

# Mixed Non-linear Quantization for Vision Transformers

Gihwan Kim<sup>1†</sup>, Jemin Lee<sup>2†</sup>, Sihyeong Park<sup>3</sup>, Yongin Kwon<sup>2§</sup>, and  
Hyungshin Kim<sup>1§</sup>

<sup>1</sup> Chungnam National University

`gihwan.kim98@o.cnu.ac.kr` and `hyungshin@cnu.ac.kr`

<sup>2</sup> Electronics and Telecommunications Research Institute

`{leejaymin,yongin.kwon}@etri.re.kr`

<sup>3</sup> Korea Electronics Technology Institute

`sihyeong@keti.re.kr`

**Abstract.** The majority of quantization methods have been proposed to reduce the model size of Vision Transformers, yet most of them have overlooked the quantization of non-linear operations. Only a few works have addressed quantization for non-linear operations, but they applied a single quantization method across all non-linear operations. We believe that this can be further improved by employing a different quantization method for each non-linear operation. Therefore, to assign the most error-minimizing quantization method from the known methods to each non-linear layer, we propose a mixed non-linear quantization that considers layer-wise quantization sensitivity measured by SQNR difference metric. The results show that our method outperforms I-BERT, FQ-ViT, and I-ViT in both 8-bit and 6-bit settings for ViT, DeiT, and Swin models by an average of 0.6%p and 19.6%p, respectively. Our method outperforms I-BERT and I-ViT by 0.6%p and 20.8%p, respectively, when training time is limited. We plan to release our code<sup>4</sup>.

**Keywords:** Quantization · Vision Transformer · Non-linear Quantization

## 1 Introduction

Vision Transformers have replaced traditional Convolutional Neural Networks (CNNs) in various vision tasks due to their high accuracy. Alongside this, research on quantization to reduce model size for deployment on diverse devices, including resource-constrained devices, are actively ongoing. Quantization methods studied for ViTs are divided into quantization-aware training (QAT) [11, 15–17, 21, 28] which involves learning, and post-training quantization (PTQ) [3, 19, 23, 29, 31, 32] which does not require learning. However, these studies have applied

<sup>†</sup> These authors contributed equally to this work and are listed alphabetically.

<sup>§</sup> Corresponding authors

<sup>4</sup> <https://gitlab.com/ones-ai/mixed-non-linear-quantization>

quantization only to the linear operations, neglecting many non-linear operations such as Softmax, GELU, and layer normalization (LayerNorm). Consequently, during inference, some or all operations utilize dequantized floating-point parameters, failing to take full advantage of efficient low-precision arithmetic units and thus leading to insufficient acceleration of the model.

To address these issues, studies on quantizing the non-linear operations in ViT models have been recently proposed [4, 13, 18–20, 20, 26, 32]. However, these methods use polynomial [4, 13, 20], logarithm [19, 20, 32], and bit-shifting [18] techniques to transform all non-linear operations into linear operations and perform the same quantization for all non-linear operations in ViTs. However, the output activation distribution of non-linear operations varies widely depending on the position of the operators. Consequently, uniformly applying a single non-linear quantization method to the entire non-linear operations is suboptimal in terms of minimizing quantization errors.

In this paper, we propose a mixed non-linear quantization approach that considers layer-wise quantization sensitivity within vision transformers. We utilize multiple quantization methods and we assign the best quantization method for each non-linear operation. Our method selects a quantization approach with the least quantization error according to the characteristics of the non-linear operation and the position of each non-linear layer. To measure the quantization error on a layer-wise basis, we devised a new metric called *SQNR diff*. This metric represents the difference in SQNR values between input and output activations. This allows the assignment of non-linear quantization methods based on improvements in SQNR.

The considered non-linear quantization methods include three methods proposed in I-BERT [13], FQ-ViT [20], and I-ViT [18]: bit-shifting [18], logarithm [20], and polynomial [13]. Quantizing vision transformers with our mixed non-linear quantization approach reduces quantization error in non-linear layers on a layer-wise basis, allowing for more precise quantization than previous works. When performing layer-wise non-linear operation analysis, mixed non-linear quantization enhanced quantization sensitivity compared to existing quantization methods. Furthermore, when measuring top-1 accuracy for image classification on various vision transformer models, the accuracy was higher than that of existing integer-only vision transformer models. The proposed method improved accuracy by an average of 0.6%p and 19.6%p in 8-bit and 6-bit environments for ViT, DeiT, and Swin models compared to the conventional studies I-BERT, FQ-ViT, I-ViT. Notably, even when quantization was performed with only 1-epoch QAT, it resulted in a mere -1.33%p accuracy difference compared to full training, a significantly different result from the -22.27%p average accuracy drop caused by I-ViT in full training comparisons.

The main contributions of this paper are as follows:

- We discover that applying a single quantization method to non-linear operations in ViT models does not minimize quantization errors effectively. This observation is based on layer-wise *SQNR diff* analysis, which demon-

strate that a single quantization approach often fails to address the diverse activation distribution of different non-linear layers.

- We propose a mixed non-linear quantization method that combines three existing quantization methods—bit-shifting, logarithm, and polynomial quantization—to reduce quantization error in each non-linear operation.
- Our method achieves higher accuracy compared to previous works including I-BERT, FQ-ViT, and I-ViT. Furthermore, even with only one epoch of retraining, it attains higher accuracy than prior studies.

## 2 Related Work

### 2.1 Vision Transformer Quantization

Recent studies have actively pursued the quantization of Vision Transformer (ViT) models to compress them [8, 9]. Research on quantizing ViT models can be divided into the PTQ and the QAT methods. In the PTQ, quantization have been proposed to address the increase in quantization error due to the wide distribution of activations [3, 19, 23, 29, 31, 32]. QAT methods [11, 15–17, 21, 28] enable quantization below 4-bit thanks to retraining, unlike PTQ. However, these ViT works have applied quantization only to linear operations existing in ViTs, and many non-linear operations like Softmax have not been quantized. Therefore, throughout the inference process, specific operations are carried out using floating-point parameters that have been converted to a continuous scale, so not fully using the capabilities of high-performance low-precision arithmetic units and resulting in inefficient acceleration of the model.

To address the aforementioned issues, recent studies have proposed integer-only vision transformer models that also quantize all non-linear operations, as listed in Table 1 [4, 18, 20, 26, 30]. I-ViT [18] performed quantization by approximating the operations of Softmax, LayerNorm, and GELU to integer operations using bit-shifting. FQ-ViT [20] used log2 quantization and *i-exp* [13] to quantize LayerNorm and Softmax. PackQViT [4] quantized non-linear operations using polynomial approximation. However, existing non-linear quantization methods have only quantized some non-linear operations [20] or work exclusively on custom-designed hardware, thus lacking versatility [26]. Furthermore, fundamentally, these methods are not optimal in minimizing quantization errors because they uniformly apply a non-linear quantization approach to all non-linear operations within the model, failing to quantize according to the distribution of activations [4, 13, 18, 20, 30].

### 2.2 Mixed Quantization

Mixed-precision quantization is a quantization technique that assigns different bit-widths to each layer or block of a Neural Network considering the trade-off between accuracy and efficiency. By variably assigning bit-widths per layer, it can reduce accuracy degradation compared to fixed-precision quantization [9].

**Table 1:** Previous works for non-linear quantization in Vision Transformers.

Model	Softmax	LayerNorm	GELU	Code	QAT/PTQ	Approximation
I-ViT [18]	✓	✓	✓	✓	QAT	Bit-shifting
I-BERT [13]	✓	✓	✓	✓	QAT	Polynomial
PackQViT [4]	✓	✓	✓		QAT	Polynomial
FQ-ViT [20]	✓		✓	✓	PTQ	Logarithm
EdgeKernel [30]	✓	✓	✓		PTQ	Logarithm
SOLE [26]	✓	✓	✓		PTQ	Logarithm

The method of HAQ [25] uses Deep Reinforcement Learning to decide bit-widths. To determine bit-widths, HAWQ [6] and HAWQ-V2 [5] respectively use the top eigenvalue and average hessian trace to measure layer-wise sensitivity to quantization. Furthermore, HAWQ-V3 [27] has implemented models with mixed bit-widths for integer-only quantization. Recently, applying mixed bit-width quantization to vision transformers [23] optimized the bit-width allocation per layer by considering the similarity between full-precision tensor and quantized tensor. However, these methods all focus on applying mixed bit-widths solely to linear operations and do not consider non-linear operators. Our work focuses on selecting and applying mixed methods optimally for quantizing non-linear operations, unlike existing mixed quantization studies that only consider bit-width. The proposed method is independent of the existing methods of applying mixed bit-widths and consequently has the potential to generate synergy when used simultaneously.

### 3 Method

In this section, we describe the background of the quantization, and the mixed non-linear quantization. Additionally, we explain how the layer-wise non-linear operation analysis is conducted for mixed non-linear quantization, demonstrating that the quantization error varies depending on the quantization method applied to each non-linear operation.

#### 3.1 Background

In this section, we explain basic concept of the quantization used in this paper.

**Basic Quantization Method and Quantization Aware Training.** The linear operations of the vision transformer exhibit homogeneity, thus they can be quantized through linear operation quantization [5, 12] for the embedding layer (Conv), MatMul, and Dense layers. For ease of implementation, input and weight are quantized using symmetric uniform manner as described in Eq. (1). In Eq. (1),  $R$  represents the full-precision value, and  $\alpha$  denotes the clipping range,

which is determined by the larger of the absolute values of  $r_{min}$  and  $r_{max}$ .  $n$  represents the bit-precision.

$$I = \left\lfloor \frac{\text{clip}(R, -\alpha, \alpha)}{S} \right\rfloor, \quad \text{where; } S = \frac{2\alpha}{2^n - 1}, \alpha = \max(|r_{max}|, |r_{min}|) \quad (1)$$

To address the issue where gradients cannot back-propagate in Eq. (1), the Straight-Through Estimator (STE) [1] is used to approximate the gradient during quantization-aware training. The gradient of the rounding operation is approximated as 1 within the quantization limit. In back-propagation using STE, the gradient of the loss  $\mathcal{L}$  with respect to the real-valued data  $R$  is given by Eq. (2)

$$\frac{\partial \mathcal{L}}{\partial R} = \frac{\partial \mathcal{L}}{\partial I} \cdot \mathbf{1}_{-\alpha \leq \frac{R}{S} \leq \alpha}, \quad (2)$$

where  $\mathbf{1}$  functions as an indicator, yielding 1 within the boundaries of quantization and 0 outside these boundaries.

**Non-linear Quantization Methods.** A vision transformer contains three non-linear operations: layer normalization (LayerNorm), Softmax, and GELU. Studies have proposed approximating each of these non-linear operations with specific integer-only functions, as shown in Table 1. In this study, we utilize three techniques that are open-source and easily accessible: I-BERT [13], FQ-ViT [20], and I-ViT [18], applying a mixed approach to each non-linear operation. The quantization methods for non-linear operations are detailed as follows.

LayerNorm includes division, square, and square root as shown in Eq. (3). The computation of the mean and standard deviation is dynamically calculated based on the input values, which presents challenges.

$$\text{LayerNorm}(X) = \frac{X - \mu_X}{\sqrt{\sigma_X^2 + \epsilon}} \cdot \gamma + \beta \quad (3)$$

Non-linear quantizations of I-BERT and I-ViT compute exact value of  $\sqrt{n}$  using an iterative algorithm based on Newton’s Method [2]. In this process, Newton’s Method requires only integer arithmetic operations. Quantization of I-ViT differs from I-BERT in that it represents the division by 2 of the sum of the  $i$ -th iteration value and the rounded value with a bit-shift, and the number of iterations is determined empirically. In FQ-ViT, LayerNorm is approximated to integers with the output scaling factor obtained from PTQ along with the log2 and the sign function.

In Softmax, as shown in Eq. (4), the exponential function introduces non-linearity, making it crucial to approximate. I-ViT applies the base change formula to transform the base of the exponential function from  $e$  to 2, enabling shift operations. As shown in Eq. (5), where the  $\log_2 e$  appearing in the exponent is approximated as  $(1.0111)_b$ , allowing the binary value to be represented by shift operations and addition. I-BERT approximates the exponential function with

a second-order polynomial. To determine the coefficients of the polynomial, i-exp [13] is proposed, which minimizes the  $L^2$  distance in the exponential function. FQ-ViT proposes Log-Int-Softmax, which adds log2 quantization to the i-exp of I-BERT.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j^d e^{x_j}} \quad (4)$$

$$e^x = 2^{x \cdot \log_2 e} \approx 2^{x \cdot (1 + (1 \gg 1) - (1 \gg 4))} \quad (5)$$

The GELU function poses quantization issues with the erf function as in Eq. (6). I-BERT proposed i-GELU by approximating the erf function with a second order polynomial function. I-ViT employs an approximation based on the sigmoid proposed by GELUs [10], as in Eq. (7).

$$\text{GELU}(x) = x \cdot \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x}{\sqrt{2}} \right) \right] \quad (6)$$

$$\text{GELU}(x) \approx x \cdot \sigma(1.702x) \quad (7)$$

The value 1.702 in Eq. (7) is approximated as the binary  $(1.1011)_b$  and represented with the shift operation as shown in Eq. (8). The sigmoid function uses ShiftExp, which was employed for quantizing LayerNorm, to quantize the exponential function, resulting in the proposed ShiftGELU.

$$x\sigma(1.702x) = x\sigma(x + (x \gg 1) + (x \gg 3) + (x \gg 4)) \quad (8)$$

### 3.2 Mixed Non-linear Quantization

In this paper, we propose a method to select the optimal non-linear quantization method for each non-linear layer. This is implemented exploiting the distribution of output activation and input tensor. We calculate the quantization error for each layer to choose the quantization method for non-linear operations. We measure layer-wise quantization sensitivity of the three non-linear quantization methods, I-BERT, FQ-ViT, and I-ViT, and select the quantization method that induces the minimum error. For GELU, since FQ-ViT does not provide a quantization method, there are only two available options. For Softmax and LayerNorm, there are three available methods (I-BERT, FQ-ViT, and I-ViT), and the final search space of the non-linear quantization becomes Eq. (9).

$$3^{(\# \text{ of Softmax})} \times 3^{(\# \text{ of LayerNorm})} \times 2^{(\# \text{ of GELU})} \quad (9)$$

In the case of the ViT, which contains 12 Softmax, 12 GELU, and 25 LayerNorm, the size of the search space becomes approximately  $9.47 \times 10^{18}$ . Calculating all dependencies among non-linear operations is intractable. Therefore, like previous studies, this study also assumes that all layers are independent [5, 6, 27]. In this case, only  $3 \times 12 + 3 \times 12 + 2 \times 25 = 122$  computations are required.

The proposed mixed non-linear quantization progresses through the following three stages: i) Quantized model setting, ii) Calculate layer-wise quantization sensitivity, and iii) Selecton of the optimal non-linear quantization method.

**Quantized Model Setting.** To calculate layer-wise quantization sensitivity for mixed non-linear quantization, models quantized with three non-linear quantization methods [13, 18, 20] are prepared. The linear quantization part follows the Dyadic Quantization of HAWQv3 [27], and the base code from the official I-ViT [18] was used. Only the non-linear quantization part of the base code was implemented differently using the three methods to configure the quantized models.

**Calculate Layer-wise Quantization Sensitivity.** We perform inference on both full-precision vision transformer and quantized vision transformers, and generate a list of quantization sensitivity of each layer. The Signal-to-Quantization-Noise Ratio (SQNR) is used to measure the quantization sensitivity. SQNR represents the ratio between the original data and the quantization noise. We measure the quantization error induced from rounding during the quantization process by comparing the original tensor with the tensor after quantization and dequantization. A higher SQNR value indicates that the quantized non-linear layer produces a tensor similar to that of the full-precision non-linear layer, implying excellent integer approximation accuracy of the layer. In Eq. (10),  $x$  represents the original tensor, and  $q$  represents the tensor after quantization and dequantization. Therefore, quantization error is calculated through  $x - q$ . For the final SQNR calculation, the original tensor is divided by the quantization error and the ratio is converted into a *logscale*. The calculation of SQNR is ultimately done for a batch size  $N$ , and thus it is computed as the arithmetic mean accumulated over the batch. This average SQNR (ASQNR) is defined as Eq. (10). This process is carried out for all layers in the ViT model. Finally, layer-wise quantization sensitivity for the model quantized with the three considered quantization methods is generated.

$$\text{ASQNR} = 20 \log \left( \frac{1}{N} \sum_{i=1}^N \frac{\mathbb{E}[(x_i)^2]}{\mathbb{E}[(x_i - q_i)^2]} \right) \quad (10)$$

**Selection of the optimal non-linear quantization method.** In the previous stage, three list of layer-wise quantization sensitivities were generated, allowing for the selection of the optimal non-linear quantization for each non-linear layer. This paper does not solely choose the quantization method based on the output tensor’s ASQNR, but utilizes the newly devised *SQNR diff* metric. *SQNR diff* is the difference between the quantization sensitivities of the current non-linear operation’s input and output tensors, thus considering the quantization sensitivity of both input and output. As a criterion for selecting the non-linear quantization method, the method with the lowest *SQNR diff* for each non-linear layer is chosen for the quantization of that non-linear layer. Experimental comparisons between the *SQNR diff*-based selection proposed in this paper and the method of choosing quantization based on high output quantization sensitivity (*SQNR output*) showed that *SQNR diff* is more effective in improving accuracy.

The results of accuracy comparison between the two assignment methods are summarized in Table 2.

$$\begin{aligned} \text{SQNR diff} &= \beta - \gamma \\ \text{where; } \beta &= \text{ASQNR}(X_{out}, Q_{out}), \gamma = \text{ASQNR}(X_{in}, Q_{in}) \end{aligned} \quad (11)$$

*SQNR diff* is calculated as shown in Eq. (11). In Eq. (11),  $X_{out}$  and  $X_{in}$  refer to the output and input tensors of the non-linear layer in the full-precision model, respectively, and  $Q_{out}$  and  $Q_{in}$  denote the output and input tensors that represent the tensor after quantization and dequantization. The *SQNR diff* can be calculated by subtracting the input quantization sensitivity  $\gamma$  from the output quantization sensitivity  $\beta$  generated after passing through the quantized non-linear layer.

**Table 2:** Accuracy comparison of two decision rules for the 8-bit mixed non-linear quantization in vision transformers. The *SQNR diff* rule achieved higher accuracy than SQNR output decision rule. All models were trained in 5 epochs and the same batch size for each model.

Decision rule	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S
SQNR output	63.01	34.27	70.45	79.74	81.19	5.81	79.49
SQNR diff	79.28	83.25	71.88	79.78	81.65	79.32	80.67

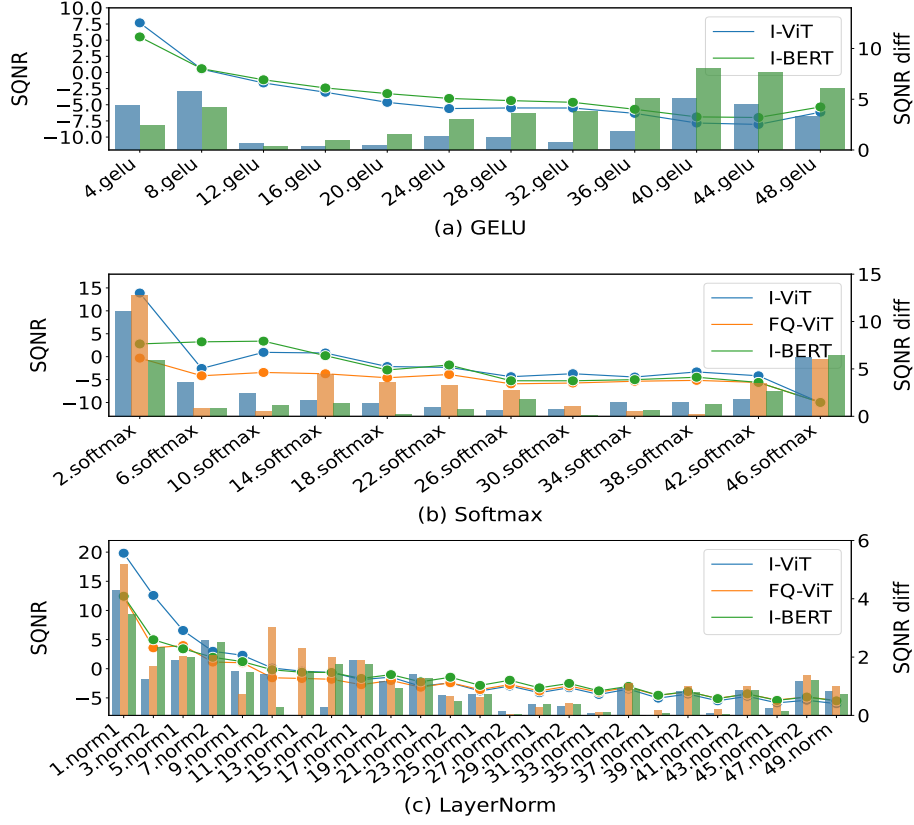
## 4 Experiments

In this section, we describe experiment environment, results of layer-wise non-linear operation analysis and performance evaluation results. SQNR analysis related to the position of non-linear operations and the resulting non-linear quantization mappings is performed. Effectiveness of the proposed mixed non-linear quantization is shown by implementing onto the well known vision transformers. Various non-linear quantizations are compared in 8-bit and 6-bit environment. Efficiency from a training time perspective is discussed.

### 4.1 Experiment Setup

ViT [7], DeiT [24], and Swin [22] are selected to implement the mixed non-linear quantization with the ImageNet dataset [14]. All these models utilized pre-trained weights downloaded from the timm<sup>5</sup> library, and Quantization Aware Training (QAT) was conducted for quantization. Uniform symmetric quantization was applied to quantize all weights and activations of the full-precision pre-trained vision transformer models, and basic QAT based on STE [1] was

<sup>5</sup> <https://github.com/huggingface/pytorch-image-models>



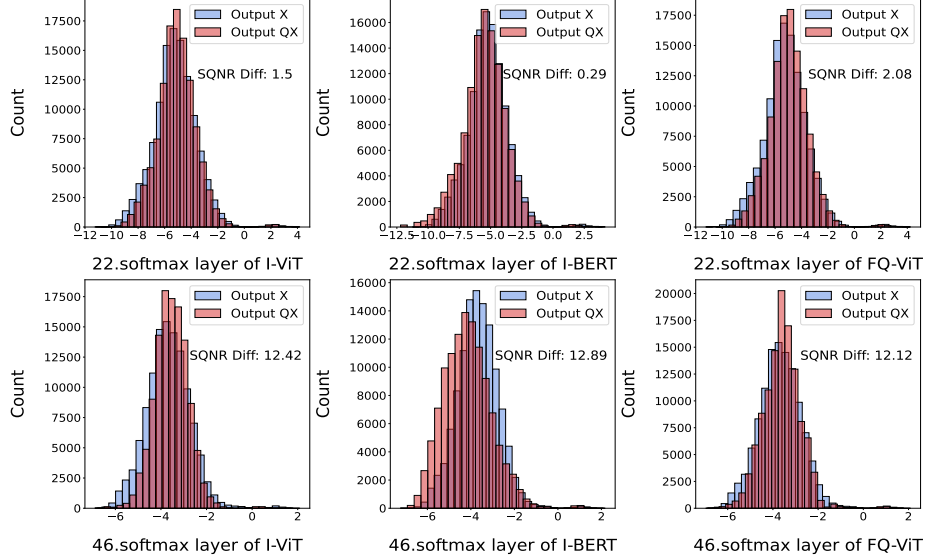
**Fig. 1:** Layer-wise quantization sensitivity of non-linear operations for quantized DeiT-T with the existing non-linear quantization methods. Bar plots denotes  $SQNR\ diff$  of each non-linear layer, while the line plots denote  $SQNR$  output. Bar plots indicate no single quantization method uniformly minimizes  $SQNR\ diff$  across all non-linear layers.

performed. The optimizer used was AdamW. Four A6000 and four A100 GPUs were utilized for sensitivity analysis and QAT. The GPU training time for QAT, based on the A100-80G, took an average of 7 days for the Tiny, Small, and Base models.

## 4.2 Layer-Wise Non-linear Operation Analysis

In this section, we experimentally demonstrate that both  $SQNR$  and  $SQNR\ diff$  vary depending on the layer position of the non-linear operation in the quantization process. Additionally, we present the mapping results of the proposed  $SQNR\ diff$ -based mixed non-linear quantization for each model.

Figure 1 shows the  $SQNR$  and  $SQNR\ diff$  values for the non-linear operations of the DeiT-T model, quantized by different non-linear quantization methods (I-



**Fig. 2:** Histogram comparison of output tensor for specific non-linear layers in full-precision and quantized DeiT-T models. There is no effective non-linear quantization method which approximate full-precision output tensor  $X$  for all non-linear layers. Each subplot shows the distribution of full-precision (*Output X*) and quantized (*Output QX* after quantization and dequantization) tensor, along with their *SQNR diff*. The layer names are in accordance with Figure 1.

ViT, FQ-ViT and I-BERT). The quantization sensitivity of the output tensor for GELU, Softmax, and LayerNorm is represented in a line plot, while the measurements taken using the proposed *SQNR diff* metric are displayed in a bar plot. Despite applying the same non-linear quantization to the same operations in the non-linear layers, different quantization sensitivities can be observed in Figure 1. For example, in the line plot for Softmax, the *2.softmax* layer showed superior performance with I-ViT, while the *6.softmax*, *10.softmax* and *22.softmax* layers exhibited higher quantization sensitivity values with I-BERT, and the *46.softmax* layer did so with FQ-ViT. The individual *SQNR diff* for each non-linear layer, the bar plot in Figure 1 confirmed that the *SQNR diff* values varied among all layers. Therefore, no single quantization method demonstrated superior performance across all non-linear layers.

In the Figure 1, the effectiveness of the quantization performance varied across different positions of the Softmax layers. The original output tensor and the quantized output tensor are analyzed in a histogram as shown in Figure 2. In the histogram, the blue area labeled *Output X* represents the output tensor of the layer from the full-precision model, and the red area labeled *Output QX* represents the output tensor that has passed through the layer with non-linear quantization. For the *22.Softmax*, the I-BERT has the smallest *SQNR diff*=0.29

**Table 3:** This table shows which and how much of non-linear quantization was selected for each non-linear layer with the mixed non-linear quantization.

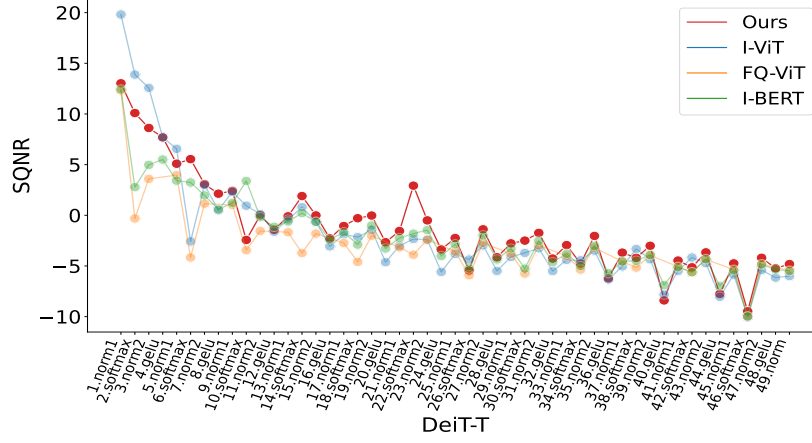
Model	Softmax			GELU		LayerNorm		
	I-BERT	FQ-ViT	I-ViT	I-BERT	I-ViT	I-BERT	FQ-ViT	I-ViT
ViT-S	1	7	4	12	0	2	0	23
ViT-B	1	8	3	12	0	0	1	24
DeiT-T	6	4	2	3	9	13	4	8
DeiT-S	8	1	3	11	1	23	1	1
DeiT-B	11	0	1	11	1	2	0	23
Swin-T	5	0	7	11	1	18	4	3
Swin-S	6	15	3	18	6	11	27	11

**Table 4:** Top-1 accuracy (%) of quantized 8-bit and 6-bit vision transformer models. \* denotes the results that are re-produced with the official code.

Method	Bits	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S
Full-Precision	32-bit	81.39	84.53	72.21	79.85	81.85	81.35	83.2
FQ-ViT [20]	8-bit	—	83.31	71.61	79.17	81.20	80.51	82.71
I-BERT [13]	8-bit	80.47	83.70	71.33	79.11	80.79	80.15	81.86
I-ViT [18]	8-bit	81.27	84.76	72.24	80.12	81.74	81.50	83.01
<b>Ours</b>	8-bit	<b>81.35</b>	<b>84.88</b>	<b>72.55</b>	<b>79.93</b>	<b>81.88</b>	<b>80.95</b>	<b>82.42</b>
FQ-ViT [20]	6-bit	4.26	0.10	58.66	45.51	64.63	66.50	52.09
I-ViT* [18]	6-bit	70.24	76.89	63.80	74.48	76.0	71.89	81.04
<b>Ours</b>	6-bit	<b>73.64</b>	<b>82.00</b>	<b>67.81</b>	<b>77.19</b>	<b>79.91</b>	<b>80.54</b>	<b>79.10</b>

and for the *46.Softmax*, the FQ-ViT has the smallest  $SQNR\ Diff=12.12$  so that they are the non-linear quantization methods that achieve the highest preservation of the original tensors. Since the optimal non-linear method varies depending on the activation distributions, applying a mixed approach to non-linear quantization can be effective in reducing quantization errors.

From the quantization sensitivity analysis for non-linear operations, non-linear quantization techniques can be mixed and applied to each non-linear layer. As shown in Table 2, the  $SQNR\ diff$  decision rule demonstrated higher accuracy than the output decision rule in all models. Therefore, the proposed mixed non-linear quantization method, based on  $SQNR\ diff$  values, was applied to the ViT, DeiT, and Swin models, and the final mixed mapping results are presented in Table 3. These results indicate that no single method consistently outperforms others across the individual non-linear operations of a single model.



**Fig. 3:** Layer-wise quantization sensitivity of non-linear operations for quantized DeiT-T model, our method (red) improves overall quantization sensitivity for each non-linear layers.

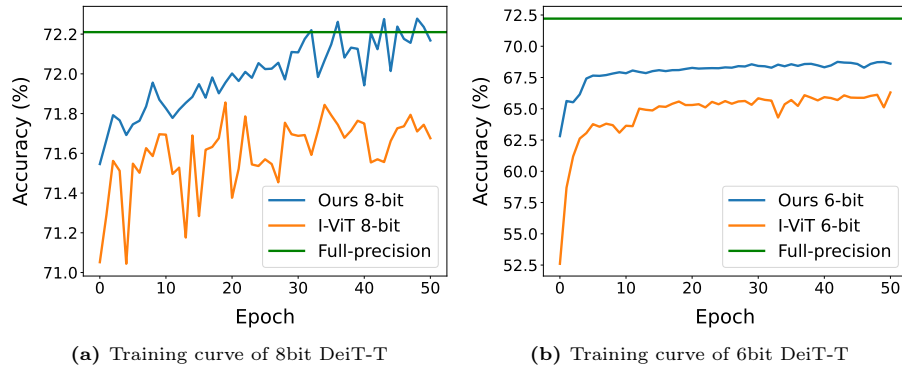
### 4.3 Accuracy Evaluation

Experiments were conducted comparing the proposed mixed non-linear quantization method with previous studies I-BERT, FQ-ViT, and I-ViT in terms of 8-bit and 6-bit quantization accuracy on ViT, DeiT, and Swin models. The results are presented in Table 4, showing an overall improvement in accuracy compared to existing quantized models, with ViT-S, ViT-B, DeiT-T, DeiT-S, and DeiT-B models even surpassing their full-precision counterparts. Specifically, the DeiT-T model achieved an accuracy of 72.55% with our 8-bit mixed non-linear quantization, which is 0.34%p higher than the full-precision model. It also exceeded the accuracy of I-ViT by 0.31%p, I-BERT by 1.22%p, and FQ-ViT by 0.94%p. For the 6-bit quantization comparison, results were reproduced and compared using official codes after sufficient retraining. The proposed method improved accuracy by an average of 35.5%p and 19.6%p over FQ-ViT and I-ViT, respectively.

The SQNR of each quantized non-linear layer after QAT is shown in the Figure 3. Experimental results indicate that, unlike previous methods where the SQNR values of the same non-linear operations decrease with increasing layer depth, leading to increased quantization errors, our proposed method maintains the highest SQNR in overall. This is due to our *SQNR diff*-based selection approach for each non-linear layer, which reduces the cumulative quantization error as layers deepen.

### 4.4 Training Time

The mixed non-linear quantization method is evaluated for its learning efficiency by measuring training time to achieve the accuracy with the QAT. For compar-



**Fig. 4:** Training curve of DeiT-T 6-bit and 8-bit models. Our method achieves faster training convergence and higher final accuracy than I-ViT QAT.

**Table 5:** 1-epoch training results for 8-bit quantization. Our method shows faster training speed than I-ViT and I-BERT. \* indicates the results that are re-produced by paper I-ViT [18].

Method	Bit-prec.	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S
I-ViT [18]	8bit	57.20	30.53	71.05	78.27	28.93	77.65	65.13
I-BERT* [13]	8bit	76.63	81.62	70.81	79.17	81.35	80.54	80.63
Ours	8bit	77.68	82.55	71.54	79.24	81.67	80.52	81.43

ison, training time was measured against the most recent studies on I-ViT QAT with the official code. Experimental results are shown in Figure 4, which indicates that at the same epoch, our method achieved higher accuracy in both 8-bit and 6-bit quantization, showing not only faster convergence but also higher final accuracy.

For rapid deployment, the accuracy improvement when applying QAT for only one epoch is shown in Table 5. Mixed non-linear quantization achieved higher Top-1 accuracy across all models compared to I-ViT and I-BERT. As for performance drop from full training to only one epoch of training, I-ViT showed an average accuracy decrease of -22.27%p, whereas the proposed method only reduced accuracy by an average of -1.33%p compared to the 8-bit quantization results of the Table 4. Thus, the proposed method demonstrates that it can quickly correct accuracy even when QAT is applied for just one epoch, making it suitable for rapid deployment.

## 5 Conclusion

In this paper, we proposed a mixed non-linear quantization method that considers the sensitivity of each non-linear layer. This method strategically assigns the most error-minimizing non-linear quantization method from the known non-

linear quantization operations. We also devised a novel SQNR difference metric, *SQNR diff*, to accurately assess layer-wise quantization effects. Experimental results demonstrated that our method significantly improved accuracy, outperforming existing methods such as I-BERT, FQ-ViT, and I-ViT in both 8-bit and 6-bit environments. Even under the constraint of a single epoch training time for rapid deployment, our approach still yielded higher accuracy improvements over existing methods.

## Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2023-00277060, Development of open edge AI SoC hardware and software platform). This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA(Defense Acquisition Program Administration)) (No.KRIT-CT-22-040, Heterogeneous Satellite constellation based ISR Research Center, 2024).

## References

1. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
2. Crandall, R.E., Pomerance, C.: Prime numbers: a computational perspective, vol. 2. Springer (2005)
3. Ding, Y., Qin, H., Yan, Q., Chai, Z., Liu, J., Wei, X., Liu, X.: Towards accurate post-training quantization for vision transformer. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 5380–5388 (2022)
4. Dong, P., Lu, L., Wu, C., Lyu, C., Yuan, G., Tang, H., Wang, Y.: Packqvint: Faster sub-8-bit vision transformers via full and packed quantization on the mobile. Advances in Neural Information Processing Systems **36** (2024)
5. Dong, Z., Yao, Z., Arfeen, D., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq-v2: Hessian aware trace-weighted quantization of neural networks. Advances in neural information processing systems **33**, 18518–18529 (2020)
6. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq: Hessian aware quantization of neural networks with mixed-precision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 293–302 (2019)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Du, D., Gong, G., Chu, X.: Model quantization and hardware acceleration for vision transformers: A comprehensive survey. arXiv preprint arXiv:2405.00314 (2024)
9. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. In: Low-Power Computer Vision, pp. 291–326. Chapman and Hall/CRC (2022)

10. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
11. Huang, X., Shen, Z., Cheng, K.T.: Variation-aware vision transformer quantization. arXiv preprint arXiv:2307.00331 (2023)
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2704–2713 (2018)
13. Kim, S., Gholami, A., Yao, Z., Mahoney, M.W., Keutzer, K.: I-bert: Integer-only bert quantization. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 5506–5518. PMLR (18–24 Jul 2021), <https://proceedings.mlr.press/v139/kim21d.html>
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
15. Li, Y., Xu, S., Lin, M., Cao, X., Liu, C., Sun, X., Zhang, B.: Bi-vit: Pushing the limit of vision transformer quantization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3243–3251 (2024)
16. Li, Y., Xu, S., Zhang, B., Cao, X., Gao, P., Guo, G.: Q-vit: Accurate and fully quantized low-bit vision transformer. Advances in neural information processing systems **35**, 34451–34463 (2022)
17. Li, Z., Yang, T., Wang, P., Cheng, J.: Q-vit: Fully differentiable quantization for vision transformer. arXiv preprint arXiv:2201.07703 (2022)
18. Li, Z., Gu, Q.: I-vit: Integer-only quantization for efficient vision transformer inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17065–17075 (October 2023)
19. Li, Z., Xiao, J., Yang, L., Gu, Q.: Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17227–17236 (2023)
20. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: Fq-vit: Post-training quantization for fully quantized vision transformer. In: International Joint Conference on Artificial Intelligence (2021), <https://api.semanticscholar.org/CorpusID:248300269>
21. Liu, S.Y., Liu, Z., Cheng, K.T.: Oscillation-free quantization for low-bit vision transformers. In: International Conference on Machine Learning. pp. 21813–21824. PMLR (2023)
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
23. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Advances in Neural Information Processing Systems **34**, 28092–28103 (2021)
24. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
25. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8612–8620 (2019)

26. Wang, W., Zhou, S., Sun, W., Sun, P., Liu, Y.: Sole: Hardware-software co-design of softmax and layernorm for efficient transformer inference. In: 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD). pp. 1–9. IEEE (2023)
27. Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M., et al.: Hawq-v3: Dyadic neural network quantization. In: International Conference on Machine Learning. pp. 11875–11886. PMLR (2021)
28. Yuan, Z., Zhou, R., Wang, H., He, L., Ye, Y., Sun, L.: Vit-1.58 b: Mobile vision transformers in the 1-bit era. arXiv preprint arXiv:2406.18051 (2024)
29. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In: European conference on computer vision. pp. 191–207. Springer (2022)
30. Zhang, Z., He, B., Zhang, Z.: Practical edge kernels for integer-only vision transformers under post-training quantization. *Proceedings of Machine Learning and Systems* **5**, 35–47 (2023)
31. Zhong, Y., Hu, J., Huang, Y., Zhang, Y., Ji, R.: Erq: Error reduction for post-training quantization of vision transformers. In: Forty-first International Conference on Machine Learning (2024)
32. Zhong, Y., Hu, J., Lin, M., Chen, M., Ji, R.: I&s-vit: An inclusive & stable method for pushing the limit of post-training vits quantization. arXiv preprint arXiv:2311.10126 (2023)