ERQ: Error Reduction for Post-Training Quantization of Vision Transformers

Yunshan Zhong [1 2] Jiawei Hu [2 3] You Huang [2 3] Yuxin Zhang [2 3] Rongrong Ji [1 2 3 4 *]

Abstract

Post-training quantization (PTQ) for vision transformers (ViTs) has garnered significant attention due to its efficiency in compressing models. However, existing methods typically overlook the intricate interdependence between quantized weight and activation, leading to considerable quantization error. In this paper, we propose **ERQ**, a two-step PTQ approach meticulously crafted to sequentially reduce the quantization error arising from activation and weight quantization. ERQ first introduces Activation quantization error reduction (Aqer) that strategically formulates the minimization of activation quantization error as a Ridge Regression problem, tackling it by updating weights with full-precision. Subsequently, ERQ introduces Weight quantization error reduction (Wqer) that adopts an iterative approach to mitigate the quantization error induced by weight quantization. In each iteration, an empirically derived, efficient proxy is employed to refine the rounding directions of quantized weights, coupled with a Ridge Regression solver to curtail weight quantization error. Experimental results attest to the effectiveness of our approach. Notably, ERQ surpasses the state-of-the-art GPTQ by 22.36% in accuracy for W3A4 ViT-S.

1. Introduction

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have significantly challenged the convolutional neural networks (CNNs), emerging as a new paradigm in the field of computer vision. ViTs leverage multi-head self-attention (MHSA) mechanics to capture long-range relationships among image patches, demonstrating impressive progress in a variety of vision tasks (Touvron et al., 2021; Carion et al., 2020; Zhu et al., 2020; Arnab et al., 2021).

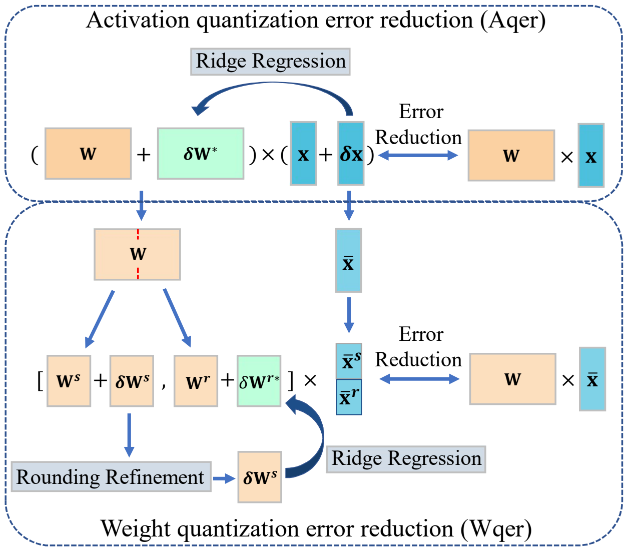
Activation quantization error reduction (Aqer)
Ridge Regression
$(\mathbf{W} + \delta\mathbf{W}^*) \times (\mathbf{x} + \delta\mathbf{x})$
Error Reduction
$\mathbf{W} \times \mathbf{x}$
$\mathbf{W}$
$\bar{\mathbf{x}}$
$[\mathbf{W}^s + \delta\mathbf{W}^s, \mathbf{W}^r + \delta\mathbf{W}^{r*}] \times$
$\bar{\mathbf{x}}^s$
$\bar{\mathbf{x}}^r$
Error Reduction
$\mathbf{W} \times \bar{\mathbf{x}}$
Rounding Refinement
$\delta\mathbf{W}^s$
Ridge Regression
Weight quantization error reduction (Wqer)

Figure 1. Illustration of the proposed ERQ.

However, great power comes with considerable complexity. The inherent architectural intricacies of ViTs result in high computational demands and substantial memory requirements, posing challenges for deployments in resource-constrained environments (Tang et al., 2022; Hou & Kung, 2022; Zheng et al., 2023). To mitigate this dilemma, model quantization has garnered sustained attention from both industry and academia (Krishnamoorthi, 2018). Quantization reduces model complexity by enabling low-bit representation of weight and activation, offering a promising pathway for efficient deployments. Recently, researchers have been gravitating towards post-training quantization (PTQ) for ViTs, which seeks to quantize models with a tiny calibration dataset and minor costs (Li et al., 2022b; Lin et al., 2022; Liu et al., 2023b; Frumkin et al., 2023).

Various PTQ methods have been explored to accommodate the ViTs' unique structure. For example, for handling long-tail post-Softmax activation, the log2/log$\sqrt{2}$ quantizer (Li et al., 2023; Lin et al., 2022) and the twin uniform quantizer (Yuan et al., 2022) are introduced. To manage high

[1]Institute of Artificial Intelligence, Xiamen University. [2]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University. [3]Department of Artificial Intelligence, School of Informatics, Xiamen University. [4]Peng Cheng Laboratory.. Correspondence to: Rongrong Ji <rrji@xmu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

variant activation, reparameterization technique (Li et al., 2023) and power-of-two factor are employed. Additionally, evolutionary search methods (Frumkin et al., 2023) are utilized for determining unstable scale factors. Nevertheless, existing methods typically overlook the intricate interdependence between weight and activation quantization, leading to considerable quantization error when it comes to weight-activation quantization.

In this paper, we introduce **ERQ**, a two-step post-training quantization method tailored for ViTs, that sequentially mitigates quantization error induced by quantized activations and weights. As shown in Fig. 1, ERQ consists of two steps, i.e., Activation quantization error reduction (Aqer) followed by Weight quantization error reduction (Wqer). Aqer formulates a Ridge Regression problem to mitigate the quantization error induced by activation quantization, which can be solved with a closed-form solution via weight updating. Subsequently, Wqer is introduced to mitigate the quantization error caused by weight quantization in an iterative quantization-and-correction manner. In particular, at each iteration, we quantize the first half of the full-precision weight and mitigate the resulting quantization error by first performing Rounding Refinement and then again solving a Ridge Regression problem. The former derives an efficient proxy for output error, which is used to refine the rounding directions of quantized weight to lower the quantization error. The latter further mitigates the quantization error by updating the remaining full-precision weight. Such a process continuously performs until all weights are accurately quantized.

The proposed ERQ's effectiveness is demonstrated in extensive experiments on various ViTs variants (ViT, DeiT, and Swin) and tasks (image classification, object detection, and instance segmentation). Notably, on the image classification task, ERQ outperforms GPTQ by 22.36% for W3A4 ViT-S.

2. Related Work

2.1. Vision Transformers (ViTs)

Inspired by the success of transformers in the natural language processing field, ViTs, by treating images as patch tokens, have emerged as a groundbreaking development in computer vision (Dosovitskiy et al., 2021). Addressing the dependency of ViTs on large datasets, DeiT (Touvron et al., 2021) showcases an efficient teacher-student training approach. Then, Swin Transformers (Liu et al., 2021a) employs a hierarchical structure that integrates a shifted window-based self-attention mechanism, marking further improvements. The applications of ViTs has broadened considerably, including areas such as object detection (Carion et al., 2020; Zhu et al., 2020), image segmentation (Zheng et al., 2021), low-level image processing (Liang et al., 2021), video classification (Arnab et al., 2021), and medical imaging (Shamshad et al., 2023), *etc*. However, ViTs are accompanied by substantial computational overhead and increased memory requirements, posing challenges for their deployment in resource-constrained environments (Mehta & Rastegari, 2022; Zhang et al., 2022).

2.2. Post-training Quantization for ViTs

Model quantization reduces the numerical precision of weight and activation to mitigate computational and storage costs of neural networks (Krishnamoorthi, 2018). In contrast to quantization-aware training (QAT) (Li et al., 2022a; Li & Gu, 2023; Xijie Huang & Cheng, 2023) that involves complete training data and compute-heavy retraining, post-training quantization (PTQ) operates on a tiny dataset with a reduced time overhead, harvesting extensive attention (Banner et al., 2019). The unique architecture of ViTs, such as LayerNorm and attention mechanisms, makes distinct PTQ methods compared to those used for convolutional neural networks (CNNs) (Li et al., 2021; Wei et al., 2022). Liu *et al.* (Liu et al., 2021b) introduce the first PTQ method for ViTs. To maintain the order of softmax scores and adapt various quantization sensitivities of different layers, they respectively introduce a ranking loss and a nuclear norm-based mixed-precision scheme. FQ-ViT (Lin et al., 2022) introduces a fully-quantized method, which respectively designs Powers-of-Two Scale and Log-Int-Softmax for post-LayerNorm and post-Softmax activation. In PTQ4ViT (Yuan et al., 2022), a twin uniform quantizer is introduced to handle the long-tail post-Softmax activation and uneven post-GELU activation. APQ-ViT (Ding et al., 2022) establishes a block-wise error reconstruction and a Matthew-effect preserving quantizer for post-Softmax activation. In Evol-Q (Frumkin et al., 2023), an evolutionary search method is employed to search extremely sensitive quantization parameters. RepQ-ViT (Li et al., 2023) introduces a reparameterization technique to handle high-variant post-LayerNorm activation, where the channel-wise quantizers are simplified to layer-wise quantizers. Also, a Log$\sqrt{2}$ quantizer is adopted to accommodate post-Softmax activation. GPTQ (Frantar et al., 2022) employs OBS (Frantar & Alistarh, 2022) to progressively compensate for weight quantization error by utilizing Hessian information. Despite sharing a certain similarity with GPTQ, our ERQ introduces Aqer for mitigating the error of quantized activation, while GPTQ does not quantize activations. Moreover, ERQ uses a derived proxy for output error to refine weight rounding, which has not been proposed before.

3. Preliminaries

Quantizers. For a fair comparison, our quantization settings are aligned with the earlier work (Li et al., 2023).

Specifically, we quantize the weight and activation of all matrix multiplications in ViTs. The channel-wise quantizer and layer-wise quantizer are adopted for weight and activation, respectively. For weights and the activation except for the post-Softmax activation, we adopt the uniform quantizer. Given full-precision values $\mathbf{x}$ and the bit-width b, the uniform quantizer is defined as:

$$\bar{\mathbf{x}} = \mathrm{Q}_{un}(\mathbf{x}, b) = s \cdot \mathrm{clip}\left(\left\lfloor \frac{\mathbf{x}}{s} \right\rceil + z, 0, 2^b - 1\right), \quad (1)$$

where $\lfloor\cdot\rceil$ denotes the round function, clip function makes the output between 0 and $2^b - 1$, the scale factor s is grid-searched by minimizing the error before and after quantization, and the zero-point $z = \left\lfloor -\frac{\min(\mathbf{x})}{s} \right\rceil$. For long-tail post-Softmax activation, the $\log\sqrt{2}$ quantizer (Li et al., 2023) is adopted:

$$\bar{\mathbf{x}} = \mathrm{Q}_{lg\sqrt{2}}(\mathbf{x}, b) = s \cdot 2^{\lfloor -\frac{x_q}{2} \rfloor}(\mathbb{1}(\mathbf{x}_q)(\sqrt{2} - 1) + 1), \quad (2)$$

$$\mathbf{x}_q = \mathrm{clip}\left(\left\lfloor -2\log_2 \frac{\mathbf{x}}{s} \right\rceil, 0, 2^b - 1\right), \quad (3)$$

where $\mathbb{1}(\cdot)$ returns 0 for even numbers and 1 for odd numbers, s is grid-searched by minimizing the error before and after quantization. All scale factors of the above-mentioned quantizers are determined by the calibration datasets.

Objective. Denoting the full-precision activation as $\mathbf{x} \sim \mathcal{P}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^{D_{in}}$, and the weight $\mathbf{W} \in \mathbb{R}^{D_{out} \times D_{in}}$. Here, D_{in} and D_{out} are the input and output dimensions, respectively. The quantization error induced by activation and weight quantization is denoted as $\delta\mathbf{x} = \bar{\mathbf{x}} - \mathbf{x}$ and $\delta\mathbf{W} = \bar{\mathbf{W}} - \mathbf{W}$, respectively. For each layer, we aim to minimize the Mean Squared Error (MSE) before and after weight-activation quantization:

$$\begin{aligned} \mathcal{L}^{\mathrm{MSE}} &= \mathbb{E}\left[\|\mathbf{W}\mathbf{x} - \bar{\mathbf{W}}\bar{\mathbf{x}}\|_2^2\right] \\ &= \mathbb{E}\left[\|\mathbf{W}\mathbf{x} - (\mathbf{W} + \delta\mathbf{W})(\mathbf{x} + \delta\mathbf{x})\|_2^2\right]. \end{aligned} \quad (4)$$

Eq. 4 indicates that output error is contributed both by activations and weight quantization error.

4. Method

The entangled $\delta\mathbf{x}$ and $\delta\mathbf{W}$ make it a challenge to find an optimal solution for Eq. 4 (Li et al., 2021). To make it tractable, we relax Eq. 4 to two sequential sub-problems by respectively minimizing error from quantized activation and weight. As shown in Fig. 1, we first perform Activation quantization error reduction (Aqer) followed by Weight quantization error reduction (Wqer), which respectively detailed in the following.

4.1. Activation Quantization Error Reduction

To mitigate error induced by activation quantization, we introduce Activation quantization error reduction (Aqer), which formulates the error mitigation as the Ridge Regression problem. Specifically, we retain the weight as full-precision and solely consider the MSE caused by activation quantization error $\delta\mathbf{x}$ as:

$$\mathcal{L}^{\mathrm{MSE}} = \mathbb{E}\left[\|\mathbf{W}\mathbf{x} - \mathbf{W}(\mathbf{x} + \delta\mathbf{x})\|_2^2\right]. \quad (5)$$

To minimize Eq. 5, we formulate the Ridge Regression problem, where the minimization is completed by adding $\mathbf{W}$ with an adjustment $\delta\mathbf{W}^*$:

$$\begin{aligned} &\mathbb{E}\left[\|\mathbf{W}\mathbf{x} - (\mathbf{W} + \delta\mathbf{W}^*)(\mathbf{x} + \delta\mathbf{x})\|_2^2\right] + \lambda_1\|\delta\mathbf{W}^*\|_2^2 \\ &= \mathbb{E}\left[\| - \delta\mathbf{W}^*(\mathbf{x} + \delta\mathbf{x}) - \mathbf{W}\delta\mathbf{x}\|_2^2\right] + \lambda_1\|\delta\mathbf{W}^*\|_2^2 \\ &= \mathbb{E}\left[\|\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x}\|_2^2\right] + \lambda_1\|\delta\mathbf{W}^*\|_2^2. \end{aligned} \quad (6)$$

Here, $\delta\mathbf{W}^*$ denotes adjustment that is computed by Ridge Regression, $\bar{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$ is the quantized input, $\lambda_1\|\delta\mathbf{W}^*\|_2^2$ acts as the regularization term, λ_1 is a hyper-parameter that control the intensity of the regularization. Eq. 6 constitutes the Ridge Regression problem. To minimize it, we first compute its gradient *w.r.t.* $\delta\mathbf{W}^*$:

$$\begin{aligned} &\frac{\partial}{\partial\delta\mathbf{W}^*}\mathbb{E}\left[\|\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x}\|_2^2\right] + \lambda_1\|\delta\mathbf{W}^*\|_2^2 \\ &\qquad = \mathbb{E}\left[2(\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x})\bar{\mathbf{x}}^T\right] + 2\lambda_1\delta\mathbf{W}^*. \end{aligned} \quad (7)$$

Then, we solve for $\delta\mathbf{W}^*$ by setting Equation 7 to zero:

$$\begin{aligned} &\mathbb{E}\left[2(\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x})\bar{\mathbf{x}}^T\right] + 2\lambda_1\delta\mathbf{W}^* = 0 \\ &\Rightarrow \delta\mathbf{W}^* = -\mathbf{W}\mathbb{E}\left[\bar{\mathbf{x}}\delta\mathbf{x}^T\right](\mathbb{E}\left[\bar{\mathbf{x}}\bar{\mathbf{x}}^T\right] + \lambda_1\mathbf{I})^{-1}. \end{aligned} \quad (8)$$

The regularization term $\lambda_1\mathbf{I}$ ensures the inverse of $\mathbb{E}\left[\bar{\mathbf{x}}\bar{\mathbf{x}}^T\right] + \lambda_1\mathbf{I}$ always exists, which is crucial for computational stability. In addition, it suppresses outliers, thereby mitigating overfitting and enhancing the model's generalizability. Suppressing outliers is also crucial for subsequent weight quantization since it restricts the range of weight. This restriction prevents the quantization points from being distributed in the uncovered region, thus enhancing the expressive ability of quantization (Li et al., 2020). In practice, given the calibration dataset, we estimate $\mathbb{E}\left[\bar{\mathbf{x}}\delta\mathbf{x}^T\right]$ and $\mathbb{E}\left[\bar{\mathbf{x}}\bar{\mathbf{x}}^T\right]$ using $\frac{1}{N}\sum_n^N \bar{\mathbf{x}}_n\delta\mathbf{x}_n^T$ and $\frac{1}{N}\sum_n^N \bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^T$, respectively. Here, $N = B \times T >> D_{in}^s$, where B is the size of the calibration dataset, and T is the number of tokens of one image. Note that $\delta\mathbf{x}$ and $\bar{\mathbf{x}}$ are determined given the input and the quantization parameters. After obtaining $\delta\mathbf{W}^*$, we incorporate it into the network's weight by $\mathbf{W} = \mathbf{W} + \delta\mathbf{W}^*$. By doing so, the proposed Aqer explicitly mitigates the quantization error from quantized activation into the weight.

4.2. Weight Quantization Error Reduction

After Aqer, we perform weight quantization and propose Weight quantization error reduction (Wqer) to mitigate the

Algorithm 1 Weight Quantization Error Reduction

```
1: Input: $\mathbf{W}$, $\bar{\mathbf{W}} = \varnothing$, $\{\bar{\mathbf{x}}_n\}_{n=1}^N$, maximum iteration T
2: for i in range($D_{out}$):
3:     $\bar{\mathbf{W}}_{i,:} = \varnothing$, $\{\bar{\mathbf{x}}_n\}_{n=1}^N = \{\bar{\mathbf{x}}_n\}_{n=1}^N$
4:     while $|\mathbf{W}_{i,:}| > 0$
5:         Partition $\mathbf{W}_{i,:}$ into $\left[\mathbf{W}^s_{i,:}, \mathbf{W}^r_{i,:}\right]$
6:         Partition $\{\bar{\mathbf{x}}_n\}_{i=1}^N$ into $\{\left[\bar{\mathbf{x}}^s_n, \bar{\mathbf{x}}^r_n\right]\}_{i=1}^N$
7:         /* Rounding Refinement */
8:         Obtain $\hat{\boldsymbol{\mu}}^s\hat{\boldsymbol{\mu}}^{sT} + \hat{\boldsymbol{\Sigma}}^s$ from cache or calculate it with $\{\bar{\mathbf{x}}^s_n\}_{n=1}^N$, calculate $\delta\mathbf{W}^{s\downarrow}_{i,:}$, $\delta\mathbf{W}^{s\uparrow}_{i,:}$ with $\mathbf{W}^s_{i,:}$
9:         while $0 \leq$ T--:
10:            Calculate proxy $\mathcal{L}_{old}$ with $\delta\mathbf{W}^s_{i,:}$ by Eq. 12
11:            Calculate gradients $\boldsymbol{G}_{\delta\mathbf{W}^s_{i,:}}$ by Eq. 14
12:            Obtain $\mathcal{S}$ by Eq. 15
13:            Obtain adjusted $\delta\mathbf{W}^{'}_{i,:}$ by Eq. 13
14:            Calculate proxy $\mathcal{L}_{now}$ with $\delta\mathbf{W}^{'}_{i,:}$ by Eq. 12
15:            if $\mathcal{L}_{now} > \mathcal{L}_{old}$: break
16:            $\delta\mathbf{W}^s_{i,:} = \delta\mathbf{W}^{'}_{i,:}$
17:        $\bar{\mathbf{W}}_{i,:} \leftarrow \bar{\mathbf{W}}_{i,:} \cup (\mathbf{W}^s_{i,:} + \delta\mathbf{W}^s_{i,:})$
18:        /* END Rounding Refinement */
19:        /* Ridge Regression */
20:        Calculate $\delta\mathbf{W}^{r*}_{i,:}$ by Eq. 17
21:        $\mathbf{W}_{i,:} \leftarrow \mathbf{W}^r_{i,:} + \delta\mathbf{W}^{r*}_{i,:}$
22:        /* END Ridge Regression */
23:        $\{\bar{\mathbf{x}}_n\}_{n=1}^N \leftarrow \{\bar{\mathbf{x}}^r_n\}_{n=1}^N$
24:    $\bar{\mathbf{W}} \leftarrow \bar{\mathbf{W}} \cup \bar{\mathbf{W}}_{i,:}$
25: Output: Quantized weight $\bar{\mathbf{W}}$
```

resulting quantization error. Here, the target is defined as:

$$\mathcal{L}^{\text{MSE}} = \mathbb{E}\left[\|\mathbf{W}\bar{\mathbf{x}} - (\mathbf{W} + \delta\mathbf{W})\bar{\mathbf{x}}\|_2^2\right] = \sum_i^{D_{out}} \mathcal{L}_i^{\text{MSE}} \\ = \sum_i^{D_{out}} \mathbb{E}\left[\|\mathbf{W}_{i,:}\bar{\mathbf{x}} - (\mathbf{W}_{i,:} + \delta\mathbf{W}_{i,:})\bar{\mathbf{x}}\|_2^2\right]. \tag{9}$$

Note that after Aqer, the activation is quantized. Eq. 9 indicates that the minimization across output channels operates independently. Consequently, we analyze the minimization of each $\mathcal{L}_i^{\text{MSE}}$ individually. Simultaneously quantizing the entire full-precision weight yields unrecoverable quantization error (Frantar et al., 2022). Thus, we adopt an iterative quantization-and-correction manner to gradually minimize quantization error caused by weight quantization (Zhou et al., 2017). In each iteration, the first half of unquantized weights is quantized, followed by a mitigation of the resulting quantization error. Specifically, we begin with the current full-precision weight $\mathbf{W}_{i,:}$ and the corresponding $\bar{\mathbf{x}}$. We then partition $\mathbf{W}$ into two segments: the first half, $\mathbf{W}^s_{i,:} \in \mathbb{R}^{1\times D^s_{in}}$, is designated for quantization, while the remaining part, $\mathbf{W}^r_{i,:} \in \mathbb{R}^{1\times D^r_{in}}$, is retained at full-precision. Correspondingly, we derive $\bar{\mathbf{x}}^s \in \mathbb{R}^{D^s_{in}}$ and $\bar{\mathbf{x}}^r \in \mathbb{R}^{D^r_{in}}$ from $\bar{\mathbf{x}}$, where $\bar{\mathbf{x}}^s$ and $\bar{\mathbf{x}}^r$ respectively contains the rows of $\bar{\mathbf{x}}$ corresponding to $\mathbf{W}^s_{i,:}$ and $\mathbf{W}^r_{i,:}$. The quantization error of the quantized $\mathbf{W}^s_{i,:}$ is denoted as $\delta\mathbf{W}^s_{i,:} = \bar{\mathbf{W}}^s_{i,:} - \mathbf{W}^s_{i,:}$, and the resulting MSE is:

Table 1. Results of W4A4 DeiT-S with different methods for minimizing $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$. "baseline" indicates only performing calibration and no error reduction is involved.

Method	Time costs	Acc. (%)
Baseline	$\sim$ 1 minute	68.41
+ MIPQ w/o Proxy	$\sim$130 hours	69.67
+ MIPQ w/ Proxy	$\sim$10 hours	69.55
+ Rounding Refinement	$\sim$4 minutes	69.24

$$\mathcal{L}_i^{\text{MSE}} = \mathbb{E}\big[\|[\mathbf{W}^s_{i,:}, \mathbf{W}^r_{i,:}][\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^r] \\ - [\mathbf{W}^s_{i,:} + \delta\mathbf{W}^s_{i,:}, \mathbf{W}^r_{i,:}][\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^r]\|_2^2\big] \\ = \mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]. \tag{10}$$

Here, $\mathbf{W}_{i,:} = [\mathbf{W}^s_{i,:}, \mathbf{W}^r_{i,:}]$, $\bar{\mathbf{x}} = [\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^r]$. To mitigate Eq. 10, we first introduce Rounding Refinement, in which the rounding direction of the quantized weight is refined, i.e., adjusting $\delta\mathbf{W}^s_{i,:}$, to reduce $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$ itself. Then, given $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$ after Rounding Refinement, we formulate a Ridge Regression problem to further mitigate it by adjusting $\mathbf{W}^r_{i,:}$.

4.2.1. Rounding Refinement

At first, we aim to adjust the rounding direction of quantized weights to minimize $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$. Specifically, for the j-th value in $\mathbf{W}^s_{i,:}$, denoted as $\mathbf{W}^s_{i,j}$, the quantization involves rounding it either to the floor or ceil (Nagel et al., 2020a). Thereby the quantization error for $\mathbf{W}^s_{i,:}$, denoted as $\delta\mathbf{W}^s_{i,j}$, can be represented as either $\delta\mathbf{W}^{s\downarrow}i,j$ or $\delta\mathbf{W}^{s\uparrow}i,j$. Here, $\delta\mathbf{W}^{s\downarrow}_{i,j} = \mathbf{W}^s_{i,j} - \text{Q}_{un\downarrow}(\mathbf{W}^s_{i,j}, b) > 0$ denotes error from rounding-to-floor strategy, $\delta\mathbf{W}^{s\uparrow}_{i,j} = \mathbf{W}^s_{i,j} - \text{Q}_{un\uparrow}(\mathbf{W}^s_{i,j}, b) < 0$ denotes error from rounding-to-ceil strategy, where $\downarrow$ / $\uparrow$ denote replacing $\lfloor\cdot\rceil$ in Eq. 1 with $\lfloor\cdot\rfloor$/$\lceil\cdot\rceil$. The selection of $\delta\mathbf{W}^s_{i,:}$ is an NP-hard problem, whose solution can be searched by the mixed-integer quadratic program (MIPQ) (Pia et al., 2017; Kuzmin et al., 2023). However, the high computational complexity of $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$ makes it a challenge to find the solution with reasonable time costs. As shown in Tab. 1, using $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$ as the target of MIPQ consumes prohibitive $\sim$130 hours.

Efficient Proxy. Therefore, we aim to find an efficient proxy for $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$. First, we re-write $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right]$ as:

$$\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|_2^2\right] \triangleq (\mathbb{E}\left[\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\right])^2 + \text{Var}\left[\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\right]. \tag{11}$$

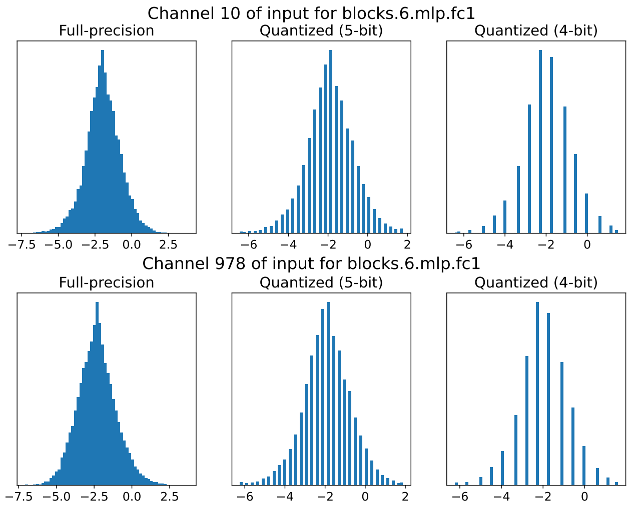

Figure 2. Distribution of activation on different channels. Results are extracted from DeiT-S with 32 images.

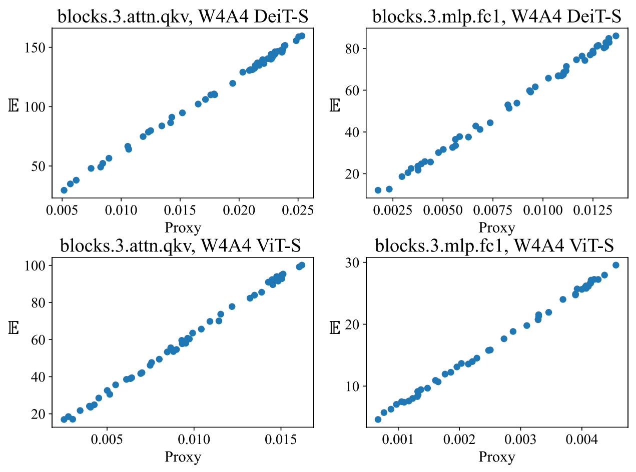

Figure 3. $\mathbb{E}$ denotes $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|^2_2\right]$. The proxy values are positively correlated with the real values.

Here, Δ indicates utilizing $\mathbb{E}\left[Z^2\right] = (\mathbb{E}\left[Z\right])^2 + \text{Var}\left[Z\right]$. As proved by (Klambauer et al., 2017), according to the central limit theorem, the numerous multiplication and addition operations within neural networks make the activation generally follow a Gaussian distribution, which is also a basic assumption in many previous works in the quantization field (Ding et al., 2019; Sun et al., 2022; Lin et al., 2022; Chmiel et al., 2020). Meanwhile, Fig. 2 illustrates the channel distribution of the full-precision and quantized activation. It can be seen that quantized activation continues to exhibit an approximated Gaussian distribution (Krishnamoorthi, 2018). Thus, we consider channel distribution of $\bar{\mathbf{x}}^s$ still can be captured by the Gaussian distribution, and model $\bar{\mathbf{x}}^s$ with a D^s_{in}-dimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^s, \boldsymbol{\Sigma}^s)$, where D^s_{in} is the dimension of $\bar{\mathbf{x}}^s$, $\boldsymbol{\mu}^s \in \mathbb{R}^{D^s_{in}}$, $\boldsymbol{\Sigma}^s \in \mathbb{R}^{D^s_{in}\times D^s_{in}}$. Then, the Eq. 11 becomes:

$$\begin{aligned}&\mathbb{E}\left[\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\right]^2 + \text{Var}\left[\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\right]\\&= \delta\mathbf{W}^s_{i,:}\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT}(\delta\mathbf{W}^s_{i,:})^T + \delta\mathbf{W}_{i,:}\boldsymbol{\Sigma}^s(\delta\mathbf{W}^s_{i,:})^T \quad (12)\\&= \delta\mathbf{W}^s_{i,:}(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s)(\delta\mathbf{W}^s_{i,:})^T.\end{aligned}$$

Here, Eq. 12 is the obtained proxy for $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|^2_2\right]$. In practice, we estimate the empirical $\hat{\boldsymbol{\mu}}^s$ and $\hat{\boldsymbol{\Sigma}}^s$ with the given calibration dataset. Note that for all output channels, $\hat{\boldsymbol{\mu}}^s$ and $\hat{\boldsymbol{\Sigma}}^s$ are shared and require only a single computation. Fig. 3 presents the relationship between the proxy and $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|^2_2\right]$. It can be seen that the proposed proxy is proportional to the real value, demonstrating its fidelity.

The computational complexity of using our proxy is $O((D^s_{in})^2)$, while the complexity of $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|^2_2\right]$ is $O(ND^s_{in})$, where $N >> D^s_{in}$. Thus, the proxy can serve as a low-cost objective for solving $\delta\mathbf{W}^s_{i,:}$. As shown in Tab. 1, using Eq. 12 as the target of MIPQ reduces the time costs from $\sim$130 hours to $\sim$10 hours. However, this still incurs moderate costs since current open-source implementations of MIPQ only support CPU and cannot fully exploit the capacity of GPU. In the next, we introduce Rounding Refinement, a GPU-support method that uses the gradient of the proxy to adjust $\delta\mathbf{W}^s_{i,:}$ faster.

Rounding Refinement. At first, we initialize $\delta\mathbf{W}^s_{i,j}$ with the rounding-to-nearest strategy. Now, $\delta\mathbf{W}^s_{i,j}$ is either equal to $\delta\mathbf{W}^{s\downarrow}_{i,j}$ or $\delta\mathbf{W}^{s\uparrow}_{i,j}$. Then, we aim to determine an index set $\mathcal{S}$ that contains the index set of the elements necessitating modifications, whose rounding direction is overturned:

$$\delta\mathbf{W}^s_{i,j} = \begin{cases}\delta\mathbf{W}^{s\downarrow}_{i,j} & \text{if } \delta\mathbf{W}^s_{i,j} = \delta\mathbf{W}^{s\uparrow}_{i,j}\\ \delta\mathbf{W}^{s\uparrow}_{i,j} & \text{otherwise.}\end{cases}, j \in \mathcal{S}. \quad (13)$$

To determine $\mathcal{S}$, we first take the derivative of the proxy (Eq. 12) *w.r.t* the $\delta\mathbf{W}^s_{i,:}$:

$$\begin{aligned}\boldsymbol{G}_{\delta\mathbf{W}^s_{i,:}} &= \frac{\partial}{\partial\delta\mathbf{W}^s_{i,:}}\delta\mathbf{W}^s_{i,:}(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s)(\delta\mathbf{W}^s_{i,:})^T\\&= 2\delta\mathbf{W}^s_{i,:}(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s).\end{aligned} \quad (14)$$

We only select the elements whose gradients are the same sign, since this is the only way to allow overturn. For example, given $\delta\mathbf{W}^s_{i,j} = \delta\mathbf{W}^{s\downarrow}_{i,j}$, replacing it by $\delta\mathbf{W}^{s\uparrow}_{i,j}$ is feasible only if $\boldsymbol{G}_{\delta\mathbf{W}^s_{i,j}}$ has the same sign as $\delta\mathbf{W}^s_{i,j}$. Thus, the index set $\mathcal{S}$ is defined as:

$$\begin{aligned}&\mathcal{S} = \text{topk\_index}(\mathcal{M}),\\&\mathcal{M} = |\boldsymbol{G}_{\delta\mathbf{W}^s_{i,:}} \odot \mathbb{1}(\boldsymbol{G}_{\delta\mathbf{W}^s_{i,:}} \odot \delta\mathbf{W}^s_{i,:})| \in \mathbb{R}^{D^s_{in}}.\end{aligned} \quad (15)$$

Here, topk_index return the index of the top k elements, $\mathbb{1}(\cdot)$ returns 1 for non-negative input and 0 for negative input, $|\cdot|$ returns the absolute value of the input. After obtaining $\mathcal{S}$,

Table 2. Results on ImageNet dataset. The top-1 accuracy (%) is reported as the metric. "W/A" indicates that the bit-width of the weight and activation are W and A bits, respectively. "*" indicates the results are re-produced by using the official code. More results are provided in the appendix.

Method	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT* (Lin et al., 2022)	3/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT* (Yuan et al., 2022)	3/4	0.10	0.10	0.20	0.15	0.59	0.64	0.53
GPTQ* (Frantar et al., 2022)	3/4	23.32	44.63	42.25	48.95	61.75	66.71	71.43
RepQ-ViT* (Li et al., 2023)	3/4	15.65	26.98	29.34	45.82	58.92	59.83	44.17
AdaRound* (Nagel et al., 2020b)	3/4	11.04	4.72	36.05	33.56	62.50	68.12	53.92
BRECQ* (Li et al., 2021)	3/4	4.97	1.25	29.23	18.58	40.49	66.93	53.38
QDrop* (Wei et al., 2022)	3/4	9.77	11.87	17.85	30.27	61.12	73.47	74.33
PD-Quant* (Liu et al., 2023a)	3/4	4.56	21.81	41.87	41.65	53.63	70.07	56.48
ERQ (Ours)	3/4	**45.68**	**53.88**	**44.09**	**57.63**	**70.33**	**75.08**	**75.78**
FQ-ViT (Lin et al., 2022)	4/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT (Yuan et al., 2022)	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT (Ding et al., 2022)	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
GPTQ* (Frantar et al., 2022)	4/4	67.59	75.12	58.96	70.85	76.10	80.17	81.08
RepQ-ViT (Li et al., 2023)	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
AdaRound* (Nagel et al., 2020b)	4/4	63.09	70.51	55.65	69.24	75.20	76.05	78.12
BRECQ* (Li et al., 2021)	4/4	11.31	3.03	38.41	32.89	59.10	68.40	56.51
QDrop* (Wei et al., 2022)	4/4	17.77	21.72	31.65	35.79	65.47	78.92	80.49
PD-Quant* (Liu et al., 2023a)	4/4	32.64	34.86	58.50	64.85	60.06	77.04	75.84
ERQ (Ours)	4/4	**68.91**	**76.63**	**60.29**	**72.56**	**78.23**	**80.74**	**82.44**
FQ-ViT* (Lin et al., 2022)	5/5	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT* (Yuan et al., 2022)	5/5	72.74	72.32	65.00	70.26	72.65	80.90	81.87
GPTQ* (Frantar et al., 2022)	5/5	78.63	82.06	69.05	77.12	80.17	82.19	83.00
RepQ-ViT* (Li et al., 2023)	5/5	78.43	82.03	69.00	77.04	80.08	82.08	83.22
AdaRound* (Nagel et al., 2020b)	5/5	77.53	82.00	68.87	76.22	80.18	82.12	84.09
BRECQ* (Li et al., 2021)	5/5	47.35	43.51	62.12	63.15	75.61	80.66	82.31
QDrop* (Wei et al., 2022)	5/5	56.32	57.92	62.36	70.07	78.41	81.73	83.61
PD-Quant* (Liu et al., 2023a)	5/5	65.06	58.40	68.02	74.94	74.61	81.27	82.12
ERQ (Ours)	5/5	**78.83**	**82.81**	**69.42**	**77.58**	**80.65**	**82.44**	**84.50**

the overturn is performed with Eq. 13. The above process iterates until the adjusted $\delta\mathbf{W}^s_{i,:}$ incurs a larger proxy value or reaches maximum iterations. After obtaining $\delta\mathbf{W}^s_{i,:}$, the quantization can be completed by $\bar{\mathbf{W}}^s_{i,:} = \mathbf{W}^s_{i,:} + \delta\mathbf{W}^s_{i,:}$. $\bar{\mathbf{W}}^s_{i,:}$ is then added into the set of quantized weight. The overall process of Rounding Refinement is presented in Lines 7 - Lines 18 of Alg. 1. As shown in Tab. 1, Rounding Refinement significantly reduces the time costs from 10 hours to 4 minutes by 150× at the cost of affordable accuracy loss.

4.2.2. Ridge Regression

After Rounding Refinement, we suggest adjusting $\mathbf{W}^r_{i,:}$ with $\delta\mathbf{W}^{r*}_{i,:}$ to further counteract $\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s\|^2_2\right]$, which yields the following target:

$$\mathbb{E}\left[\|\delta\mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s + \delta\mathbf{W}^{r*}_{i,:}\bar{\mathbf{x}}^r\|^2_2\right] + \lambda_2\|\delta\mathbf{W}^{r*}_{i,:}\|^2_2, \tag{16}$$

where λ_2 is a hyper-parameter to control intensity of the regularization term $\lambda_2\|\delta\mathbf{W}^{r*}_{i,:}\|^2_2$. The minimization of Eq. 16 formulates the Ridge Regression problem and the solution is defined as:

$$\delta\mathbf{W}^{r*}_{i,:} = -\delta\mathbf{W}^s_{i,:}\mathbb{E}\left[\bar{\mathbf{x}}^s\bar{\mathbf{x}}^{rT}\right]\left(\mathbb{E}\left[\bar{\mathbf{x}}^r\bar{\mathbf{x}}^{rT}\right] + \lambda_2\mathbf{I}\right)^{-1}. \tag{17}$$

In practice, we estimate $\mathbb{E}\left[\bar{\mathbf{x}}^r\bar{\mathbf{x}}^{sT}\right]$ and $\mathbb{E}\left[\bar{\mathbf{x}}^r\bar{\mathbf{x}}^{rT}\right]$ by using $\frac{1}{N}\sum_n^N \bar{\mathbf{x}}^r_n\bar{\mathbf{x}}^{sT}_n$ and $\frac{1}{N}\sum_n^N \bar{\mathbf{x}}^r_n\bar{\mathbf{x}}^{rT}_n$. Afterward, $\mathbf{W}^r_{i,:} = \mathbf{W}^r_{i,:} + \delta\mathbf{W}^{r*}_{i,:}$ to mitigate the error. Currently, $\mathbf{W}^r_{i,:}$ remains as full-precision and will be processed in the next iteration.

Such a process continuously runs until all weights are accurately quantized. The proposed Rounding Refinement and Ridge Regression collectively form Wqer, whose overall process is presented in Alg. 1. In practice, we perform the Wqer for multiple output channels in parallel.

5. Experiments

5.1. Implementation details

Models and datasets. We conduct extensive experiments on image classification, object detection, and instance segmentation. For the image classification task, we evaluate the ERQ on the ImageNet dataset (Russakovsky et al., 2015), with different ViT variants including ViT (Dosovitskiy et al., 2021), DeiT (Touvron et al., 2021), and Swin (Liu et al., 2021a). As for object detection and instance segmentation tasks, we evaluate ERQ on the COCO dataset (Lin et al., 2014) with Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai & Vasconcelos, 2018), both using Swin (Liu et al., 2021a) as their backbone.

Implementation details. Consistent with previous study (Li et al., 2023), we randomly select 32 images each from the ImageNet and 1 image from the COCO dataset. The quantization parameters are determined by forwarding the calibration datasets, and the reparameterization technique is used to initialize the activation quantizer as in (Li et al., 2023). In our experiments, the k and maximum iteration of Rounding Refinement are set to 1 and 100, respectively. We use the pulp (a CPU-only LP modeler written in Python) to solve the MIPQ. For the image classification task, we set $\lambda_1 = \lambda_2 = 1e4$ for ViT, $\lambda_1 = \lambda_2 = 1e3$ for DeiT-T, $\lambda_1 = \lambda_2 = 1e4$ for DeiT-S and DeiT-B, and $\lambda_1 = \lambda_2 = 1e4$ for Swin. For detection and segmentation tasks, we set $\lambda_1 = \lambda_2 = 1e5$ for all models. All experiments are implemented using PyTorch framework (Paszke et al., 2019) with a single NVIDIA 3090 GPU and an Intel Xeon 4214R CPU.

Compared methods. We re-implement BRECQ (Li et al., 2021), QDrop (Wei et al., 2022), PD-Quant (Liu et al., 2023a), GPTQ (Frantar et al., 2022) with their official code with 32 images as the calibration dataset as the same ours. The initial implementation of GPTQ did not involve activation quantization. Thus, we employed the same quantizer as our own to quantize the activation for them, including the reparameterization technique and the $\log\sqrt{2}$ quantizer. For other PTQ of ViT methods including PSAQ-ViT (Li et al., 2022b), Ranking (Liu et al., 2021b), EasyQuant (Wu et al., 2020), PTQ4ViT (Yuan et al., 2022), APQ-ViT (Ding et al., 2022), NoisyQuant (Liu et al., 2023b), RepQ-ViT (Li et al., 2023), we use the result reported in their paper if it exists, otherwise, we re-implement based on their official code. The ablation study of image numbers and comparisons of time costs are presented in the appendix.

5.2. Results on ImageNet Dataset

The comparison between ERQ and other PTQ of ViTs methods is presented in Tab. 2. It can be seen that the proposed ERQ showcases advantages over the compared methods in all bit-width settings, especially in the low-bit cases. Specifically, due to the small amount of the calibration dataset, many methods typically suffer from the overfitting problem and exhibit unstable performance. For instance, for the W3A4 case, QDrop and PD-Quant obtain 9.77% and 4.56% on ViT-S, respectively. In contrast, the proposed ERQ shows stable improvements across all variants. Notably, ERQ demonstrates 22.36% and 9.25% performance gains on ViT-S and ViT-B, 1.84%, 8.68% and 8.58% gains on DeiT-T, DeiT-S, and DeiT-B, 1.61% and 1.45% gains on Swin-S and Swin-B. When it comes to the W4A4 case, ERQ respectively obtains 1.32% and 1.51% performance gains on ViT-S and ViT-B, 1.33%, 1.71% and 2.13% performance gains on DeiT-T, DeiT-S, and DeiT-B, 0.57% and 1.36% performance gains on Swin-S and Swin-B. For the W5A5 case, ERQ also presents the best performance. For example, ERQ improves the accuracy by 1.28% on Swin-B.

5.3. Results on COCO Dataset

The results of object detection and instance segmentation are reported in Tab. 3. It can be seen that ERQ improves performance in most cases. For instance, ERQ augments the box AP and mask AP by 0.5 and 0.3 for W4A4 Mask R-CNN with Swin-T as its backbone, respectively. Also, ERQ augments the box AP and mask AP by 0.8 and 0.6 for W4A4 Cascade Mask R-CNN with Swin-T as its backbone. The above results further demonstrate the effectiveness and generalization ability of the proposed ERQ.

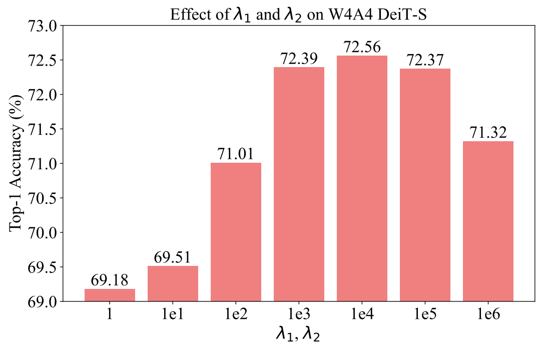

Figure 4. Ablation studies of λ_1 and λ_2.

5.4. Ablation Study

All ablation studies are conducted on the W4A4 DeiT-S. Tab. 4 reports the results of various components within the ERQ. Note that the Wqer consists of Rounding Refinement and Ridge Regression. It can be observed that, compared

Table 3. Results on COCO dataset. "AP^{box}" denotes the box average precision for object detection, and "AP^{mask}" denotes the mask average precision for instance segmentation. "*" and "†" indicate the results are re-produced by using the official code.

Method	W/A	Mask R-CNN				Cascade Mask R-CNN			
		w. Swin-T		w. Swin-S		w. Swin-T		w. Swin-S	
		AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-Precision	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT (Yuan et al., 2022)	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT (Ding et al., 2022)	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
GPTQ* (Frantar et al., 2022)	4/4	36.3	36.3	42.9	40.2	47.1	41.5	49.2	43.2
RepQ-ViT (Li et al., 2023)	4/4	36.1	36.0	44.2$_{42.7}$†	40.2$_{40.1}$†	47.0	41.4	49.3	43.1
AdaRound* (Nagel et al., 2020a)	4/4	16.3	19.8	22.3	22.5	34.6	33.4	35.8	34.5
BRECQ* (Li et al., 2021)	4/4	25.2	27.3	32.4	32.9	40.4	35.9	41.5	37.2
QDrop* (Wei et al., 2022)	4/4	10.4	11.3	39.7	37.8	17.9	16.2	20.1	17.4
PD-Quant* (Liu et al., 2023a)	4/4	15.7	16.1	30.2	28.4	34.5	30.1	38.6	34.1
ERQ (Ours)	4/4	**36.8**	**36.6**	43.4	**40.7**	**47.9**	**42.1**	**50.0**	**43.6**

Table 4. Ablations on different components of ERQ. "baseline" indicates only performing calibration and no error reduction is involved. "Aqer" and "Wqer" represent Activation quantization error reduction and Weight quantization error reduction, respectively. "Rounding" and "Ridge" represent Rounding Refinement and Ridge Regression used in Wqer, respectively. Results are reported with W4A4 DeiT-S on ImageNet.

Aqer	Wqer		Top-1 Acc. (%)
	Rounding	Ridge	
Baseline			68.41
✓			71.45 (+3.04)
	✓		69.24 (+0.83)
		✓	70.06 (+1.65)
	✓	✓	70.49 (+2.08)
✓	✓		71.83 (+3.42)
✓		✓	72.01 (+3.60)
✓	✓	✓	**72.56 (+4.15)**

to the baseline, Aqer enhances accuracy by 3.04%. Furthermore, Rounding Refinement and Ridge Regression results in accuracy improvements of 0.83% and 1.65%, respectively. When these two approaches are both employed, i.e., using Wqer, the performance is improved by 2.08%. Ultimately, the combination of Aqer with Wqer showcases the optimal performance, with an accuracy increment of 4.15% points to 72.56% over the baseline. These results confirm the effectiveness of the components in ERQ. Then, we provide the ablation study of λ_1 of Eq. 6 and λ_2 of Eq. 16. For simplicity, we set $\lambda_1 = \lambda_2$ and search for the best value. Despite this may not be the best choice, it yields desirable performance. Fig. 4 presents the results of different values. It can be seen that the best performance is exhibited when $\lambda_1 = \lambda_2 = 1e4$ for W4A4 Deit-S. Tab. 5 presents the ablation study of different k in Eq. 15 of Rounding Refinement. It can be observed that when $k = 1$, the best accuracy is achieved. Note that when $k = 0$, the Rounding Refinement is invalid. Finally, in Sec. D of the appendix, we demonstrate that each component of ERQ including Aqer, Rounding Refinement of Wqer, and Ridge Regression of Wqer effectively reduces the quantization error.

Table 5. Ablation studies of different k in Rounding Refinement.

Model	k	Top-1 Acc. (%)
DeiT-S (W4/A4)	0	72.01
	1	**72.56**
	2	72.38
	3	71.79

6. Discussion

We further discuss several unexplored limitations of the proposed ERQ, which will be our future focus. First, despite achieving considerable performance gains, ERQ currently focuses on layers with weight. A further improvement would be feasible if the error of quantized self-attention can be taken into consideration. Meanwhile, the Rounding Refinement remains further exploration. There is substantial potential for exploring other refinement techniques such as offering more flexibility in rounding targets. Finally, limited by the computational resources, we are currently unable to apply ERQ to Large Language Models (LLMs). Extending ERQ to LLMs is an imperative task for our future studies.

7. Conclusion

In this paper, we present ERQ, consisting of Activation quantization error reduction (Aqer) and Weight quantization error reduction (Wqer) to respectively mitigate the quantization error induced by activation and weight quantization. In Aqer, the mitigation of activation quantization error is formulated as a Ridge Regression problem, presenting a closed-form solution to tackle the error by updating weights with full-precision. In Wqer, the weight quantization error is progressively mitigated in a quantization-and-correction manner. At each iteration, the first half of weights are quantized and the resulting error is first mitigated by Rounding Refinement and then again by solving a Ridge Regression problem. The former first mitigates the quantization error by leveraging an empirically derived efficient proxy of output error to refine the rounding directions of quantized weights. The latter further mitigates the quantization error into the remaining full-precision weight with a closed-form solution. The effectiveness of ERQ is demonstrated by extensive experiments on a variety of ViTs across diverse tasks.

Acknowledgements

This work was supported by National Key R&D Program of China (No.2022ZD0118202), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001).

Impact Statement

This paper presents a PTQ method that aims to boost the performance of quantized ViTs. The model compression community may benefit from this research. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 6836–6846, 2021.

Banner, R., Nahshan, Y., Soudry, D., et al. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7950–7958, 2019.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6154–6162, 2018.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229. Springer, 2020.

Chmiel, B., Banner, R., Shomron, G., Nahshan, Y., Bronstein, A., Weiser, U., et al. Robust quantization: One model to rule them all. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5308–5317, 2020.

Ding, R., Chin, T.-W., Liu, Z., and Marculescu, D. Regularizing activation distribution for training binarized deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11408–11417, 2019.

Ding, Y., Qin, H., Yan, Q., Chai, Z., Liu, J., Wei, X., and Liu, X. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia (ACMMM)*, pp. 5380–5388, 2022.

Dong, P., Lu, L., Wu, C., Lyu, C., Yuan, G., Tang, H., and Wang, Y. Packqvit: Faster sub-8-bit vision transformers via full and packed quantization on the mobile. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

Frantar, E. and Alistarh, D. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 4475–4488, 2022.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Frumkin, N., Gope, D., and Marculescu, D. Jumping through local minima: Quantization in the loss landscape of vision transformers. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision (ICCV), pp. 16978–16988, 2023.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2961–2969, 2017.

Hou, Z. and Kung, S.-Y. Multi-dimensional vision transformer compression via dependency guided gaussian process search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3669–3678, 2022.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

Kuzmin, A., Nagel, M., van Baalen, M., Behboodi, A., and Blankevoort, T. Pruning vs quantization: Which is better?, 2023.

Li, Y., Dong, X., and Wang, W. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. Brecq: Pushing the limit of post-training quantization by block reconstruction. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Li, Y., Xu, S., Zhang, B., Cao, X., Gao, P., and Guo, G. Q-vit: Accurate and fully quantized low-bit vision transformer. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 34451–34463, 2022a.

Li, Z. and 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, 2023.

Li, Z., Ma, L., Chen, M., Xiao, J., and Gu, Q. Patch similarity aware data-free quantization for vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–170. Springer, 2022b.

Li, Z., Xiao, J., Yang, L., and Gu, Q. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17227–17236, 2023.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 1833–1844, 2021.

Lin, C., Peng, B., Li, Z., Tan, W., Ren, Y., Xiao, J., and Pu, S. Bit-shrinking: Limiting instantaneous sharpness for improving post-training quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16196–16205, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014.

Lin, Y., Zhang, T., Sun, P., Li, Z., and Zhou, S. Fq-vit: Post-training quantization for fully quantized vision transformer. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 1173–1179, 2022.

Liu, J., Niu, L., Yuan, Z., Yang, D., Wang, X., and Liu, W. Pd-quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24427–24437, 2023a.

Liu, Y., Yang, H., Dong, Z., Keutzer, K., Du, L., and Zhang, S. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20321–20330, 2023b.

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

Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., and Gao, W. Post-training quantization for vision transformer. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 28092–28103, 2021b.

Mehta, S. and Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 7197–7206, 2020a.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 7197–7206. PMLR, 2020b.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8026–8037, 2019.

Pia, A. D., Dey, S. S., and Molinaro, M. Mixed-integer quadratic programming is in np. *Mathematical Programming*, 162:225–240, 2017.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115:211–252, 2015.

Shamshad, F., Khan, S., Zamir, S. W., Khan, M. H., Hayat, M., Khan, F. S., and Fu, H. Transformers in medical imaging: A survey. *Medical Image Analysis*, pp. 102802, 2023.

Sun, Z., Ge, C., Wang, J., Lin, M., Chen, H., Li, H., and Sun, X. Entropy-driven mixed-precision quantization for deep network design. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 21508–21520, 2022.

Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., and Tao, D. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12165–12174, 2022.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10347–10357. PMLR, 2021.

Van Baalen, M., Louizos, C., Nagel, M., Amjad, R. A., Wang, Y., Blankevoort, T., and Welling, M. Bayesian bits: Unifying quantization and pruning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5741–5752, 2020.

Wei, X., Gong, R., Li, Y., Liu, X., and Yu, F. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Wu, D., Tang, Q., Zhao, Y., Zhang, M., Fu, Y., and Zhang, D. Easyquant: Post-training quantization via scale optimization. *CoRR*, abs/2006.16669, 2020.

Xijie Huang, Z. S. and Cheng, K.-T. Variation-aware vision transformer quantization. *arXiv preprint arXiv:2307.00331*, 2023.

Yuan, Z., Xue, C., Chen, Y., Wu, Q., and Sun, G. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 191–207. Springer, 2022.

Zhang, J., Peng, H., Wu, K., Liu, M., Xiao, B., Fu, J., and Yuan, L. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12145–12154, 2022.

Zheng, D., Dong, W., Hu, H., Chen, X., and Wang, Y. Less is more: Focus attention for efficient detr. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6674–6683, 2023.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 6881–6890, 2021.

Zhou, A., Yao, A., Guo, Y., Xu, L., and Chen, Y. Incremental network quantization: Towards lossless cnns with low-precision weights. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2017.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A. More Results on ImageNet Dataset

Tab. 6 presents the comparisons between the proposed ERQ and other PTQ methods. It can be observed that for the W6A6 case, ERQ also exhibits the best results by further improving the performance. For example, ERQ only incurs 0.39% and 0.25% drops compared to the full-precision model for DeiT-s and Swin-B, respectively.

Table 6. More results on ImageNet dataset. The top-1 accuracy (%) is reported as the metric. "W/A" indicates that the bit-width of the weight and activation are W and A bits, respectively. †/‡ respectively indicates applying NoisyQuant to linear quantization/PTQ4ViT. "*" indicates the results are re-produced by using the official code.

Method	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT (Lin et al., 2022)	6/6	4.26	0.10	58.66	45.51	64.63	66.50	52.09
PSAQ-ViT (Li et al., 2022b)	6/6	37.19	41.52	57.58	63.61	67.95	72.86	76.44
Ranking (Liu et al., 2021b)	6/6	-	75.26	-	74.58	77.02	-	-
PTQ4ViT (Yuan et al., 2022)	6/6	78.63	81.65	69.68	76.28	80.25	82.38	84.01
APQ-ViT (Ding et al., 2022)	6/6	79.10	82.21	70.49	77.76	80.42	82.67	84.18
NoisyQuant† (Liu et al., 2023b)	6/6	76.86	81.90	-	76.37	79.77	82.78	84.57
NoisyQuant‡ (Liu et al., 2023b)	6/6	78.65	82.32	-	77.43	80.70	82.86	84.68
GPTQ* (Frantar et al., 2022)	6/6	80.44	83.72	71.05	78.95	81.37	82.82	84.89
RepQ-ViT (Li et al., 2023)	6/6	80.43	83.62	70.76	78.90	81.27	82.79	84.57
EasyQuant (Wu et al., 2020)	6/6	75.13	81.42	-	75.27	79.47	82.45	84.30
Bit-shrinking (Lin et al., 2023)	6/6	80.44	83.16	-	78.51	80.47	82.44	-
BRECQ* (Li et al., 2021)	6/6	61.18	71.29	69.62	70.93	79.46	81.85	84.08
QDrop* (Wei et al., 2022)	6/6	68.57	74.38	69.98	76.57	80.66	82.53	84.31
PD-Quant* (Liu et al., 2023a)	6/6	71.38	63.14	70.74	77.63	79.32	82.33	84.38
ERQ (Ours)	6/6	**80.48**	**83.89**	**71.14**	**79.03**	**81.41**	**82.86**	**85.02**

B. Ablation Study of Image Numbers

Tab. 7 presents the ablation study using different image numbers. It can be observed that as the image number increases, the performance increases correspondingly. For example, the accuracy is 71.58% and 72.56% for 4 and 32 images, respectively. After 256 images, the performance reaches the plateau. Despite using more images can improve the performance, in our main paper, we adopt 32 images to align with the previous study (Li et al., 2023) for a fair comparison.

Table 7. Ablation studies of different image numbers.

Model	Image Numbers	Top-1 Acc. (%)
DeiT-S (W4/A4)	4	71.58
	8	71.87
	16	72.54
	32	72.56
	64	72.94
	128	73.19
	256	73.51
	512	73.68
	1024	73.69

Table 8. Time costs of different methods. "*" indicates the results are re-produced by using the official code. Experiments are conducted with W4A4 DeiT-S.

Method	Runtime	Top-1 Acc. (%)
BRECQ* (Li et al., 2021)	∼48 minutes	32.89
QDrop* (Wei et al., 2022)	∼80 minutes	35.79
PD-Quant* (Liu et al., 2023a)	∼110 minutes	64.85
GPTQ* (Frantar et al., 2022)	∼3 minutes	70.85
RepQ-ViT (Li et al., 2023)	∼1 minute	69.03
ERQ (Ours)	∼4 minutes	**72.56**

C. Comparisons of Time Costs

Tab. 8 presents comparisons of time costs between the proposed ERQ and other PTQ methods. It can be seen that the BRECQ, QDrop, and PD-Quant require longer time overhead. In contrast, GPTQ, RepQ-ViT, and the proposed ERQ demonstrated significantly reduced time costs. Notably, our ERQ achieved the best Top-1 Accuracy of 72.56% with a runtime of only 4 minutes.

D. Validation of Error Reduction

In the main paper, we demonstrate that the proposed ERQ improves performance. In this section, we validate that each component of ERQ successfully reduces the quantization error, making the output of quantized computation closer to the full-precision output.

D.1. Activation quantization error reduction (Aqer)

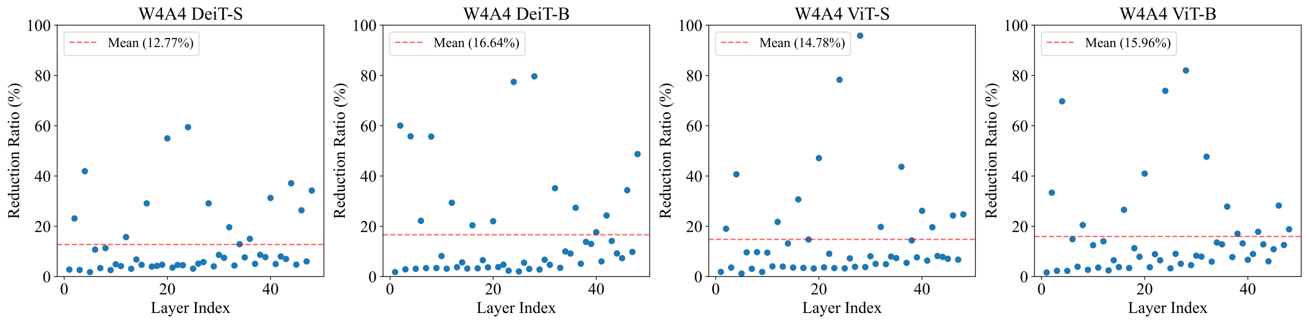

Figure 5. Illustration of the error reduction ratio brought by Aqer. We plot the error reduction ratio for each layer.

Fig. 5 presents the error reduction ratio brought by Aqer. The ratio is computed using the value of Eq. 4 before and after applying Aqer. As can be observed, Aqer generally yields a 13%-17% average error reduction. Therefore, applying Aqer improves the performance as shown in the result in Tab. 4 of the main paper.

D.2. Weight quantization error reduction (Wqer)

The proposed Wqer consists of Rounding Refinement and Ridge Regression. In the next, we respectively plot the error reduction ratio of applying Rounding Refinement, applying Ridge Regression, and applying both Rounding Refinement and Ridge Regression, i.e., applying Wqer.

D.2.1. Rounding Refinement

Fig. 6 presents the error reduction ratio brought by Rounding Refinement. The ratio is computed using the value of Eq. 4 before and after applying Rounding Refinement. As can be observed, Rounding Refinement generally yields a 7%-11% average error reduction. Such results support that applying Rounding Refinement improves the performance as shown in the

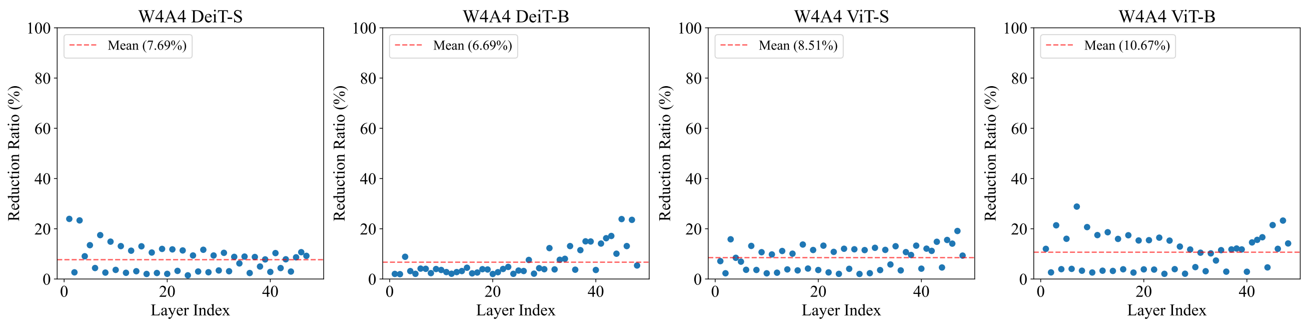

Figure 6. Illustration of the error reduction ratio brought by Rounding Refinement. We plot the error reduction ratio for each layer.

result in Tab. 4 of the main paper.

D.2.2. Ridge Regression

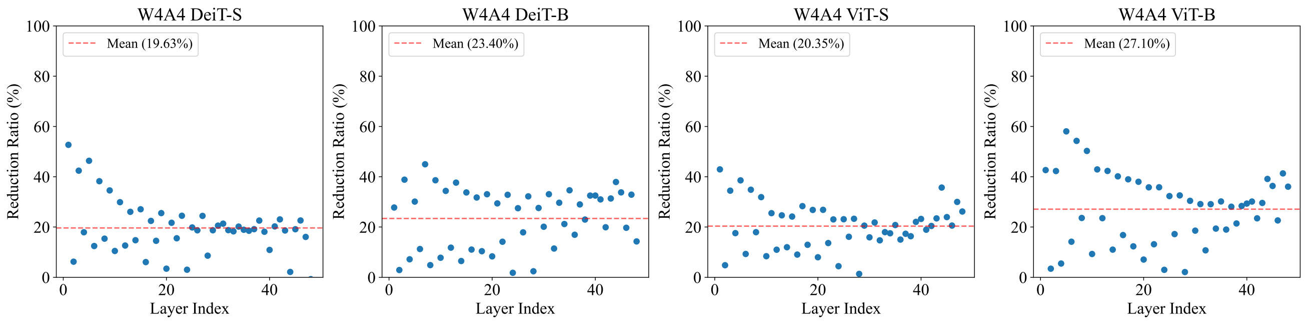

Figure 7. Illustration of the error reduction ratio brought by Ridge Regression. We plot the error reduction ratio for each layer.

Fig. 7 presents the error reduction ratio brought by the Ridge Regression of Wqer. The ratio is computed using the value of Eq. 4 before and after applying Ridge Regression. As can be observed, Rounding Refinement generally yields a 20%-27% average error reduction. Thus, applying Ridge Regression improves the performance as shown in the result in Tab. 4 of the main paper.

D.2.3. Wqer (Rounding Refinement + Ridge Regression

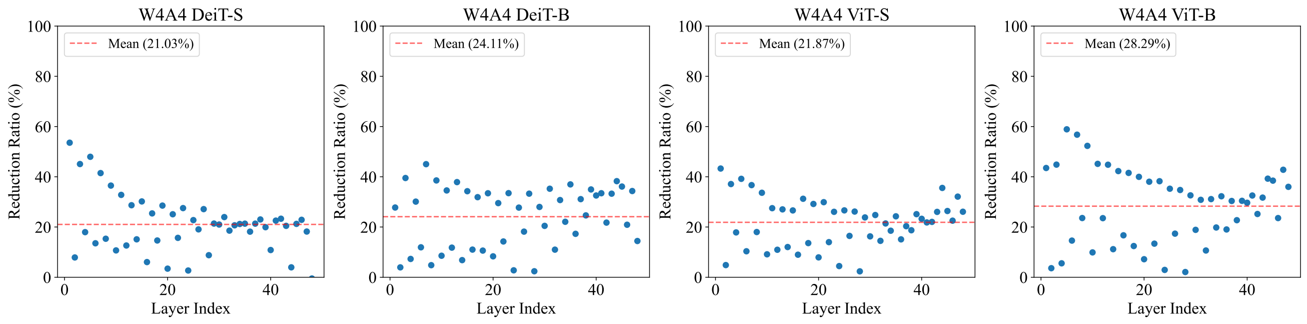

Figure 8. Illustration of the error reduction ratio brought by Wqer. We plot the error reduction ratio for each layer.

When both Rounding Refinement and Ridge Regression are applied, it is equal to applying Wqer. Fig. 8 presents the error reduction ratio brought by Wqer. The ratio is computed using the value of Eq. 4 before and after Wqer. As can be observed, Wqer generally yields a 21%-28% average error reduction. Note that compared to the ratio of applying Ridge Regression, the reduction ratio further increases. Specifically, the reduction ratio increases by 0.40%, 0.71%, 1.52%, and 1.19% for W4A4

DeiT-S, DeiT-B, ViT-S, and ViT-B, respectively. Such results demonstrate that the combination of Rounding Refinement and Ridge Regression brings further error reduction, thereby resulting in a better performance, which is also supported by the result in Tab. 4 of the main paper.

D.3. ERQ (Aqer + Wqer)

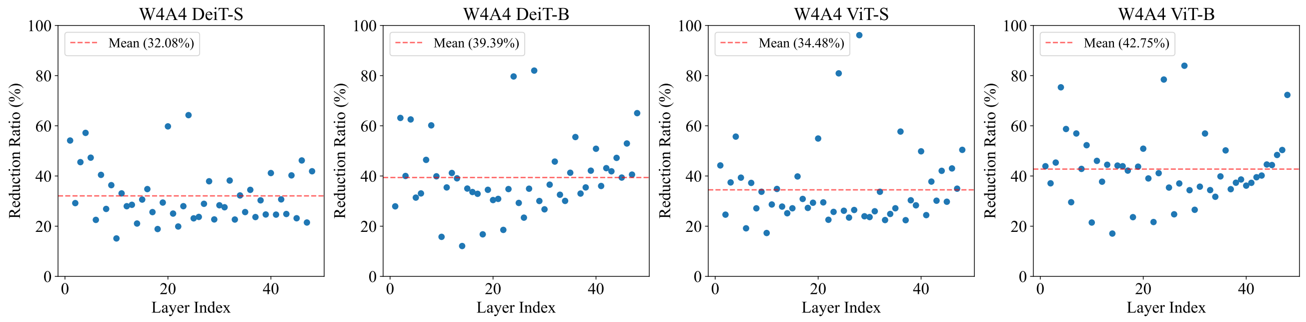

Figure 9. Illustration of the error reduction ratio brought by ERQ (Aqer + Wqer). We plot the error reduction ratio for each layer.

In this subsection, we provide the error reduction ratio when both Aqer and Wqer are applied, i.e., our ERQ. Fig. 9 presents the error reduction ratio brought by Wqer. The reduction ratio is computed using the value of Eq. 4. It can be observed that combining ERQ yields a 32%-47% average error reduction. Meanwhile, the results prove that combining Aqer and Wqer presents a higher error reduction ratio, supporting the performance results as shown in Tab. 4 of the main paper.

In conclusion, the aforementioned results prove the effectiveness of the proposed ERQ in reducing quantization error, which narrows the difference between the output of the quantized layer and that of the full-precision layer. These findings also justify the two-step approach of ERQ, which addresses activation and weight quantization errors sequentially.

E. Comparisons of Time Costs

Table 9. Model latency and throughput of W8A8 ViTs.

Model	Latency (ms)	Throughput (img/s)
ViT-S	184 104 (1.77x)	5.43 9.62 (1.77x)
ViT-B	746 443 (1.68x)	1.34 2.26 (1.68x)
DeiT-T	54 31 (1.74x)	18.51 32.26 (1.74x)
DeiT-S	163 106 (1.54x)	6.13 9.43 (1.54x)
DeiT-B	745 376 (1.98x)	1.34 2.66 (1.98x)
Swin-S	337 217 (1.55x)	2.97 4.61 (1.55x)
Swin-B	683 461 (1.48x)	1.46 2.17 (1.48x)

Tab. 9 presents the latency and throughput of W8A8 ViT-S, DeiT-S, and Swin-S. The experiments are conducted with onnx framework on Intel i5-10210U CPU. The thread number is set to 1 and the results are evaluated by forwarding a single 224*224 image. We repeat the process 5 times and report the average outcomes. It can be seen that the quantized model typically achieves 1.5x to 2x speedups, demonstrating the effectiveness of quantization. Reducing the bit-width further has the potential to yield greater speedup. For example, the W4A4 case could achieve 3x to 6x speedups (Dong et al., 2024).

However, the implementation of bit-widths below 8-bit typically requires specialized hardware (Dong et al., 2024; Li et al., 2021), and is not supported by the public software framework. At this point, we lack the necessary toolset to reproduce this. Nonetheless, we will study this as long as the toolset is available in the future.

It is worth noting that the results in W8A8 prove the effectiveness of our ERQ in facilitating acceleration. Thus, it can be expected that our ERQ is also able to achieve similar acceleration in the lower bit-width cases if supported by the necessary hardware and software framework.

Table 10. CSize and FLOPs for different bit-width configurations.

Model	Method	W/A	Size (MB)	FLOPs (G)
DeiT-S	Baseline	FP	88	4.6
	ERQ	3/4	8.3	0.054
	ERQ	4/4	11	0.072
	ERQ	5/5	13.8	0.11

F. Analysis on model size and computational costs

For the quantized model, ERQ has the same model size and computational costs as the other PTQ methods since the quantized models have the same bit-width. In Tab. 10, we present the model size and the FLOPs of DeiT-S as the example. Here, the FLOPs are converted from Bit Operations (BOPs) (Van Baalen et al., 2020).

Table 11. Performance comparison of W4A4 and W4A8 ViTs on ImageNet Dataset.

Method	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
ERQ	4/4	68.91	76.63	60.29	72.56	78.23	80.74	82.44
ERQ	4/8	77.84	81.98	68.31	77.53	80.22	82.24	84.23

G. Analysis on activation bit-width

Table 12. Performance comparison of W4A4 and W4A8 ViTs on COCO Dataset.

Model	Method	W/A	AP(box)	AP(mask)
Mask R-CNN (Swin-T)	Full-Precision	32/32	46.0	41.6
	ERQ	4/4	36.8	36.6
	ERQ	4/8	41.0	39.2
Mask R-CNN (Swin-S)	Full-Precision	32/32	48.5	43.3
	ERQ	4/4	43.4	40.7
	ERQ	4/8	46.1	42.2
Cascade R-CNN (Swin-T)	Full-Precision	32/32	50.4	43.7
	ERQ	4/4	47.9	42.2
	ERQ	4/8	49.5	43.3
Cascade R-CNN (Swin-S)	Full-Precision	32/32	51.9	45.0
	ERQ	4/4	50.0	43.6
	ERQ	4/8	51.3	44.5

Tab. 11 presents the performance comparison of W4A4 and W4A8 ViTs on ImageNet Dataset. It can be seen that using 8-bit activation leads to considerable gains for ViT variants. Specifically, the gains are respectively 8.93% and 5.35% on ViT-S and ViT-B, 8.02%, 4.97%, and 1.99% on DeiT-T, DeiT-S, and DeiT-B, 1.50% and 1.79% on Swin-S and Swin-B. Tab. 12

presents the performance comparison of W4A4 and W4A8 ViTs on the detection task. For the detection task, using 8-bit activation also yields significant improvements. For example, the AP(box) and AP(mask) are respectively improved by 4.2 and 2.6 for Mask R-CNN (Swin-T).