RepQ-ViT: Scale Reparameterization for Post-Training Quantization of Vision Transformers

Zhikai Li[1,2], Junrui Xiao[1,2], Lianwei Yang[1,2], and Qingyi Gu[1,*]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
{lizhikai2020, xiaojunrui2020, yanglianwei2021, qingyi.gu}@ia.ac.cn

Abstract

Post-training quantization (PTQ), which only requires a tiny dataset for calibration without end-to-end retraining, is a light and practical model compression technique. Recently, several PTQ schemes for vision transformers (ViTs) have been presented; unfortunately, they typically suffer from non-trivial accuracy degradation, especially in low-bit cases. In this paper, we propose RepQ-ViT, a novel PTQ framework for ViTs based on quantization scale reparameterization, to address the above issues. RepQ-ViT decouples the quantization and inference processes, where the former employs complex quantizers and the latter employs scale-reparameterized simplified quantizers. This ensures both accurate quantization and efficient inference, which distinguishes it from existing approaches that sacrifice quantization performance to meet the target hardware. More specifically, we focus on two components with extreme distributions: post-LayerNorm activations with severe inter-channel variation and post-Softmax activations with power-law features, and initially apply channel-wise quantization and log$\sqrt{2}$ quantization, respectively. Then, we reparameterize the scales to hardware-friendly layer-wise quantization and log2 quantization for inference, with only slight accuracy or computational costs. Extensive experiments are conducted on multiple vision tasks with different model variants, proving that RepQ-ViT, without hyperparameters and expensive reconstruction procedures, can outperform existing strong baselines and encouragingly improve the accuracy of 4-bit PTQ of ViTs to a usable level. Code is available at `https://github.com/zkkli/RepQ-ViT`.

1. Introduction

With the powerful representational capabilities of the self-attention mechanism, vision transformers (ViTs) have recently demonstrated surprising potential in a range of vision applications, including image classification [5, 23], object detection [2, 38], semantic segmentation [28], etc., and are thus being widely investigated as new vision backbones [8]. However, ViTs rely on heavy and intensive computations, resulting in intolerable memory footprint, power consumption, and inference latency, which hinders their deployment on resource-constrained edge devices [10, 29]. Consequently, compression techniques for ViTs are essential in real-world applications, particularly where low-cost deployment and real-time inference are desired.

Model quantization, which reduces model complexity by decreasing the representation precision of weights and activations, is an effective and prevalent compression approach [7, 12]. A notable research line is based on quantization-aware training (QAT) [3, 6], which relies on end-to-end retraining to compensate for the accuracy of the quantized model. Despite the good performance, such retraining requires gradient backpropagation and parameter updates on the entire training dataset, which brings undesirably large time and resource costs [20, 33]. Fortunately, another family of methods, referred to as post-training quantization (PTQ), can overcome the above challenges [32, 16, 26]. It simply takes a tiny unlabeled dataset to calibrate the quantization parameters without retraining and thus is regarded as a promising and practical solution.

Although various PTQ methods for convolutional neural networks (CNNs) have been proposed in previous works with good performance, they produce disappointing results on ViTs, with more than 1% accuracy drop even in 8-bit quantization [35]. To this end, several efforts identify the key components that limit the quantization performance of ViTs, such as LayerNorm, Softmax, and GELU, and propose PTQ schemes accordingly [22, 24]. Nevertheless, when performing ultra-low-bit (*e.g.*, 4-bit) quantization, the performance of these schemes is still far from satisfactory [4]. The core reason for their low performance is that they invariably follow the traditional quantization paradigm, in which the initial design of the quantizers must account for the future inference overhead. This forces previous meth-

[*] Corresponding author.

ods to carefully design simple quantizers to accommodate the characteristics of the target hardware, even at the cost of remarkably sacrificing accuracy.

Is the traditional quantization-inference dependency paradigm the only option? To answer this question, we explore the feasibility of decoupling the quantization and inference processes, and reveal that complex quantizers and hardware standards are not always antagonistic; instead, the two can be explicitly bridged via *scale reparameterization*. This potentially derives an interesting quantization-inference decoupling paradigm, in which complex quantizers are employed in the initial quantization to adequately preserve the original parameter distributions, and then they are transformed to simple hardware-friendly quantizers via scale reparameterization for actual inference, resulting in both high quantization accuracy and inference efficiency.

With the above insights, we propose a novel PTQ framework for ViTs, called RepQ-ViT, in this paper. In RepQ-ViT, we focus on two components with extreme distributions in ViTs that challenge the direct use of simple quantizers. Specifically, for post-LayerNorm activations, we initially apply channel-wise quantization to maintain their severe inter-channel variation, and then reparameterize the scales to layer-wise quantization to match the hardware, which is achieved by adjusting the LayerNorm's affine factors and the next layer's weights; for post-Softmax activations, since our study shows that their power-law distributions and the properties of attention scores prefer $\log\sqrt{2}$ quantizers, we are interested in reparameterizing the scales to change the base to 2 to enable bit-shifting operations in inference. The overview of the RepQ-ViT framework is illustrated in Figure 1. Note that the scale reparameterization methods presented in this paper enjoy theoretical support, with only a slight accuracy drop compared to complex quantizers or a slight computational overhead compared to simple quantizers, and thus have the potential to ensure interpretability and robustness.

Our main contributions are summarized as follows:

- We propose a novel PTQ framework for ViTs that escapes from the traditional paradigm by decoupling the quantization and inference processes, with the former employing complex quantizers and the latter employing scale-reparameterized simplified quantizers, which has great potential in quantizing components with extreme distributions in ViTs.

- For post-LayerNorm and post-Softmax activations, we initially apply channel-wise and $\log\sqrt{2}$ quantization, respectively, to maintain the original data distribution, and then transform them to simple quantizers via interpretable scale reparameterization to match the hardware in inference.

- We evaluate RepQ-ViT on various vision tasks, including image classification, object detection, and instance segmentation, and RepQ-ViT, without hyperparameters and expensive reconstruction procedures, can encouragingly outperform existing baselines.

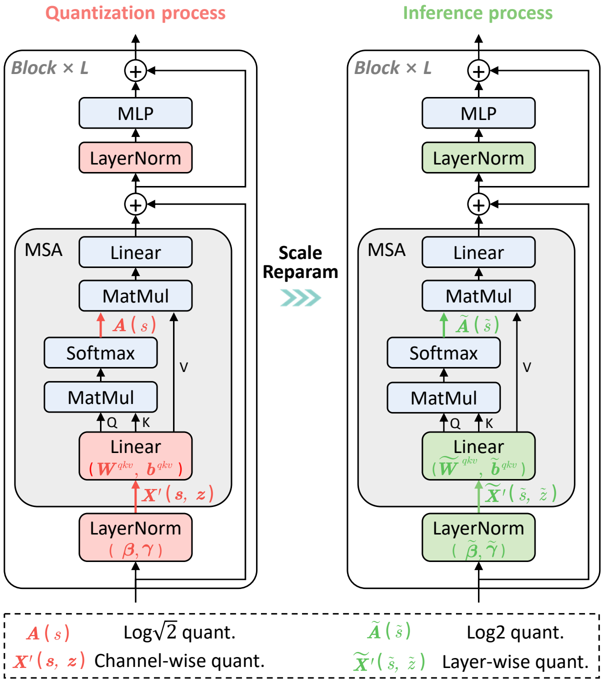

Figure 1. Overview of the RepQ-ViT framework. Building on the quantization-inference decoupling paradigm, for post-LayerNorm and post-Softmax activations, complex quantizers are employed in the quantization process and simple quantizers are employed in the inference process, with scale reparameterization bridging the two.

2. Related Works

2.1. Vision Transformers

ViTs, which exploit the self-attention mechanism to extract global information, have recently achieved excellent performance on a variety of computer vision tasks, showing great potential as general-purpose vision backbones [8]. ViT [5] attempts to remove all convolutions and apply a pure transformer-based model to the image classification task for the first time and achieves competitive results. Afterwards, several variants are proposed to further improve the performance. DeiT [30] introduces an efficient training strategy to reduce the dependency on large-scale training data, and Swin [23] applies a hierarchical architecture with shifted windows to enhance the modeling power of the self-attention mechanism. In addition, ViTs have also been successfully applied to high-level applications, such as object detection [2, 38] and semantic segmentation [28].

Despite the promising performance, the massive large matrix multiplications in ViTs incur huge memory footprints and computational overheads in real-world applications, which are intolerable in resource-constrained edge scenarios [29, 10]. Several works, such as MobileViT [25] and MiniViT [36], attempt to address the above issues through lightweight architecture design, while they still provide substantial room for further compression as they keep floating-point parameters.

2.2. Model Quantization

Model quantization, which represents the original floating-point parameters with low-bit values, is an effective approach for compressing neural networks [7, 12]. To achieve competitive quantization performance, lots of methods follow the QAT pipeline and perform retraining on the entire training dataset [3, 6, 31, 37]; however, such retraining is resource-intensive and time-consuming. Thus, PTQ, which is free from retraining, is believed to be a more promising solution for low-cost and rapid deployment. Several impressive PTQ methods have been proposed with great success on CNNs, such as DFQ [27], AdaRound [26], and BRECQ [16], yet they have poor performance on ViTs with substantially different structures.

As a result, designing PTQ methods for ViTs has recently received widespread interest. Ranking loss [24] is utilized to maintain the relative order of attention scores before and after quantization. FQ-ViT [22] introduces Powers-of-Two Scale and Log-Int-Softmax to quantize LayerNorm and Softmax operations to obtain fully quantized ViTs. PSAQ-ViT [19, 17] designs a relative value metric to invert images and pushes PTQ for ViTs to data-free scenarios. PTQ4ViT [35] presents twin uniform quantization to cope with the unbalanced distributions of post-Softmax and post-GELU activations and uses a Hessian-guided metric to search for quantization scales. APQ-ViT [4] works on preserving the Matthew effect of post-Softmax activations and proposes a calibration scheme that perceives the overall quantization disturbance in a block-wise manner. Unfortunately, the above methods produce non-trivial accuracy drops or even crashes in ultra-low-bit quantization. The main performance bottleneck stems from their direct use of simple hardware-oriented quantizers that cannot represent the extreme distributions well; in contrast, the novel paradigm proposed in this paper can potentially eliminate these issues.

3. Methodology

Overview Figure 1 illustrates the overview of the proposed RepQ-ViT framework. In the quantization-inference decoupling paradigm, the main challenge is to convert the initial complex quantizers to the simple quantizers for inference. Thus, we propose scale reparameterization methods for post-LayerNorm and post-Softmax activations, respectively, as detailed in Sections 3.2 and 3.3. Moreover, their flows are described in Algorithm 1.

Algorithm 1 Pipeline of RepQ-ViT framework.

```
Input: Pretrained full-precision model, Calib data
Initialize the quantized model with calib data and Eq. 9, where post-LayerNorm activations $\boldsymbol{X}'$ apply channel-wise quantization $(\boldsymbol{s}, \boldsymbol{z})$ and post-Softmax activations $\boldsymbol{A}$ apply $\log\sqrt{2}$ quantization $(s)$;
    # Scale reparam for post-LayerNorm activations
Update the quantizer of $\boldsymbol{X}'$ via $\tilde{s} = \mathrm{E}[\boldsymbol{s}]$ and $\tilde{z} = \mathrm{E}[\boldsymbol{z}]$;
Calculate $\boldsymbol{r}_1 = \boldsymbol{s}/(\tilde{s} \cdot \mathbf{1})$ and $\boldsymbol{r}_2 = \boldsymbol{z} - (\tilde{z} \cdot \mathbf{1})$;
Update LayerNorm's affine factors $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\boldsymbol{\gamma}}$ based on Eq. 15;
Update next layer's weights $\widetilde{\boldsymbol{W}}^{qkv}$ and $\widetilde{\boldsymbol{b}}^{qkv}$ based on Eq. 17;
Re-calibrate $\widetilde{\boldsymbol{W}}^{qkv}$ with calib data;
    # Scale reparam for post-Softmax activations
Update quantization procedure based on Eq. 18;
Update $\tilde{s}$ in de-quantization procedure based on Eq. 20;
Output: Quantized model
```

3.1. Preliminaries

ViTs' standard structure First, the input image is reshaped into N flatted 2D patches, and they are subsequently projected by the embedding layer to a D-dimensional vector sequence, which is denoted as $\boldsymbol{X}_0 \in \mathbb{R}^{N\times D}$ [1]. Then, $\boldsymbol{X}_0$ is fed into a stack of transformer blocks, where each block consists of a multi-head self-attention (MSA) module and a multi-layer perceptron (MLP) module. With LayerNorm applied before each module and residuals added after each module, the transformer block is formulated as:

$$\boldsymbol{Y}_{l-1} = \mathrm{MSA}(\mathrm{LayerNorm}(\boldsymbol{X}_{l-1})) + \boldsymbol{X}_{l-1} \tag{1}$$

$$\boldsymbol{X}_{l} = \mathrm{MLP}(\mathrm{LayerNorm}(\boldsymbol{Y}_{l-1})) + \boldsymbol{Y}_{l-1} \tag{2}$$

where $l = 1, 2, \cdots, L$, and L is the number of the transformer blocks.

The MSA module learns inter-patch correlations of the input $\boldsymbol{X}' \in \mathbb{R}^{N\times D}$ through the following processes:

$$[\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i] = \boldsymbol{X}'\boldsymbol{W}^{qkv} + \boldsymbol{b}^{qkv} \quad i = 1, 2, \cdots, h \tag{3}$$

$$\mathrm{Attn}_i = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_i \cdot \boldsymbol{K}_i^T}{\sqrt{D_h}}\right)\boldsymbol{V}_i \tag{4}$$

$$\mathrm{MSA}(\boldsymbol{X}') = [\mathrm{Attn}_1, \mathrm{Attn}_2, \ldots, \mathrm{Attn}_h]\boldsymbol{W}^o + \boldsymbol{b}^o \tag{5}$$

where $\boldsymbol{W}^{qkv} \in \mathbb{R}^{D\times 3D_h}$, $\boldsymbol{b}^{qkv} \in \mathbb{R}^{3D_h}$, $\boldsymbol{W}^o \in \mathbb{R}^{h\cdot D_h \times D}$, $\boldsymbol{b}^o \in \mathbb{R}^D$, and h is the number of the attention heads and D_h is the feature size of each head.

The MLP module projects the features into a higher D_f-dimensional space to learn representations. Denoting the input to the MLP module as $\boldsymbol{Y}' \in \mathbb{R}^{N\times D}$, the calculation is as follows:

$$\mathrm{MLP}(\boldsymbol{Y}') = \mathrm{GELU}(\boldsymbol{Y}'\boldsymbol{W}^1 + \boldsymbol{b}^1)\boldsymbol{W}^2 + \boldsymbol{b}^2 \tag{6}$$

[1] To simplify the formulation, we ignore the batch dimension.

where $\boldsymbol{W}^1 \in \mathbb{R}^{D \times D_f}$, $\boldsymbol{b}^1 \in \mathbb{R}^{D_f}$, $\boldsymbol{W}^2 \in \mathbb{R}^{D_f \times D}$, and $\boldsymbol{b}^2 \in \mathbb{R}^D$.

As one can see, the large matrix multiplications contribute the most computational costs; hence, following previous works [24, 35], we quantize all the weights and inputs of matrix multiplications, leaving LayerNorm and Softmax operations as floating-point types. Also, for efficient inference, we employ the hardware-friendly quantizers discussed below in the inference process.

Hardware-friendly quantizers The uniform quantizer is one of the most popular choices that is well supported by the hardware, which is defined as:

$$Quant: \boldsymbol{x}^{(\mathbb{Z})} = \text{clip}\left(\left\lfloor \frac{\boldsymbol{x}}{s} \right\rceil + z, 0, 2^b - 1\right) \quad (7)$$

$$DeQuant: \hat{\boldsymbol{x}} = s\left(\boldsymbol{x}^{(\mathbb{Z})} - z\right) \approx \boldsymbol{x} \quad (8)$$

where $\boldsymbol{x}$ and $\boldsymbol{x}^{(\mathbb{Z})}$ are the floating-point and quantized values, respectively, $\lfloor \cdot \rceil$ denotes the round function, and $b \in \mathbb{N}$ is the quantization bit-width. In the de-quantization procedure[2], the de-quantized value $\hat{\boldsymbol{x}}$ approximately recovers $\boldsymbol{x}$. Importantly, $s \in \mathbb{R}^+$ is the quantization scale and $z \in \mathbb{Z}$ is the zero-point, both of which are determined by the lower and upper bounds of $\boldsymbol{x}$ as follows:

$$s = \frac{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}{2^b - 1}, \quad z = \left\lfloor -\frac{\min(\boldsymbol{x})}{s} \right\rceil \quad (9)$$

The log2 quantizer is another common and hardware-oriented choice. Since it is only applied on post-Softmax activations in this paper, we just consider the quantization of positive values as follows:

$$Quant: \boldsymbol{x}^{(\mathbb{Z})} = \text{clip}\left(\left\lfloor -\log_2 \frac{\boldsymbol{x}}{s} \right\rceil, 0, 2^b - 1\right) \quad (10)$$

$$DeQuant: \hat{\boldsymbol{x}} = s \cdot 2^{-\boldsymbol{x}^{(\mathbb{Z})}} \approx \boldsymbol{x} \quad (11)$$

where both the log2 function and the base-2 power function can be implemented using the fast and efficient bit-shifting operations [14, 22].

For the application granularity of the above quantizers, channel-wise quantization for weights and layer-wise quantization for activations can balance accuracy and efficiency [7, 12], and are well supported by both hardware and software [11, 34, 18], and thus have become a consensus in previous works [35, 4]. In this paper, we follow the above quantization granularity in the inference process.

3.2. Scale Reparam for LayerNorm Activations

In ViTs, LayerNorm is applied to normalize the input $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ in the hidden feature dimension, and its calculation process is as follows:

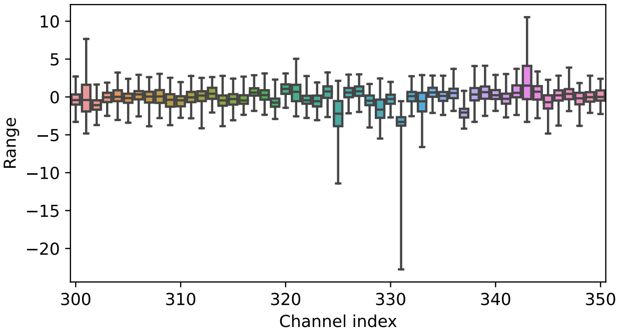

Figure 2. Boxplot of the 300th to 350th channels of the first module's post-LayerNorm activations in DeiT-S. As is evident, there is a severe inter-channel variation.

$$\text{LayerNorm}(\boldsymbol{X}_{n,:}) = \frac{\boldsymbol{X}_{n,:} - \text{E}[\boldsymbol{X}_{n,:}]}{\sqrt{\text{Var}[\boldsymbol{X}_{n,:}] + \epsilon}} \odot \boldsymbol{\gamma} + \boldsymbol{\beta} \quad (12)$$

where $n = 1, 2, \cdots, N$, $\text{E}[\boldsymbol{X}_{n,:}]$ and $\text{Var}[\boldsymbol{X}_{n,:}]$ are the mean and variance, respectively, and $\boldsymbol{\gamma} \in \mathbb{R}^D$ and $\boldsymbol{\beta} \in \mathbb{R}^D$ are the row vectors[3] of linear affine factors. Here, $\odot$ denotes Hadamard product.

Looking into the post-LayerNorm activations, we find that they have a severe inter-channel variation, which is a critical limitation to the quantization performance. More intuitively, the distribution boxplot of the 300th to 350th channels of the first module's post-LayerNorm activations in DeiT-S is illustrated in Figure 2, where the minimum, mean, and maximum ranges are 3.94, 7.11, and 22.2, respectively. In this case, layer-wise quantization that simply applys a unified quantization scale to each channel cannot accommodate such severe inter-channel variation, resulting in significant accuracy degradation. As an alternative, channel-wise quantization can address the above challenge. However, channel-wise quantization for activations requires the support of the dedicated hardware and incurs additional computational overhead.

To address the above issues, we apply the quantization-inference decoupling paradigm and propose a scale reparameterization method for post-LayerNorm activations that transforms channel-wise quantization to layer-wise quantization, achieving both the accuracy of the former and the efficiency of the latter. Specifically, given the post-LayerNorm activations $\boldsymbol{X}'$, we first perform channel-wise quantization to obtain the quantization scale $\boldsymbol{s} \in R^D$ and zero-point $\boldsymbol{z} \in Z^D$. Our goal is to reparameterize them to $\tilde{\boldsymbol{s}} = \tilde{s} \cdot \boldsymbol{1}$ and $\tilde{\boldsymbol{z}} = \tilde{z} \cdot \boldsymbol{1}$, where $\boldsymbol{1}$ is a D-dimensional row vector of all ones, and the scalars $\tilde{s} \in R^1$ and $\tilde{z} \in Z^1$ are ready for layer-wise quantization. Here, $\tilde{s}$ and $\tilde{z}$ are pre-specified and we set them to the corresponding mean values in this

[2] In actual inference, the floating-point multiplication with s is replaced by re-quantization to implement integer arithmetic.

[3] In this paper, we make the convention that all vectors serve as row vectors by default to facilitate the formulation.

paper, *i.e.*, $\tilde{s} = \mathrm{E}[s], \tilde{z} = \mathrm{E}[z]$. Defining the variation factors $\boldsymbol{r}_1 = \boldsymbol{s}/\tilde{\boldsymbol{s}}$ [4] and $\boldsymbol{r}_2 = \boldsymbol{z} - \tilde{\boldsymbol{z}}$, the following equations hold:

$$\tilde{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{r}_2 = \left\lfloor -\frac{\left[\min(\boldsymbol{X}'_{:,d})\right]_{1\le d\le D} + \boldsymbol{s}\odot\boldsymbol{r}_2}{\boldsymbol{s}} \right\rceil \tag{13}$$

$$\tilde{\boldsymbol{s}} = \frac{\boldsymbol{s}}{\boldsymbol{r}_1} = \frac{\left[\max(\boldsymbol{X}'_{:,d}) - \min(\boldsymbol{X}'_{:,d})\right]_{1\le d\le D}/\boldsymbol{r}_1}{2^b - 1} \tag{14}$$

Eq. 13 shows that adding $\boldsymbol{s}\odot\boldsymbol{r}_2$ to each channel of $\boldsymbol{X}'$ can yield $\tilde{\boldsymbol{z}}$, and Eq. 14 shows that dividing each channel of $\boldsymbol{X}'$ by $\boldsymbol{r}_1$ can yield $\tilde{\boldsymbol{s}}$. These operations can be achieved by adjusting the LayerNorm's affine factors as follows:

$$\widetilde{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta} + \boldsymbol{s}\odot\boldsymbol{r}_2}{\boldsymbol{r}_1}, \quad \widetilde{\boldsymbol{\gamma}} = \frac{\boldsymbol{\gamma}}{\boldsymbol{r}_1} \tag{15}$$

The above procedure accomplishes the reparameterization of $\tilde{\boldsymbol{s}}$ and $\tilde{\boldsymbol{z}}$, while this results in a distribution shift of activations, *i.e.*, $\widetilde{\boldsymbol{X}}'_{n,:} = (\boldsymbol{X}'_{n,:} + \boldsymbol{s}\odot\boldsymbol{r}_2)/\boldsymbol{r}_1$. Fortunately, such distribution shift can be eliminated by the inverse compensation of the next layer's weights. To be specific, through equivalent transformations we have that:

$$\begin{aligned}\boldsymbol{X}'_{n,:}\boldsymbol{W}^{qkv}_{:,j} + \boldsymbol{b}^{qkv}_j = {} & \frac{\boldsymbol{X}'_{n,:} + \boldsymbol{s}\odot\boldsymbol{r}_2}{\boldsymbol{r}_1}\left(\boldsymbol{r}_1\odot\boldsymbol{W}^{qkv}_{:,j}\right) \\ & + \left(\boldsymbol{b}^{qkv}_j - (\boldsymbol{s}\odot\boldsymbol{r}_2)\boldsymbol{W}^{qkv}_{:,j}\right)\end{aligned} \tag{16}$$

Where $j = 1, 2, \cdots, 3D_h$. Thus, to align the next layer's outputs before and after the reparameterization, the weights can be adjusted as follows:

$$\begin{aligned}\widetilde{\boldsymbol{W}}^{qkv}_{:,j} &= \boldsymbol{r}_1\odot\boldsymbol{W}^{qkv}_{:,j} \\ \widetilde{\boldsymbol{b}}^{qkv}_j &= \boldsymbol{b}^{qkv}_j - (\boldsymbol{s}\odot\boldsymbol{r}_2)\boldsymbol{W}^{qkv}_{:,j}\end{aligned} \tag{17}$$

Since the inter-channel variation factor $\boldsymbol{r}_1 \in \mathbb{R}^D$ works in different dimensions from the quantization scale $\boldsymbol{s}^{qkv} \in \mathbb{R}^{3D_h}$ and zero-point $\boldsymbol{z}^{qkv} \in \mathbb{R}^{3D_h}$ of $\boldsymbol{W}^{qkv}$, the explicit solutions of the corresponding parameters of $\widetilde{\boldsymbol{W}}^{qkv}$ cannot be directly derived and need to be re-calibrated. It is worth noting that since the weights are inherently applied channel-wise quantization, the quantization performance is not sensitive to compensating $\boldsymbol{r}_1$ on the weights, making the re-calibration of $\widetilde{\boldsymbol{W}}^{qkv}$ incur only a slight accuracy loss.

In this way, by interpretable adjustment of the LayerNorm's affine factors $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\boldsymbol{\gamma}}$ as well as the next layer's weight $\widetilde{\boldsymbol{W}}^{qkv}$ and $\widetilde{\boldsymbol{b}}^{qkv}$, we confidently reparameterize the channel-wise quantization of $\boldsymbol{X}'$ with $\boldsymbol{s}$ and $\boldsymbol{z}$ to the layer-wise quantization with $\tilde{s}$ and $\tilde{z}$. And the adjustment strategy also works for the input $\boldsymbol{Y}'$ to the MLP module. This easy-to-implement process allows us to fully benefit from the efficient inference of layer-wise quantization while featuring a robust characterization of the inter-channel variation that has only a slight performance drop compared to channel-wise quantization.

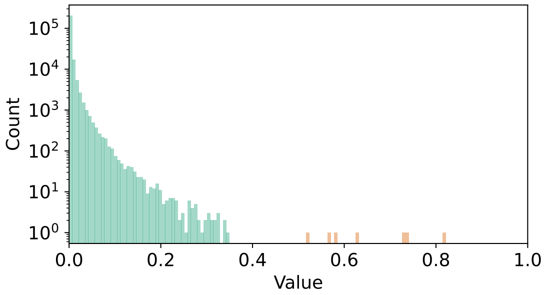

Figure 3. Histogram of the first MSA module's post-Softmax activations in DeiT-S. As one can see, it is extremely unbalanced, with the majority concentrated in small values (in green) and a few scattered in large values (in orange).

3.3. Scale Reparam for Softmax Activations

In ViTs, the Softmax operation converts the attention scores of the MSA module into probabilities, bounding the values to the (0, 1) interval. However, these probabilities, termed as post-Softmax activations, have a power-law distribution far from the Gaussian and Laplace distributions, which is extremely unbalanced and thus is identified as another key obstacle to quantization. For instance, Figure 3 shows the distribution histogram of the first MSA module's post-Softmax activations in DeiT-S. It can be observed that the majority of activations are concentrated in relatively small values, and only a few activations are discretely scattered in large values (close to 1). Statistically, even 99.2% of the activations are smaller than 0.3. It should be noted that the remaining 0.8% of activations cannot be viewed as outliers to be naively clipped; instead, these values reflect important correlations between patches that guide the MSA module to give more attention, thus we have to preserve them well in the quantization process.

To deal with the above power-law distribution, previous work [22] directly applies the log2 quantizer depending on hardware efficiency considerations. Despite the better performance than the uniform quantizer, the log2 quantizer still fails to provide a reliable and robust description of the distribution in practice. Taking the simple case of $s = 1$ as an example, the log2 quantizer takes values at levels $\{2^0, 2^{-1}, 2^{-2}, \cdots\}$, and according to Eq. 10, the values in the relatively large interval $[2^{-1.5}, 2^{-0.5}]$, *i.e.*, [0.354, 0.707], are in principle rounded to 2^{-1}. This overly sparse description of important attention scores greatly weakens the represen-

[4] In this paper, division between vectors is an element-wise operation like Hadamard product.

tational power of the MSA module. In contrast, the log$\sqrt{2}$ quantizer, which provides a higher quantization resolution for large values, can describe the distribution in a more accurate fashion. Nevertheless, it is unfriendly to hardware and fails to benefit from efficient bit-shifting operations in inference as the log2 quantizer does.

Inspired by the quantization-inference decoupling paradigm, we are motivated to explore how to convert the log$\sqrt{2}$ quantizer to log2 quantizer. With it, we can enjoy both the high accuracy of the former and the bit-shifting operations of the latter. To this end, the base changing methods are designed for the quantization and de-quantization procedures, respectively. First, given the post-Softmax activations $\boldsymbol{A}$ and the log$\sqrt{2}$ quantizer's scale $s \in \mathbb{R}^1$, according to the base changing formula of the log function we have:

$$\begin{aligned} \boldsymbol{A}^{(\mathbb{Z})} &= \operatorname{clip}\left(\left\lfloor -\log_{\sqrt{2}} \frac{\boldsymbol{A}}{s} \right\rceil, 0, 2^b - 1\right) \\ &= \operatorname{clip}\left(\left\lfloor -2\log_2 \frac{\boldsymbol{A}}{s} \right\rceil, 0, 2^b - 1\right) \end{aligned} \tag{18}$$

Thus, for the quantization procedure, the conversion to the log2 quantizer can be achieved by simply multiplying by a constant factor. Similarly, in the de-quantization procedure, the base changing formula of the pow function is utilized to obtain the base-2 form; however, the new exponential term $-\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2}$ is not guaranteed to be an integer that is necessary to perform the bit-shifting operations. Therefore, we discuss the parity of $-\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2}$ by case as follows:

$$\begin{aligned} \widehat{\boldsymbol{A}} &= s \cdot \sqrt{2}^{-\boldsymbol{A}^{(\mathbb{Z})}} = s \cdot 2^{-\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2}} \\ &= \begin{cases} s \cdot 2^{-\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2}} & \boldsymbol{A}^{(\mathbb{Z})} = 2k, k \in \mathbb{Z} \\ s \cdot 2^{-\frac{\boldsymbol{A}^{(\mathbb{Z})}+1}{2}} \cdot \sqrt{2} & \boldsymbol{A}^{(\mathbb{Z})} = 2k+1, k \in \mathbb{Z} \end{cases} \\ &= s \cdot 2^{\left\lfloor -\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2} \right\rfloor} \cdot \left[\mathbb{1}(\boldsymbol{A}^{(\mathbb{Z})}) \cdot (\sqrt{2} - 1) + 1\right] \end{aligned} \tag{19}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, $\left\lfloor -\frac{\boldsymbol{A}^{(\mathbb{Z})}}{2} \right\rfloor$ is consistently an integer, and $\mathbb{1}(\cdot)$ is a parity indicator function that is 0 at even numbers and 1 at odd numbers.

The above parity indicator function and its coefficients can be merged into s to obtain the reparameterized scale $\tilde{s}$ as follows:

$$\tilde{s} = s \cdot \left[\mathbb{1}(\boldsymbol{A}^{(\mathbb{Z})}) \cdot (\sqrt{2} - 1) + 1\right] \tag{20}$$

Eventually, thanks to the reparameterization of $\tilde{s}$, the de-quantization procedure is also able to benefit from the efficient bit-shifting operations. Note that compared to the previous scale s, the reparameterized scale $\tilde{s}$ only introduces a slight additional computational overhead in the inference process, due to the fact that the parity indicator function can be computed with great efficiency, *e.g.*, by simply querying the least significant bit of $\boldsymbol{A}^{(\mathbb{Z})}$ on FPGAs.

4. Experiments

4.1. Experimental Setup

Models and datasets For the image classification task, RepQ-ViT is evaluated on ImageNet [13] dataset with different model variants: ViT [5], DeiT [30], and Swin [23]. For the object detection and instance segmentation tasks, RepQ-ViT is evaluated on COCO [21] dataset using two typical frameworks: Mask R-CNN [9] and Cascade Mask R-CNN [1] with Swin [23] as the backbone.

Implementation details All pretrained full-precision models are obtained from Timm[5] library. For a fair comparison with the previous works [35, 4], we randomly select 32 samples from ImageNet dataset for image classification and 1 sample from COCO dataset for object detection and instance segmentation to calibrate the quantization parameters. For the calibration strategy, we apply the prevalent Percentile [15] method, with channel-wise quantization for weights and layer-wise quantization for activations in the inference process. Scale reparameterization is applied to post-LayerNorm activations in all blocks (including those in PatchMerging layers of Swin) and to post-Softmax activations in all MSA modules. Note that our proposed RepQ-ViT is free of any hyperparameters and thus offers a high ease of implementation and generality, which is significantly superior to existing methods.

4.2. Quantization Results on ImageNet Dataset

We start by comparing the quantization results of the proposed RepQ-ViT and existing methods on ImageNet dataset for image classification, as reported in Table 1. It is worth noting that hyperparameters and reconstruction procedures are also explicitly presented in the Table as conditional indicators. Thanks to the non-dependence on these two indicators, RepQ-ViT is believed to be more practical and general in real-world applications. We focus on the performance of low-bit quantization, including W4/A4 and W6/A6 quantization, to highlight the advantages of RepQ-ViT. In W4/A4 quantization, previous works all suffer from non-trivial performance degradation. For instance, FQ-ViT becomes infeasible with only 0.1% accuracy, while PTQ4ViT and APQ-ViT improve accuracy with the help of reconstruction but remain far from practical usability. Fortunately, RepQ-ViT can maintain the data distribution through the design of complex quantizers to achieve robust and substantial improvement of the quantization performance, with encouraging 27.07% and 25.48% improvement over APQ-ViT in ViT-B and DeiT-S quantization, respectively. When quantizing DeiT-B, Swin-S, and Swin-B, RepQ-ViT consistently obtains an interesting decrease in

[5] https://github.com/rwightman/pytorch-image-models

Method	No HP	No REC	Prec. (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 [22]	×	✓	4/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT [35]	×	×	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT [4]	×	×	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
RepQ-ViT (ours)	✓	✓	4/4	**65.05**	**68.48**	**57.43**	**69.03**	**75.61**	**79.45**	**78.32**
FQ-ViT [22]	×	✓	6/6	4.26	0.10	58.66	45.51	64.63	66.50	52.09
PSAQ-ViT [19]	×	✓	6/6	37.19	41.52	57.58	63.61	67.95	72.86	76.44
Ranking [24]	×	×	6/6	-	75.26	-	74.58	77.02	-	-
PTQ4ViT [35]	×	×	6/6	78.63	81.65	69.68	76.28	80.25	82.38	84.01
APQ-ViT [4]	×	×	6/6	79.10	82.21	70.49	77.76	80.42	82.67	84.18
RepQ-ViT (ours)	✓	✓	6/6	**80.43**	**83.62**	**70.76**	**78.90**	**81.27**	**82.79**	**84.57**

Table 1. Quantization results of image classification on ImageNet dataset, where each data presents the Top-1 accuracy (%) obtained by quantizing each model. Here, we abbreviate "No Hyperparameters" as "No HP" and "No Reconstruction" as "No REC", and "Prec. (W/A)" indicates that the quantization bit-precision of the weights and activations are W and A bits, respectively.

Method	No HP	No REC	Prec. (W/A)	Mask R-CNN				Cascade Mask R-CNN			
				w. Swin-T		w. Swin-S		w. Swin-T		w. Swin-S	
				$\mathrm{AP}^{\mathrm{box}}$	$\mathrm{AP}^{\mathrm{mask}}$	$\mathrm{AP}^{\mathrm{box}}$	$\mathrm{AP}^{\mathrm{mask}}$	$\mathrm{AP}^{\mathrm{box}}$	$\mathrm{AP}^{\mathrm{mask}}$	$\mathrm{AP}^{\mathrm{box}}$	$\mathrm{AP}^{\mathrm{mask}}$
Full-Precision	-	-	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT [35]	×	×	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT [4]	×	×	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
RepQ-ViT (ours)	✓	✓	4/4	**36.1**	**36.0**	44.2	**40.2**	**47.0**	**41.4**	**49.3**	**43.1**
PTQ4ViT [35]	×	×	6/6	5.8	6.8	6.5	6.6	14.7	13.6	12.5	10.8
APQ-ViT [4]	×	×	6/6	**45.4**	41.2	**47.9**	42.9	48.6	42.5	50.5	43.9
RepQ-ViT (ours)	✓	✓	6/6	45.1	**41.2**	47.8	**43.0**	**50.0**	**43.5**	**51.4**	**44.6**

Table 2. Quantization results of object detection and instance segmentation on COCO dataset. Here, "$\mathrm{AP}^{\mathrm{box}}$" is the box average precision for object detection, and "$\mathrm{AP}^{\mathrm{mask}}$" is the mask average precision for instance segmentation.

accuracy of less than 7%. To the best of our knowledge, we are the first to break the limit of 4-bit PTQ of ViTs to the usable level. In addition, in W6/A6 quantization, RepQ-ViT can achieve an accuracy comparable to that of the full-precision baseline with a model size compressed by 5.3 times. In DeiT-B and Swin-S quantization, RepQ-ViT achieves 81.27% and 82.79% accuracy, respectively, with only 0.53% and 0.44% accuracy loss.

4.3. Quantization Results on COCO Dataset

The object detection and instance segmentation experiments are conducted on COCO dataset, and the quantization results are shown in Table 2. As before, we also explicitly list whether hyperparameters and reconstruction procedures are required. Due to the more complex model architectures in high-level tasks, PTQ4ViT's twin-scale search loses its viability, leading to disappointing quantization performance. APQ-ViT is not robust to different backbones; it yields good results with Swin-S as the backbone while it causes severe performance degradation when Swin-T serves as the backbone, for instance, in W4/A4 quantization of Cascade Mask R-CNN framework with Swin-T, box AP and mask AP are degraded by 23.2 and 19.3, respectively. This greatly limits the practical deployment and application of the quantized models. Compared with previous methods, our proposed RepQ-ViT achieves more advanced performance with high robustness. When performing W4/A4 quantization in the case of Swin-T backbone, for Mask R-CNN framework, RepQ-ViT improves over APQ-ViT by 12.4 box AP and 13.4 mask AP; for Cascade Mask R-CNN framework, RepQ-ViT yields a boost of 19.8 box AP and 17.0 mask AP over APQ-ViT. Moreover, in W6/A6 quantization, RepQ-ViT produces only a slight accuracy loss over the full-precision baseline. When quantizing Cascade Mask R-CNN framework with Swin-T, RepQ-ViT reached 50.0 box AP and 43.5 mask AP, which is just 0.4 box AP and 0.2 mask AP lower than the full-precision baseline. Similar results can also be obtained when Swin-S serves as the backbone, achieving 51.4 box AP and 44.6 mask AP.

Model	Method	Hardware	Top-1 (%)
DeiT-S	Full-Precision	-	79.85
	Layer-Wise Quant.	✓	33.17
	Channel-Wise Quant.	×	**70.28**
	Scale Reparam (ours)	✓	69.03
Swin-S	Full-Precision	-	83.23
	Layer-Wise Quant.	✓	57.63
	Channel-Wise Quant.	×	**80.52**
	Scale Reparam (ours)	✓	79.45

Table 3. Ablation studies of different quantizers (W4/A4) for post-LayerNorm activations. Here, "Hardware" indicates whether the obtained quantized model is hardware-friendly and can be efficiently computed in inference.

Model	Method	Hardware	Top-1 (%)
DeiT-S	Full-Precision	-	79.85
	Log2 Quant.	✓	67.71
	Log$\sqrt{2}$ Quant.	×	**69.03**
	Scale Reparam (ours)	✓	**69.03**
Swin-S	Full-Precision	-	83.23
	Log2 Quant.	✓	77.87
	Log$\sqrt{2}$ Quant.	×	**79.45**
	Scale Reparam (ours)	✓	**79.45**

Table 4. Ablation studies of different quantizers (W4/A4) for post-Softmax activations.

4.4. Ablation Studies

To validate the effectiveness of the main components of the proposed RepQ-ViT framework, we perform two ablation studies of the scale reparameterization methods for post-LayerNorm and post-Softmax activations, respectively, as shown in Tables 3 and 4.

Table 3 reports the ablation study results of different quantizers (W4/A4) for post-LayerNorm activations. Taking DeiT-S as an example, direct layer-wise quantization cannot represent the data distribution well and achieves only 33.17% accuracy. Applying channel-wise quantization can solve the above issue with 70.28% accuracy; however, it fails to satisfy the hardware characteristics to enable efficient calculations in the inference process. Therefore, the scale reparameterization method, which converts channel-wise quantization to layer-wise quantization, can allow for both high accuracy and efficient inference. Note that due to the recalibration of $\widetilde{\boldsymbol{W}}^{qkv}$, the scale reparameterization method produces a slight performance drop (1.25%) compared to channel-wise quantization.

The results of different quantizers (W4/A4) for post-Softmax activations are reported in Table 4. Log$\sqrt{2}$ quantizers can better fit the extreme distributions of attention scores, and in particular, has better a representation of scattered large values, thus providing a 1.58% improvement in accuracy over simple log2 quantizers in the case of Swin-S quantization. To solve the inefficiency problem of log$\sqrt{2}$ quantizers, scale reparameterization is applied to accomplish the conversion to log2 quantizers. Here, the scale reparameterization method for post-Softmax activations employs exactly equivalent transformations and thus yields the same accuracy as log$\sqrt{2}$ quantizers, at the cost of only a slight additional computational overhead in inference compared to log2 quantizers.

Model	Method	Top-1 (%)	Calib Data	GPU Min.
DeiT-S	Full-Precision	79.85	-	-
	FQ-ViT [22]	0.10	1000	**0.5**
	PTQ4ViT [35]	34.08	**32**	3.2
	RepQ-ViT (ours)	**69.03**	**32**	1.3
Swin-S	Full-Precision	83.23	-	-
	FQ-ViT [22]	0.10	1000	**1.1**
	PTQ4ViT [35]	76.09	**32**	7.7
	RepQ-ViT (ours)	**79.45**	**32**	2.9

Table 5. Comparison of the data quantity and time consumption (in minutes) during the quantization (W4/A4) calibration.

4.5. Efficiency Analysis

We also compare the efficiency of different methods, including the data quantity and time consumption required for quantization calibration, as shown in Table 5. Here, time consumption is measured on a single 3090 GPU. Since there is no reconstruction in FQ-ViT, the quantized models can be obtained rapidly, however, the performance drops severely even with 1000 samples for calibration. Our RepQ-ViT requires only 32 samples as PTQ4ViT, while it is free of reconstruction and thus can yield quantized models with higher accuracy more quickly compared to PTQ4ViT.

5. Conclusions

In this paper, we propose RepQ-ViT, a novel post-training quantization framework for vision transformers. RepQ-ViT applies the quantization-inference decoupling paradigm, where complex quantizers are employed in the quantization process and simple hardware-friendly quantizers are employed in the inference process, and both are explicitly bridged by scale reparameterization. More specifically, RepQ-ViT resolves the extreme distributions of two components: for post-LayerNorm activations with severe inter-channel variation, channel-wise quantization is initially applied and then is reparameterized to layer-wise quantization; for post-Softmax activations with power-law features, log$\sqrt{2}$ quantization is initially applied and then is reparameterized to log2 quantization. Exhaustive experiments are performed to fully validate the superiority of

RepQ-ViT, showing that it significantly outperforms existing methods in low-bit quantization.

In the future, one can extend the reparameterization of channel-wise to layer-wise quantization to more activations. One can also try to combine $\log\sqrt{2}$ and log2 quantization to better describe the power-law distribution.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0119402; in part by the National Natural Science Foundation of China under Grant 62276255.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 6

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, Cham, 2020. 1, 2

[3] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 1, 3

[4] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5380–5388, 2022. 1, 3, 4, 6, 7

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 6

[6] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 1, 3

[7] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 1, 3, 4

[8] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022. 1, 2

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[10] Zejiang Hou and Sun-Yuan Kung. Multi-dimensional vision transformer compression via dependency guided gaussian process search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3669–3678, 2022. 1, 3

[11] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. 4

[12] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. 1, 3, 4

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 6

[14] Edward H Lee, Daisuke Miyashita, Elaina Chai, Boris Murmann, and S Simon Wong. Lognet: Energy-efficient neural networks using logarithmic computation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5900–5904. IEEE, 2017. 4

[15] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2810–2819, 2019. 6

[16] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. 1, 3

[17] Zhikai Li, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Psaq-vit v2: Towards accurate and general data-free quantization for vision transformers. *arXiv preprint arXiv:2209.05687*, 2022. 3

[18] Zhikai Li and Qingyi Gu. I-vit: integer-only quantization for efficient vision transformer inference. *arXiv preprint arXiv:2207.01405*, 2022. 4

[19] Zhikai Li, Liping Ma, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Patch similarity aware data-free quantization for vision transformers. In *European Conference on Computer Vision*, pages 154–170. Springer, 2022. 3, 7

[20] Zhikai Li, Liping Ma, Xianlei Long, Junrui Xiao, and Qingyi Gu. Dual-discriminator adversarial framework for data-free quantization. *Neurocomputing*, 511:67–77, 2022. 1

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[22] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. In *International Joint Conference on Artificial Intelligence*, pages 1173–1179, 2022. 1, 3, 4, 5, 7, 8

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2, 6

[24] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3, 4, 7

[25] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 3

[26] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020. 1, 3

[27] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019. 3

[28] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1, 2

[29] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12165–12174, 2022. 1, 3

[30] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2, 6

[31] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019. 3

[32] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards accurate post-training network quantization via bit-split and stitching. In *International Conference on Machine Learning*, pages 9847–9856. PMLR, 2020. 1

[33] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv preprint arXiv:2203.05740*, 2022. 1

[34] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pages 11875–11886. PMLR, 2021. 4

[35] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision*, pages 191–207. Springer, 2022. 1, 3, 4, 6, 7, 8

[36] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12145–12154, 2022. 3

[37] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 3

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2