetV3**: 805,263 handwritten character images distributed across writer profiles.
3) **Shakespeare / Char-LSTM**: Next-character prediction task across 1,129 speaking roles.

C. Main Results and Accuracy Analysis

Table I summarizes performance metrics on CIFAR-10. Our DYNQUANT-FL scheme achieves a top-1 test accuracy of 81.2%, effectively matching uncompressed FedAvg (81.4%), while requiring only 0.44 GB total network uplink transfer over the entire training lifecycle.

Compared to static 8-bit quantization (FedPAQ), DYNQUANT-FL reduces cumulative uplink payload by 75.4% and accelerates convergence speed by $1.35\times$ (requiring 118 rounds instead of 160 to reach 75% accuracy). SignSGD fails to converge under severe non-IID conditions due to gradient sign flipping across divergent client datasets.

D. Dynamic Layer-Wise Bit Allocation

Table II details the layer-wise bit-width allocation chosen by DYNQUANT-FL during different phases of federated model training. In early communication rounds (R1–30), when gradients carry coarse directional information, the algorithm aggressively allocates 2-bit precision to intermediate residual blocks. In contrast, sensitive classifier layers retain higher precision (8–16 bit) throughout training to preserve decision boundary fidelity.

E. Robustness Under Extreme Network Bottlenecks

We further evaluate system throughput under severe wireless link degradation. When client bandwidth is throttled below 1.0 Mbps, static 32-bit and 8-bit schemes suffer from high client dropouts due to timeout errors. DYNQUANT-FL automatically compresses client updates down to 2-bit levels on low-bandwidth nodes, sustaining a 98.4% round completion rate compared to 62.1% for static FedPAQ.

V. DISCUSSION AND LIMITATIONS

While DYNQUANT-FL yields substantial communication savings, two practical trade-offs warrant consideration:

- **Bit-Packing Compute Costs:** Packing dynamic low-bit elements into packed byte buffers introduces a minor CPU overhead on microcontroller architectures lacking native bit-field instructions ($< 1.2\%$ total local training time).
- **Privacy Extensions:** High-rate quantization alters noise characteristics when combined with Differential Privacy (DP). Developing joint DP-quantization bounds represents an important direction for future research.

VI. CONCLUSION

We presented DYNQUANT-FL, a dynamic gradient quantization framework for federated learning in heterogeneous edge networks. By unifying empirical layer variance metrics with real-time channel state feedback, DYNQUANT-FL dynamically allocates bit-depth precision per layer and client. Extensive experiments on a 32-node physical testbed demonstrate an 84.2% reduction in total communication payload while preserving top-1 accuracy under non-IID conditions.

ACKNOWLEDGMENT

The authors thank the engineering teams at Nexa AI Research and Vertex Laboratories for access to hardware testbed infrastructure.

TABLE I
SYSTEM PERFORMANCE COMPARISON ON CIFAR-10 ($\alpha = 0.1$ DIRICHLET NON-IID)

Optimization Method	**Quantization Scheme**	**Top-1 Accuracy (%)**	**Uplink / Round (MB)**	**Rounds to 75% Acc.**	**Total Uplink (GB)**
Standard FedAvg [1]	Full Precision (32-bit)	81.4 ± 0.3	44.70	142	6.35
QSGD [4]	4-bit Uniform Static	76.2 ± 0.5	5.59	215	1.20
SignSGD [5]	1-bit Sign Static	69.8 ± 1.2	1.40	> 400 (Diverged)	–
FedPAQ [2]	8-bit Uniform Static	79.8 ± 0.4	11.18	160	1.79
Top-k Sparsified ($k = 5\%$)	32-bit Sparse	78.1 ± 0.6	4.47	190	0.85
DYNQUANT-FL (Ours)	**Adaptive (2–8 bit)**	$\mathbf{81.2 \pm 0.2}$	**3.74**	**118**	**0.44**

TABLE II
LAYER-WISE BIT-WIDTH ALLOCATION ACROSS TRAINING STAGES

Layer Group	**Early (R1–30)**	**Mid (R31–80)**	**Late (R81–120)**
Input Conv Stem	4-bit	8-bit	8-bit
ResNet Block 1–2	2-bit	4-bit	4-bit
ResNet Block 3–4	2-bit	2-bit	4-bit
FC Classifier	8-bit	8-bit	16-bit

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends Machine Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 1709–1720.

[5] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Machine Learn. (ICML)*, 2018, pp. 560–569.

[6] S. U. Stich, J. B. Cordonnier, and M. Jappi, "Sparsified SGD with memory," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 4447–4458.

[7] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource-constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[8] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 2020.