GradQ-ViT: Robust and Efficient Gradient Quantization for Vision Transformers

Dahun Choi, Hyun Kim*

Department of Electrical and Information Engineering, The Research Center for Electrical and Information Technology,
Seoul National University of Science and Technology, Seoul 01811, Korea
dahun926@seoultech.ac.kr, hyunkim@seoultech.ac.kr

Abstract

Advancements in hardware accelerators, such as graphics processing units and neural processing units, have significantly propelled computer vision research. The vision transformer (ViT), leveraging the multi-head self-attention (MHSA) mechanism, has surpassed convolutional neural networks (CNNs) in accuracy but faces challenges in mobile and edge deployment due to its large size and computational demands. In addition, as privacy concerns push for on-device training, research on quantization methods for ViTs, particularly gradient quantization, has gained attention. Unlike CNNs, ViTs face challenges due to outliers and a complex loss landscape. To address this, we propose a gradient quantization framework that stabilizes training by adapting quantization points based on interquartile ranges and constructing an outlier-robust loss function. Additionally, we employ a scaling method to align quantized gradients with original gradients and adaptively assign the learning rate based on quantization error analysis. When quantizing weights, activations, and gradients to INT8, our method improves performance by 0.52% and 0.21% over DeiT-Base and Swin-Base, respectively, and achieves near parity with MobileViT-S with only a 0.09% accuracy drop. Furthermore, a 2.06× speedup was observed when applying our framework to MobileViT in a CUDA 11.8 environment.

Introduction

With advancements in hardware accelerators, such as graphics processing units (GPUs) and neural processing units (NPUs), to support high-performance computing, deep neural network (DNN)-based computer vision (CV) tasks have become actively researched (He et al. 2016; Ding et al. 2021; Lee and Kim 2022). In particular, vision transformers (ViTs), which apply transformer blocks utilized in natural language processing to CV tasks, have attracted attention because they achieve significantly higher accuracy than convolutional neural networks (CNNs). However, ViTs rapidly increase the model size and computation cost owing to the multi-head self-attention (MHSA) with many parameters and high dimensionality, making it difficult to deploy ViT models on mobile/edge devices with limited hardware resources (Han et al. 2022; Lee et al. 2024). Consequently, compression methods such as quantization (Peng et al. 2023; Kim, Lee, and Kim 2024; Kang, Choi, and Kim 2024) and pruning (Kim and Kim 2022; Yang et al. 2023; Xu et al. 2024; Lee and Kim 2024) have been actively researched to embed ViT models into mobile/edge devices. Quantization, which can effectively solve the memory problem stemming from the increased computational costs and parameters of ViT, has attracted significant attention (Chen et al. 2024). Quantization is a compression method that converts 32-bit floating-point (FP) values into lower-precision (*e.g.*, 8-bit) hardware-friendly data formats (*e.g.,* fixed points and integers). Recent ViT quantization studies have primarily focused on post-training quantization (PTQ) (Lin et al. 2021; Yuan et al. 2022) for weight, activation, and MHSA because of the much higher training cost of ViTs compared to CNNs. However, the need for on-device training (Zhu et al. 2022; Lee, Lee, and Kim 2024; Chun, Lee, and Kim 2024) has recently emerged due to privacy issues, and federated learning (Ovi et al. 2023) to distribute the extreme computational load imposed on the cloud is also receiving great attention. Therefore, not only forward propagation (*i.e.,* weight, activation, MHSA) but also backward propagation (*i.e.,* gradient) quantization becomes important, and eventually, research on quantization-aware training (QAT) (Lee, Kim, and Ham 2021; Yao et al. 2021) of ViT to realize gradient quantization is essential. It is noteworthy that as small ViTs (Mehta and Rastegari 2021; Cai et al. 2022) have recently been proposed along with the development of dedicated hardware (You et al. 2023), an environment capable of performing QAT for ViTs with high accuracy is being realized based on CNN gradient quantization studies (Zhao et al. 2021).

Although gradient quantization can reduce computational costs and save memory during backpropagation, it also leads to training instability due to the low representational precision. The conventional nearest-to-rounding (NR) method (Lee, Kim, and Ham 2021) is a deterministic rounding approach that incurs minimal quantization error due to rounding. However, according to (Gupta et al. 2015), gradient quantization using NR may cause the network to diverge owing to biased gradients during training. Instead, prior studies (Zhu et al. 2020; Chmiel et al. 2021) employed stochastic rounding (SR) to eliminate the bias caused by quantization and thus stabilize training. Nonetheless, SR generates random numbers for each tensor, which is inappropriate for

*Corresponding Author

high-dimensional models such as ViTs. Furthermore, unlike weight and activation, gradients converge to zero as training progresses; therefore, the impact of outliers is significant, leading to significant errors in the quantization projection.

In this paper, we propose the GradQ-ViT framework to address training instability and quantization errors caused by outliers in ViT gradient quantization. The proposed method adaptively assigns precision based on the interquartile range (IQR) by considering the gradient distribution and redesigns the loss function to be robust to outliers. In addition, the learning rate is adaptively assigned during training, and the gradient scaling (GS) algorithm is used to scale gradient changes due to quantization, thereby stabilizing the training process. With weight(W)/activation(A)/gradient(G) quantized to 8-bit integer (INT8) form, GradQ-ViT achieved 0.52% and 0.21% higher accuracy than the baseline (*i.e.,* FP32) on the DeiT-Base (Touvron et al. 2021) and Swin-Base (Liu et al. 2021) models, respectively. We also succeeded in INT8 quantization on MobileViT-S, a lightweight hybrid ViT model, with only 0.09% accuracy degradation compared to the baseline. We implemented a custom kernel, which enables low-precision integer operations such as Matmul and convolution operations, in the CUDA 11.8 environment for the quantized MobileViT-S and, consequently, achieved 2.06× acceleration in inference and training compared to FP32. The contributions of this paper are as follows:

- We propose a new loss function that is robust to outliers in ViT gradient quantization by combining the cross entropy and Huber losses.
- We propose an IQR-based adaptive precision assignment method for ViT gradient quantization that considers the gradient distribution.
- We propose a GS algorithm to restore the ViT gradient changed by quantization after the dequantization process by scaling the magnitude and direction of the quantized gradient similarly to the original gradient.
- We adaptively assign the learning rate by measuring the similarity of ViT gradients and quantization error, thereby stabilizing training.

Background

Related Works on Gradient Quantization

Because backpropagation requires more computation and greater memory bandwidth than forward propagation, many gradient quantization studies have been conducted to reduce the training and resource costs incurred by backpropagation. In (Zhu et al. 2020), authors proposed a quantization solution that can be executed by converting operations used in CNNs to INT8 format with the objectives of maximizing computational efficiency and significantly reducing memory usage. To achieve this, they constructed a quantization framework that utilizes INT8 operations in both forward and backward propagation. In addition, to improve the accuracy of gradient quantization, they used a method to find appropriate clipping values based on cosine similarity. These values help minimize errors during the quantization process by constraining the gradient value within an appropriate range. The clipping method ensures the stability of training by minimizing the difference between the quantized and original gradients. Furthermore, they deployed SR methods to eliminate quantization bias. In (Zhao et al. 2021), the channel-wise distribution of the gradient was analyzed and found to be divided into a Gaussian distribution and an inverted-T distribution, which is important information when optimizing quantization parameters. Based on this analysis, we applied gradient vectorized quantization by deploying SR to eliminate network bias stemming from quantized gradients. This is an effective way to adjust precision by considering the distribution characteristics of each gradient and eliminating bias that may occur during the quantization process.

Challenges of gradient quantization in ViTs

The gradient plays a critical role in model training and performance optimization, as it is used in the backpropagation algorithm to update parameters and minimize the loss function. When employing gradient quantization, it is essential to minimize quantization error, as low-precision representations can distort the original information and potentially destabilize convergence during training. As shown in Figure 1, the gradient has a distribution with a very narrow range and large outliers. Previous studies have used clipping values to address projection errors caused by outliers in weight and activation quantization during forward propagation (Li, Dong, and Wang 2019). However, this approach has limitations in handling outliers during gradient quantization in backpropagation. Because these outliers significantly increase the gradient of the loss function, they require substantial adjustments when updating the parameters. Therefore, to mitigate projection errors caused by outliers, applying clipping may unintentionally hinder the update of parameters requiring significant adjustments, harming the training process. According to (Frumkin, Gope, and Marculescu 2023), the loss spaces of ViTs contain a greater number of local minima compared to those of CNNs. In such loss spaces, clipping methods can cause minor updates to become trapped in local minima, thereby hindering convergence. Additionally, gradient quantization during training can significantly impact learning rate adjustments and the behavior of optimizers. Therefore, a complex loss space presents challenges to gradient quantization. Gradient quantization during training can trap the model in local minima, causing suboptimal performance. To avoid this, the learning rate must be adaptively assigned based on the similarity between the original and quantized gradients.

Convergence Theory

Convergence theory describes whether a given model reaches the minimum loss function throughout training. Inspired by (Zhu et al. 2020), we adopt a convergence-theory-based approach to stabilize training in the presence of gradient quantization. To apply this theory, we assume the loss function $L(\theta)$ to satisfy the following conditions:

Assumption 1: $L(\theta)$ is a convex function

Assumption 2: $L(\theta) - L(\theta^*) \leq \frac{\beta}{2}||\theta - \theta^*||^2$

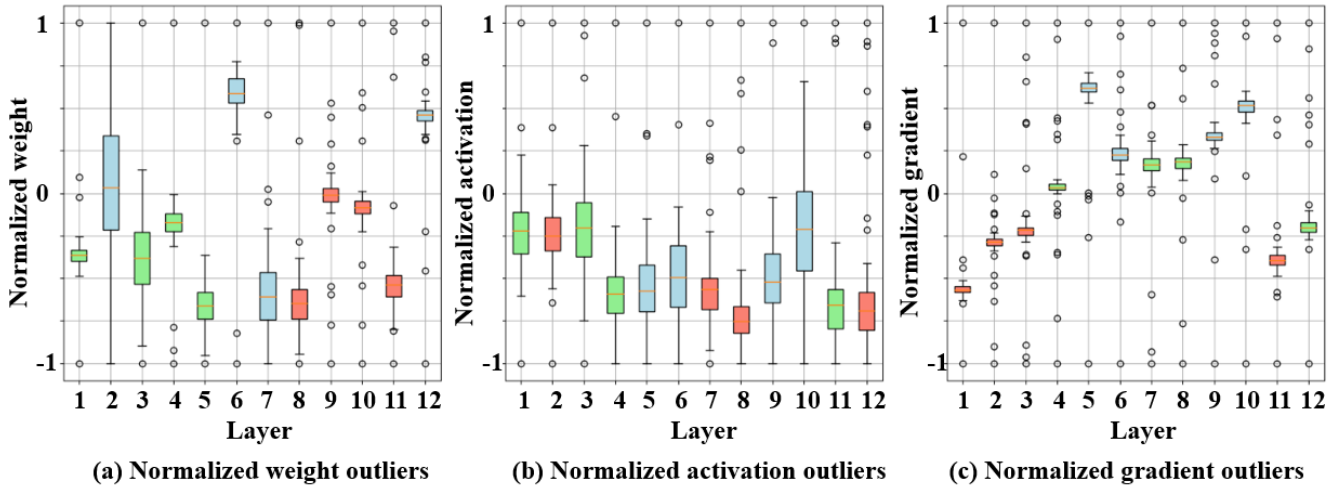


Figure 1: Box plot visualization of outliers of the normalized (a) weight, (b) activation, and (c) gradient in the DeiT-Tiny model.

where θ^* is the optimal parameter that minimizes the loss function and β is the smoothing constant. **Assumption 1** states that the function $L(\theta)$ is convex, and **Assumption 2** states that $L(\theta)$ is smooth and differentiable across all intervals. Previous studies (Duchi, Hazan, and Singer 2011; Yin et al. 2019) assumed the convexity and differentiability of the model. When applying the quantized gradient in the AdamW algorithm, a popular optimizer for ViTs, the following parameter update rules are used:

$$\theta_{t+1} = \theta_t - \eta_t Q(\nabla L_t(\theta_t)), \tag{1}$$

where $\nabla L_t(\theta_t)$ denotes the gradient at time t and η_t represents the learning rate. The gradient is quantized through the function $Q(\cdot)$, which incurs quantization errors that may significantly hinder network convergence. When considering these errors, if the loss function $L(\theta)$ satisfies **Assumptions 1 and 2**, model convergence can be expressed as follows (see the supplementary materials for a detailed proof):

$$\mathbb{E}[\|\theta_t - \theta^*\|^2] \leq \frac{F(\theta_0, \theta^*)}{\sqrt{T}} + \frac{\eta^2(\sigma^2 + \sigma_q^2)}{2\lambda T}. \tag{2}$$

where $\mathbb{E}[\|\theta_t - \theta^*\|^2$ is the error between parameter θ_t and optimal parameter θ^* at iteration time t. Let $F(\theta_0, \theta^*)$ denote the loss between the initial parameter θ_0 and optimal parameter θ^*, where σ^2 and σ_q^2 are the variances of the gradient and quantization error, respectively. Furthermore, let λ be a constant related to the strength of the convex function $L(\theta)$. In Eq.(2), the $\frac{F(\theta_0,\theta^*)}{\sqrt{T}}$ term naturally approaches zero as training progresses with global time(T). However, the $\frac{\eta^2(\sigma^2+\sigma_q^2)}{2\lambda T}$ term becomes difficult to train stably as the quantization error (σ_q^2) and learning rate (η) increase. According to Eq.(2), to ensure learning stability in a network with a quantized gradient, the propagated quantization error must be minimized, and the learning rate must be appropriately controlled.

Proposed Method

Overall Process

We propose a GradQ-ViT framework based on convergence theory to stabilize the training of gradient-quantized ViT models, as shown in Figure 2. We first (1) limit the outlier magnitude by implementing a loss function that combines the outlier-robust Huber loss with the traditional cross-entropy loss. The Huber loss exhibits linearity for large errors and quadratic growth for small errors, which can significantly reduce the size of outliers. We then (2) use IQR to analyze the proportion of gradient distributions and outliers. Based on this analysis, we adaptively assign quantization points to minimize the quantization error incurred by outliers during training. Following quantization, (3) the gradient is restored through dequantization. Owing to the error incurred by quantization, the quantized gradient has a different magnitude and direction than the original gradient, which may interfere with reaching a global minimum in the loss space. To solve this problem, the quantized gradient is scaled similarly to the original gradient following dequantization, ensuring a similar direction and magnitude. Finally, (4) the learning rate of each layer is adaptively assigned by considering the cosine similarity of the gradients and quantization error, thereby ensuring stable training. By effectively minimizing the quantization error caused by outliers and appropriately adjusting the learning rate, the proposed method achieves stable training with high accuracy. Algorithm 1 describes the overall gradient quantization process.

Cross-Huber Blend Loss

The conventional cross-entropy loss function (Zhang and Sabuncu 2018) measures the difference between the predicted and label probability distributions, with higher loss values indicating lower prediction accuracy. This loss function propagates large gradients to significantly update parameters. In other words, higher loss values are associated with larger gradients and outliers. To reduce the magnitude

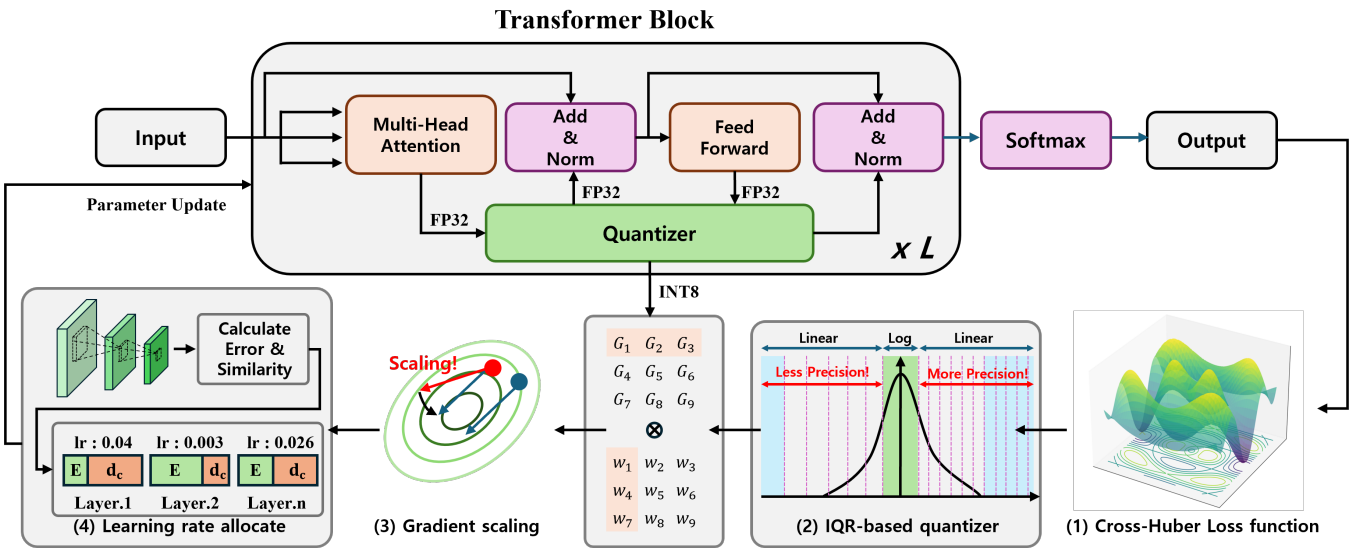


Figure 2: Overview of proposed method. The output computed from forward propagation is passed through the loss function (1) to propagate the gradient. In the quantization process (2), the central 50% range and outliers are analyzed based on IQR to assign quantization precision. In (2), positive ranges with large outlier ratios are assigned more precision. During the dequantization process, the quantized gradient is scaled (3), completing the detailed restoration process, and an adaptive learning rate is assigned to each layer (4).

of outliers in the loss function, the function must be robust to outliers. One loss function that satisfies this requirement is the Huber loss function (Huber 1992). Unlike cross-entropy loss, this function grows linearly as the loss increases, making it less sensitive to outliers. The function can be expressed by the following formula:

$$L_\beta(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \le \beta \\ \beta\left(|a| - \frac{1}{2}\beta\right) & \text{otherwise} \end{cases}. \tag{3}$$

where a is the loss value and β is the hyperparameter. The two loss functions can be combined as

$$L(y, \hat{y}) = \sum_i \left[(1-\delta)\left(-y_i \log(\hat{y}_i)\right) + \delta \cdot L_\beta(y_i - \hat{y}_i)\right]. \tag{4}$$

where δ is a hyperparameter that controls the influence between the cross-entropy and Huber losses. The integrated Cross-Huber blend loss (CH-Loss) can effectively solve classification and outlier problems by simultaneously using the features of the two losses.

IQR-Driven Quantization Strategy

We propose the IQR-based quantization strategy (IQS), which adaptively assigns quantization points based on the characteristics of the gradient distribution considering the IQR, a statistical measure encompassing the middle 50% of a given dataset. In the gradient distribution, most values are densely distributed near zero, indicating small parameter updates. However, a large gradient (*i.e.,* outlier) indicates that the parameter update requires a significant adjustment. During the initial stages of training, large outliers naturally appear due to significant changes in the weights. To reflect all characteristics of these distributions, we perform quantization by adaptively assigning quantization points without clipping:

$$x_{\mathrm{q}} = \begin{cases} \lfloor \left(\frac{x-Q3}{Q3} \times ((2^b - 2^4) \times P_r) \right) \rceil & \text{if } x > Q3 \\ \lfloor \left(\frac{x-Q1}{Q1} \times (2^b - 2^4) \times N_r \right) \rceil & \text{if } x < Q1 \\ \lfloor \left(\frac{\log_2(x-Q1)}{\log_2(Q3-Q1)} \times 2^4 \right) \rceil & \textit{otherwise} \end{cases} \tag{5}$$

where b denotes bit precision, $\lfloor \cdot \rceil$ denotes the rounding operation, and $Q1$ and $Q3$ denote the lower and upper quartiles, respectively. During the adaptive quantization point assignment process, we apply 4-bit logarithmic quantization in the IQR (*i.e.,* Q1 $\le$ x $\le$ Q3). Logarithmic quantization (Chmiel et al. 2021) has 2^n quantization points, making it effective for values that are densely distributed near zero. Because the gradient has many values close to zero, logarithmic quantization can be used to minimize the quantization error stemming from low precision. In other areas, quantization points are assigned based on the positive proportion (P_r) and negative proportion (N_r) of outliers in the distribution. For example, if b is 8-bit, N_r is 60%, and P_r is 40% of the total outliers, the quantization points in the negative region excluding the IQR are assigned $(2^8 - 2^4) \times 0.6 = 144$ quantization points according to Eq.(5). Positive regions are assigned $(2^8 - 2^4) \times 0.4 = 96$ quantization points. In the naive quantization process, uniform quantization points are assigned in the layer, resulting in large errors due to outliers. However, the proposed method minimizes quantization error by assigning more quantization points to regions with large outlier ratios. The proposed quantization strategy effectively

Algorithm 1: Overall gradient quantization process

Require: Full precision gradient g, precision b, loss function parameters δ, β

Ensure: Quantized gradients G_q and updated model parameters

Calculate IQR (Q1, Q3) from gradient distribution
for each layer **do**
  Apply IQS quantization:
  if $Q1 \leq g \leq Q3$ **then**
    $g_q = \lfloor \left(\frac{\log_2(g-Q1)}{\log(Q3-Q1)} \times 2^4 \right) \rceil$
  else if $g < Q3$ **then**
    $g_q = \lfloor \left(\frac{g-Q3-x_{\min}}{x_{\min}} \times (2^{b-4} \times N_r) \right) \rceil$
  else
    $g_q = \lfloor \left(\frac{g-K_u}{x_{\max}-K_u} \times 2^{b-4} \times P_r \right) \rceil$
  end if
  Compute Cross-Huber Blend Loss:
  $L = \sum_i \left[(1-\delta) \left(-y_i \log(\hat{y}_i) \right) + \delta \cdot L_\beta(y_i - \hat{y}_i) \right]$
  Apply Gradient scaling:
  $d_q = \frac{g_q}{\|g_q\|_2}$
  $\hat{g}_q = \left(\frac{\|g\|_2}{\|g_q\|_2} \right) \cdot d_q \cdot \|g\|_2$
  $\cos\_\text{sim} = \frac{g^\top g_q}{\|g\|_2 \|g_q\|_2}$
  $\tilde{g}_q = \hat{g}_q \cdot \cos\_\text{sim}$
  Update learning rate:
  $\eta_{t+1} = \eta_t \cdot (\alpha \cdot Q_e(g, g_q) \cdot \beta \cdot \cos\_\text{sim}) + \sum_i |\theta_i|$
  Update model parameters using $\tilde{g}_q$ and η_{t+1}
end for

reduces the quantization error(σ_q^2) presented in Eq.(2), enabling the model to converge more stably.

Gradient Scaling for Efficient Dequantization

The quantization process changes the gradient magnitude and direction, which can significantly affect model training. Consequently, although dequantization is typically performed by inversely applying the scale factor to restore the original values, gradient quantization requires more precise reconstruction. To achieve this, we determine the similarity between the original and quantized gradients and use the GS algorithm following dequantization to restore the original gradient. First, we normalize the original gradient ($\mathbf{g}$) and quantized gradient ($\mathbf{g_q}$) to obtain their respective direction vectors. The normalized direction vector of the quantized gradient is as follows:

$$\mathbf{d}_q = \frac{\mathbf{g}_q}{\|\mathbf{g}_q\|_2}, \tag{6}$$

Subsequently, the magnitude of $\mathbf{g_q}$ is adjusted to match that of $\mathbf{g}$. This is done by calculating the magnitude ratio between $\mathbf{g_q}$ and $\mathbf{g}$, and multiplying $\mathbf{g}$ by the direction vector $\mathbf{d}_q$, thereby equalizing the vector magnitude:

$$\hat{\mathbf{g}}_q = \left(\frac{\|\mathbf{g}\|_2}{\|\mathbf{g}_q\|_2} \right) \cdot \mathbf{d}_q \cdot \|\mathbf{g}\|_2, \tag{7}$$

To precisely determine the directional similarity between $\mathbf{g}$ and $\mathbf{g_q}$, we then compute the cosine similarity between the two vectors as follows:

$$\cos\_\text{sim} = \frac{\mathbf{g}^\top \mathbf{g}_q}{\|\mathbf{g}\|_2 \|\mathbf{g}_q\|_2}, \tag{8}$$

Finally, we apply cosine similarity to $\mathbf{g_q}$ to adjust $\mathbf{g_q}$ so that it closely matches $\mathbf{g}$:

$$\tilde{\mathbf{g}}_q = \hat{\mathbf{g}}_q \cdot \cos\_\text{sim}. \tag{9}$$

Our gradient quantization solution can effectively reduce the error propagation, as the quantized gradient vector has a similar direction and magnitude to those of the original vector.

Adaptive Learning Rate Allocation

An appropriate learning rate must be assigned to perform training in the quantized network through convergence theory. Specifically, an extremely low learning rate may lead to an insufficient parameter update magnitude, hindering network convergence. We therefore propose the adaptive learning rate allocation (ALA) algorithm that allocates learning rates per layer by measuring the similarity between the original and quantized gradients, along with the quantization error. The ALA algorithm can be expressed as follows:

$$\eta_{t+1} = \eta_t \cdot (\alpha \cdot Q_e(x, x_q) \cdot \beta \cdot Cos_{sim}(x, x_q)) + \sum_i |\theta_i| \,. \tag{10}$$

where $Q_e(x, x_q)$ is the quantization error function, and $Cos_{sim}(x, x_q)$ is the cosine similarity function between the original and quantized gradients. The hyperparameters α and β determine the influence of quantization error and similarity, respectively. The $\sum_i |\theta_i|$ term is an L1-regularization, which is used to enhance model generalizability. Experimentally, we set α and β to 1 and 0, respectively, to stabilize the convergence of the initial epoch. At the beginning of training, the propagating quantization error is large; therefore, according to convergence theory, a small learning rate can be used to stabilize the training process. As training progresses, α and β are adjusted to 0 and 1, respectively, to enhance the influence of gradient similarity and reach a minimum in the loss space. These changes help the network retain an optimal training state by reducing the influence of quantization error and increasing the influence of similarity between gradients. Thus, ALA enables effective control of the learning rate(η) in Eq.(2) and enhances the stability of model training, allowing for optimal parameter updates.

Experimental Results

Experimental Environments

We evaluate the performance of the proposed algorithm on the image classification task using the ImageNet dataset (Russakovsky et al. 2015) with the PyTorch framework in a GPU (RTX-3090) environment. We used three representative ViT models (*i.e.*, DeiT (Touvron et al. 2021), Swin (Liu et al. 2021), and MobileViT (Mehta and Rastegari 2021)) to demonstrate the GradQ-ViT's compatibility with various ViT models. Throughout the experiments, the weight,

activation, and gradient (W/A/G) were quantized to INT8. For the DeiT and Swin models, we used the official code, while the MobileViT setup was configured manually with the AdamW optimizer and cosine scheduler due to the lack of official training code. To evaluate the practical GPU acceleration effect of the proposed quantization framework, we implemented a custom kernel code for convolution and Matmul operations in a CUDA 11.8 environment, making full use of the parallel processing power of the GPU.

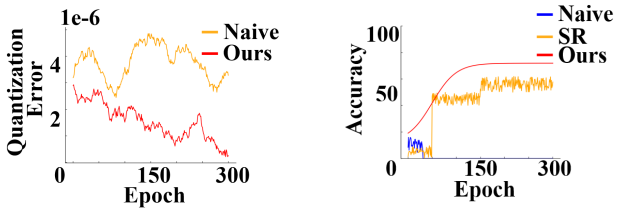
Ablation Studies

In this study, a quantization strategy based on the IQR was combined with a CH-loss function to effectively minimize the quantization error incurred by outliers. Furthermore, we developed a detailed restoration process through GS, with the learning rate adaptively assigned by analyzing the gradient similarity and quantization error. The proposed methods provide a solution to stabilize training in the gradient quantization framework. Figure 3(a) shows the effectiveness of IQS and CH-LOSS in minimizing the quantization error. When SR-based quantization was employed with the naïve method, the quantization error was very large during training. This indicates that by regulating the magnitude of outliers, the proposed method effectively reduces the quantization error and stabilizes training. Figure 3(b) shows results with and without ALA, where Ours is the result of applying all algorithms, and SR is the result of applying only the ALA and SR quantization methods. When using ALA, the accuracy curve was very smooth, indicating enhanced training stability. In contrast, SR exhibited very large oscillations in accuracy, whereas the naive method without ALA converged to zero. These results suggest that the proposed method is highly effective against outliers, further enhancing model stability.

Table 1 presents training results for the DeiT-Tiny model to evaluate the impact of the proposed methods. The severe accuracy drop associated with the ALA method can be attributed to the high quantization error during training using SR, as well as the high error in cosine similarity. We observed that a lower learning rate was assigned than that when using the combined algorithm, which negatively affected training stability. These results demonstrate the importance of appropriate learning rate assignments. However, a large improvement in accuracy was observed when ALA was used in combination with IQS. Because this configuration accounts for the gradient distribution, the highest accuracy recovery was achieved when the proposed algorithms were integrated. Additionally, it suggests that when different algorithms are integrated, they achieve the highest accuracy and complement each other to effectively optimize the quantization process.

Comparison Results on Various Networks

Table 2 presents comparison results with existing quantization methods (*i.e.*, Quantformer (Wang et al. 2022) and Q-ViT (Li et al. 2022)) using the DeiT and Swin models on the ImageNet dataset. Here, F/B denotes the precision used during forward and backward propagation, respectively. The SR method uses a scale factor for the minimum and maximum



(a) Quantization error of each method

(b) Accuracy curve of each method

Figure 3: Quantization error and accuracy measurements as epochs in the DeiT-Tiny model. (a) Quantization error according to the IQS and CH-Loss, (b) accuracy curve according to the ALA algorithm.

ALA	IQS	CH-Loss	GS	Baseline (%)	Accuracy (%)
✓				72.2	65.62
✓	✓				69.48
✓	✓	✓			71.27
✓	✓	✓	✓		72.21

Table 1: Accuracy Results (%) according to Each Proposed Scheme

values of each layer and performs rounding using random numbers generated for each tensor. The proposed quantization method outperformed the baselines in terms of accuracy for both the DeiT and Swin networks, with the exception of the Swin-T model, which exhibited a slight degradation in accuracy despite gradient quantization. These results indicate that the proposed method is a quantization solution that successfully accounts for the characteristics of ViTs (*e.g.,* outliers, complex loss space). It should be noted that the proposed method achieves high accuracy using only deterministic rounding rather than the SR method used in conventional gradient quantization. In contrast, Quantformer (Wang et al. 2022) exhibited a significant accuracy drop despite using a mix of different precision levels and employing a group-wise quantization strategy. To the best of our knowledge, no such gradient-based quantization of ViT models has been studied to date. Thus, the proposed method represents a significant step forward as the first INT8 training quantization framework to be implemented without loss of accuracy.

Table 3 presents evaluation results of the proposed gradient quantization method on the lightweight MobileViT model. For the initial layers, this architecture uses the CNN-based MobileNetV2 model, which is a hybrid transformer network that uses depth-wise convolutional blocks, which are more sensitive to quantization than the ViT model (Kim, Lee, and Kim 2024). To the best of our knowledge, attempts to quantize the gradients of MobileViT have not been made because depth-wise convolution blocks are known to incur significant quantization errors, as mentioned in (Zhu et al. 2020). Therefore, please note that the results of comparative experiments on MobileViT have not been presented. Nevertheless, the proposed method successfully quantized

Model	Method	Baseline (%)	Precision (F/B)	Accuracy (%)	Drop (%)	Model	Method	Baseline (%)	Precision (F/B)	Accuracy (%)	Drop (%)
DeiT-T	SR	72.2	8/8	57.75	14.45	Swin-T	SR	81.2	8/8	65.89	15.31
	Quantformer	72.2	4/32	69.9	2.3		Quantformer	81.2	4/32	78.3	2.9
	Q-ViT	72.86	4/32	72.79	0.07		Q-ViT	80.9	4/32	80.59	0.31
	Ours	72.2	8/8	72.21	**-0.01**		Ours	81.2	8/8	81.15	**0.05**
DeiT-S	SR	79.8	8/8	69.52	10.28	Swin-S	SR	83.2	8/8	69.83	13.37
	Quantformer	79.9	4/32	78.2	1.7		Quantformer	83.2	4/32	81	2.2
	Q-ViT	79.92	4/32	80.11	-0.19						
	Ours	79.8	8/8	80.26	**-0.46**		Ours	83.2	8/8	83.37	**-0.17**
DeiT-B	SR	81.8	8/8	74.38	7.42	Swin-B	SR	84.5	8/8	72.73	11.77
	Quantformer	81.8	4/32	79.7	2.1						
	Ours	81.8	8/8	82.32	**-0.52**		Ours	84.5	8/8	84.71	**-0.21**

Table 2: Comparison of Accuracy Results with Previous Research (%) in the ImageNet Dataset

Model	Method	Baseline(%)	Accuracy(%)	Drop(%)
MobileViT-XXS	SR	69.76	56.25	13.51
	Ours		69.58	**0.18**
MobileViT-XS	SR	75.21	61.74	13.47
	Ours		75.07	**0.14**
MobileViT-S	SR	79.05	66.02	13.03
	Ours		78.96	**0.09**

Table 3: Accuracy Results (%) of MobileViT using the ImageNet Dataset

Precision (F/B)	Forward (ms)	Backward (ms)	Iteration (ms)
32/32	84.37	180.42	281.45
8/32	41.19	180.42	240.73
4/32	**27.83**	180.42	206.72
8/8	41.19	**79.38**	**136.58**

Table 4: Running time of MobileViT on the GeForce RTX 3090 system

W/A/G to INT8 with a negligible accuracy drop of less than 0.2% from the baseline, independent of the size of the MobileViT model (*i.e.,* XXS, XS, S). This suggests that the quantization strategy of the proposed method, which accounts for the gradient distribution, has significant strength in complex depth-wise convolution blocks. Consequently, the proposed method achieves excellent accuracy in high-dimensional networks such as DeiT and Swin, as well as lightweight networks such as MobileViT.

Acceleration Results on NVIDIA GPUs

We developed a custom kernel code to verify the practical efficiency of INT8 operations on a GPU for the MobileViT model. This code was implemented in the CUDA 11.8 environment for convolution and Matmul operations. Although the conventional method (Zhu et al. 2020) relies on global memory, we instead used shared memory to reduce DRAM access. This was done to minimize the latency caused by memory access and verify the effectiveness of quantization operations. Shared memory facilitates data sharing between groups of threads in the CUDA environment, thereby enabling high-throughput data processing. This allows each thread to access the data faster, significantly reducing the overall computation time. The kernel implementation was designed by considering complex data access patterns and memory usage. Each operation was processed in INT8 units to reduce computational costs and increase processing speed. Table 4 lists the end-to-end times for the lightweight ViT model across different precisions for forward operations, backward operations, and overall iteration. When both F/B were quantized to INT8, respective speedups of 2.04× and 2.27× were observed compared to FP32. Additionally, the total iteration time measurement indicated a 2.06× speedup compared to FP32. When comparing forward and backward operations to INT4 and FP32 results (*i.e.*, 4/32), respectively, the forward operation was 1.48× slower, while the backward operation and overall iteration exhibited speedups of 2.27× and 1.51×, respectively. These results indicate that the backward operation accounts for a significant portion of the computation in the entire training process, suggesting that gradient quantization is essential to reducing training costs. Our experiments demonstrate that the gradient access approach utilizing custom kernels and shared memory can significantly accelerate MobileViT computations on the GPU.

Conclusion

In this study, we proposed solutions to reduce outliers in gradient quantization for ViT networks based on convergence theory. We combined an IQR-based quantization strategy with the CH-Loss function to minimize projection errors from outliers. The GS method optimized propagated quantization error by maximizing the similarity between original and quantized gradients, while the ALA algorithm, considering gradient similarity and quantization error, stabilized training. Our approach achieved high accuracy across ViT networks, especially in lightweight models like MobileViT. Unlike prior studies, we also verified GPU-level acceleration via custom kernel code and developed the first INT8 training framework for ViT models. We hope that our work will accelerate future research on gradient quantization.

Acknowledgments

This work was partly supported by K-CHIPS(Korea Collaborative & High-tech Initiative for Prospective Semiconductor Research)(2410000620, RS-2024-00405946, 24052-15TC) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01305, Development of AI Deep-Learning Processor and Module for 2,000 TFLOPS Server).

References

Cai, H.; Li, J.; Hu, M.; Gan, C.; and Han, S. 2022. Efficientvit: Multi-scale linear attention for high-resolution dense prediction. *arXiv preprint arXiv:2205.14756*.

Chen, F.; Luo, Z.; Zhou, L.; Pan, X.; and Jiang, Y. 2024. Comprehensive survey of model compression and speed up for vision transformers. *arXiv preprint arXiv:2404.10407*.

Chmiel, B.; Banner, R.; Hoffer, E.; Yaacov, H. B.; and Soudry, D. 2021. Logarithmic unbiased quantization: Simple 4-bit training in deep learning. *arXiv preprint arXiv:2112.10769*.

Chun, D.; Lee, H.-J.; and Kim, H. 2024. PF-Training: Parameter Freezing for Efficient On-Device Training of CNN-based Object Detectors in Low-Resource Environments. In *2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, 21–25.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Frumkin, N.; Gope, D.; and Marculescu, D. 2023. Jumping through local minima: Quantization in the loss landscape of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16978–16988.

Gupta, S.; Agrawal, A.; Gopalakrishnan, K.; and Narayanan, P. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, 1737–1746. PMLR.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conf. on Comp. Vis. and pattern recognition*, 770–778.

Huber, P. J. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, 492–518. Springer.

Kang, B. J.; Choi, D. H.; and Kim, H. 2024. Mixed Precision Quantization with Hardware-Friendly Activation Functions for Hybrid ViT Models. In *2024 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–2. IEEE.

Kim, N. J.; and Kim, H. 2022. Fp-agl: Filter pruning with adaptive gradient learning for accelerating deep convolutional neural networks. *IEEE Transactions on Multimedia*, 25: 5279–5290.

Kim, N. J.; Lee, J.; and Kim, H. 2024. HyQ: Hardware-Friendly Post-Training Quantization for CNN-Transformer Hybrid Networks. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, 4291–4299.

Lee, G.; Lee, S.; and Kim, H. 2024. ACC: Adaptive Compression Framework for Efficient On-device CNN Training. In *2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, 472–476.

Lee, J.; Kim, D.; and Ham, B. 2021. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6448–6457.

Lee, J.; and Kim, H. 2024. DCT-ViT: High-Frequency Pruned Vision Transformer with Discrete Cosine Transform. *IEEE Access*.

Lee, S. I.; and Kim, H. 2022. Gaussianmask: Uncertainty-aware instance segmentation based on gaussian modeling. In *2022 26th International Conference on Pattern Recognition (ICPR)*, 3851–3857. IEEE.

Lee, S. I.; Koo, K.; Lee, J. H.; Lee, G.; Jeong, S.; O, S.; and Kim, H. 2024. Vision transformer models for mobile/edge devices: a survey. *Multimedia Systems*, 30(2): 109.

Li, Y.; Dong, X.; and Wang, W. 2019. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *arXiv preprint arXiv:1909.13144*.

Li, Z.; Yang, T.; Wang, P.; and Cheng, J. 2022. Q-vit: Fully differentiable quantization for vision transformer. *arXiv preprint arXiv:2201.07703*.

Lin, Y.; Zhang, T.; Sun, P.; Li, Z.; and Zhou, S. 2021. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Mehta, S.; and Rastegari, M. 2021. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*.

Ovi, P. R.; Dey, E.; Roy, N.; and Gangopadhyay, A. 2023. Mixed quantization enabled federated learning to tackle gradient inversion attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5046–5054.

Peng, P.; You, M.; Jiang, K.; Lian, Y.; and Xu, W. 2023. Mbfquant: a multiplier-bitwidth-fixed, mixed-precision quantization method for mobile cnn-based applications. *IEEE Transactions on Image Processing*, 32: 2438–2453.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR.

Wang, Z.; Wang, C.; Xu, X.; Zhou, J.; and Lu, J. 2022. Quantformer: Learning extremely low-precision vision transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 8813–8826.

Xu, K.; Wang, Z.; Chen, C.; Geng, X.; Lin, J.; Yang, X.; Wu, M.; Li, X.; and Lin, W. 2024. LPViT: Low-Power Semi-structured Pruning for Vision Transformers. *arXiv preprint arXiv:2407.02068*.

Yang, H.; Yin, H.; Shen, M.; Molchanov, P.; Li, H.; and Kautz, J. 2023. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18547–18557.

Yao, Z.; Dong, Z.; Zheng, Z.; Gholami, A.; Yu, J.; Tan, E.; Wang, L.; Huang, Q.; Wang, Y.; Mahoney, M.; et al. 2021. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, 11875–11886. PMLR.

Yin, P.; Zhang, S.; Lyu, J.; Osher, S.; Qi, Y.; and Xin, J. 2019. Blended coarse gradient descent for full quantization of deep neural networks. *Research in the Mathematical Sciences*, 6: 1–23.

You, H.; Sun, Z.; Shi, H.; Yu, Z.; Zhao, Y.; Zhang, Y.; Li, C.; Li, B.; and Lin, Y. 2023. Vitcod: Vision transformer acceleration via dedicated algorithm and accelerator co-design. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 273–286. IEEE.

Yuan, Z.; Xue, C.; Chen, Y.; Wu, Q.; and Sun, G. 2022. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, 191–207. Springer.

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

Zhao, K.; Huang, S.; Pan, P.; Li, Y.; Zhang, Y.; Gu, Z.; and Xu, Y. 2021. Distribution adaptive int8 quantization for training cnns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3483–3491.

Zhu, F.; Gong, R.; Yu, F.; Liu, X.; Wang, Y.; Li, Z.; Yang, X.; and Yan, J. 2020. Towards unified int8 training for convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1969–1979.

Zhu, S.; Voigt, T.; Ko, J.; and Rahimian, F. 2022. On-device training: A first overview on existing systems. *arXiv preprint arXiv:2212.00824*.