PTQ4ViT: Post-Training Quantization for Vision Transformers with Twin Uniform Quantization

Zhihang Yuan[1,3] *, Chenhao Xue[1] *, Yiqi Chen[1], Qiang Wu[3], and Guangyu Sun[2] **

[1] School of Computer Science, Peking University
{yuanzhihang, xch927027}@pku.edu.cn
[2] School of Integrated Circuits, Peking University
gsun@pku.edu.cn
[3] Houmo AI

Abstract. Quantization is one of the most effective methods to compress neural networks, which has achieved great success on convolutional neural networks (CNNs). Recently, vision transformers have demonstrated great potential in computer vision. However, previous post-training quantization methods performed not well on vision transformer, resulting in more than 1% accuracy drop even in 8-bit quantization. Therefore, we analyze the problems of quantization on vision transformers. We observe the distributions of activation values after softmax and GELU functions are quite different from the Gaussian distribution. We also observe that common quantization metrics, such as MSE and cosine distance, are inaccurate to determine the optimal scaling factor. In this paper, we propose the twin uniform quantization method to reduce the quantization error on these activation values. And we propose to use a Hessian guided metric to evaluate different scaling factors, which improves the accuracy of calibration at a small cost. To enable the fast quantization of vision transformers, we develop an efficient framework, PTQ4ViT. Experiments show the quantized vision transformers achieve near-lossless prediction accuracy (less than 0.5% drop at 8-bit quantization) on the ImageNet classification task.

1 Introduction

The self-attention module is the basic building block of the transformer to capture global information [23]. Inspired by the success of transformers [6,2] on natural language processing (NLP) tasks, researchers have brought the self-attention module into computer vision [7,15]. They replaced the convolution layers in convolutional neural networks (CNNs) with self-attention modules and they called these networks vision transformers. Vision transformers are comparable to CNNs on many computer vision tasks and have great potential to be deployed on various applications [11].

* First author and Second Author contribute equally to this paper.
** Corresponding author

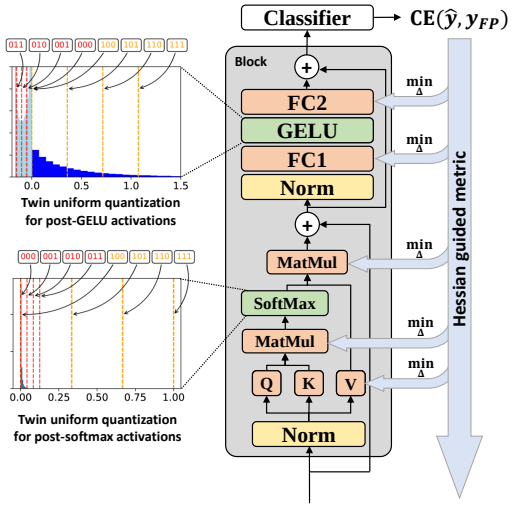

Fig. 1: Overview of the PTQ4ViT.

However, both the CNN and the vision transformer are computationally intensive and consume much energy. The larger and larger scales of neural networks block their deployment on various hardware devices, such as mobile phones and IoT devices, and increase carbon emissions. It is required to compress these neural networks. Quantization is one of the most effective ways to compress neural networks [9]. The floating-point values are quantized to integers with a low bit-width, reducing the memory consumption and the computation cost.

There are two types of quantization methods, quantization-aware training (QAT) [4,28] and post-training quantization (PTQ) [1,5]. Although QAT can generate the quantized network with a lower accuracy drop, the training of the network requires a training dataset, a long optimization time, and the tuning of hyper-parameters. Therefore, QAT is impractical when the training dataset is not available or rapid deployment is required. While PTQ quantizes the network with unlabeled calibration images after training, which enables fast quantization and deployment.

Although PTQ has achieved great success on CNNs, directly bringing it to vision transformer results in more than 1% accuracy drop even with 8-bit quantization [16]. Therefore, we analyze the problems of quantization on vision transformers. We collect the distribution of activation values in the vision transformer and observe there are some special distributions. 1) The values after softmax have a very unbalanced distribution in $[0, 1]$, where most of them are very close to zero. Although the number of large values is very small, they mean high attention between two patches, which is of vital importance in the attention mechanism. This requires a large scaling factor to make the quantization range cover the large value. However, a big scaling factor quantizes the small values to zero, resulting in a large quantization error. 2) The values after the GELU

function have an asymmetrical distribution, where the positive values have a large distribution range while the negative values have a very small distribution range. It's difficult to well quantify both the positive values and negative values with uniform quantization. Therefore, we propose the twin uniform quantization, which separately quantifies the values in two ranges. To enable its efficient processing on hardware devices, we design a data format and constrain the scaling factors of the two ranges.

The second problem is that the metric to determine the optimal scaling factor is not accurate on vision transformers. There are various metrics in previous PTQ methods, including MSE, cosine distance, and Pearson correlation coefficient between the layer outputs before and after quantization. However, we observe they are inaccurate to evaluate different scaling factor candidates because only the local information is used. Therefore, we propose to use the Hessian guided metric to determine the quantization parameters, which is more accurate. The proposed methods are demonstrated in Fig. 1.

We develop a post-training quantization framework for vision transformers using twin uniform quantization, PTQ4ViT. [4] Experiments show the quantized vision transformers (ViT, DeiT, and Swin) achieve near-lossless prediction accuracy (less than 0.5% drop at 8-bit quantization) on the ImageNet classification task.

Our contributions are listed as follows:

- We find the problems in PTQ on vision transformers are special distributions of post-softmax and post-GELU activations and the inaccurate metric.
- We propose the twin uniform quantization to handle the special distributions, which can be efficiently processed on existing hardware devices including CPU and GPU.
- We propose to use the Hessian guided metric to determine the optimal scaling factors, which replaces the inaccurate metrics.
- The quantized networks achieve near-lossless prediction accuracy, making PTQ acceptable on vision transformers.

2 Background and Related Work

2.1 Vision Transformer

In the last few years, convolution neural networks (CNNs) have achieved great success in computer vision. The convolution layer is a fundamental component of CNNs to extract features using local information. Recently, the position of CNNs in computer vision is challenged by vision transformers, which take the self-attention modules [23] to make use of the global information. DETR [3] is the first work to replace the object detection head with a transformer, which directly regresses the bounding boxes and achieves comparable results with the CNN-based head. ViT [7] is the first architecture that replaces all convolution

[4] Code is in https://github.com/hahnyuan/PTQ4ViT.

layers, which achieves better results on image classification tasks. Following ViT, various vision transformer architectures have been proposed to boost performance [25,15]. Vision transformers have been successfully applied to downstream tasks [15,24,12]. They have great potential for computer vision tasks [11].

The input of a transformer is a sequence of vectors. An image is divided into several patches and a linear projection layer is used to project each patch to a vector. These vectors form the input sequence of the vision transformer. We denote these vectors as $X \in R^{N \times D}$, where N is the number of patches and D is the hidden size, which is the size of the vector after linear projection.

A vision transformer contains some blocks. As shown in Fig. 1, each block is composed of a multi-head self-attention module (MSA) and a multi-layer perceptron (MLP). MSA generates the attention between different patches to extract features with global information. Typical MLP contains two fully-connected layers (FC) and the GELU activation function is used after the first layer. The input sequence is first fed into each self-attention head of MSA. In each head, the sequence is linearly projected to three matrices, query $Q = XW^Q$, key $K = XW^K$, and value $V = XW^V$. Then, matrix multiplication QK^T calculates the attention scores between patches. The softmax function is used to normalize these scores to attention probability P. The output of the head is matrix multiplication PV. The process is formulated as Eq. (1):

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where d is the hidden size of head. The outputs of multiple heads are concatenated together as the output of MSA.

Vision transformers have a large amount of memory, computation, and energy consumption, which hinders their deployment in real-world applications. Researchers have proposed a lot of methods to compress vision transformers, such as patch pruning [21], knowledge distillation [13] ,and quantization [16].

2.2 Quantization

Network quantization is one of the most effective methods to compress neural networks. The weight values and activation values are transformed from floating-point to integer with lower bit-width, which significantly decreases the memory consumption, data movement, and energy consumption. The uniform symmetric quantization is the most widely used method, which projects a floating-point value x to a k-bit integer value x_q with a scaling factor Δ:

$$x_q = \Psi_k(x, \Delta) = \text{clamp}(\text{round}(\frac{x}{\Delta}), -2^{k-1}, 2^{k-1} - 1), \tag{2}$$

where round projects a value to an integer and clamp constrains the output in the range that k-bit integer can represent. We propose the twin uniform quantization, which separately quantifies the values in two ranges. [8] also uses multiple quantization ranges. However, their method targets CNN and is not

suitable for ViT. They use an extra bit to represent which range is used, taking 12.5% more storage than our method. Moreover, they use FP32 computation to align the two ranges, which is not efficient. Our method uses the shift operation, avoiding the format transformation and extra FP32 multiplication and FP32 addition.

There are two types of quantization methods, quantization-aware training (QAT) [4,28] and post-training quantization (PTQ) [1,5]. QAT methods combine quantization with network training. It optimizes the quantization parameters to minimize the task loss on a labeled training dataset. QAT can be used to quantize transformers [18]. Q-BERT [20] uses the Hessian spectrum to evaluate the sensitivity of the different tensors for mixed-precision, achieving 3-bit weight and 8-bit activation quantization. Although QAT achieves lower bit-width, it requires a training dataset, a long quantization time, and hyper-parameter tuning. PTQ methods quantize networks with a small number of unlabeled images, which is significantly faster than QAT and doesn't require any labeled dataset. PTQ methods should determine the scaling factors Δ of activations and weights for each layer. Choukroun et al. [5] proposed to minimize the mean square error (MSE) between the tensors before and after quantization. EasyQuant [27] uses the cosine distance to improve the quantization performance on CNN. Recently, Liu et al. [16] first proposed a PTQ method to quantize the vision transformer. Pearson correlation coefficient and ranking loss are used as the metrics to determine the scaling factors. However, these metrics are inaccurate to evaluate different scaling factor candidates because only the local information is used.

3 Method

In this section, we will first introduce a base PTQ method for vision transformers. Then, we will analyze the problems of quantization using the base PTQ and propose methods to address the problems. Finally, we will introduce our post-training quantization framework, PTQ4ViT.

3.1 Base PTQ for Vision Transformer

Matrix multiplication is used in the fully-connected layer and the computation of QK^T and PV, which is the main operation in vision transformers. In this paper, we formulate it as $O = AB$ and we will focus on its quantization. A and B are quantized to k-bit using the symmetric uniform quantization with scaling factors Δ_A and Δ_B. According to Eq. (2), we have $A_q = \Psi_k(A, \Delta_A)$ and $B_q = \Psi_k(B, \Delta_B)$. In base PTQ, the distance of the output before and after quantization is used as metric to determine the scaling factors, which is formulated as:

$$\min_{\Delta_A, \Delta_B} \text{distance}(O, \hat{O}), \tag{3}$$

where $\hat{O}$ is the output of the matrix multiplication after quantization $\hat{O} = \Delta_A \Delta_B A_q B_q$.

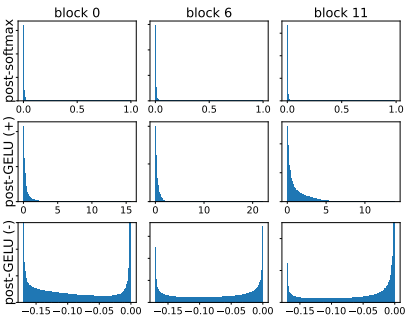

Fig. 2: Distributions of the post-softmax values, the positive post-GELU values, and the negative post-GELU values.

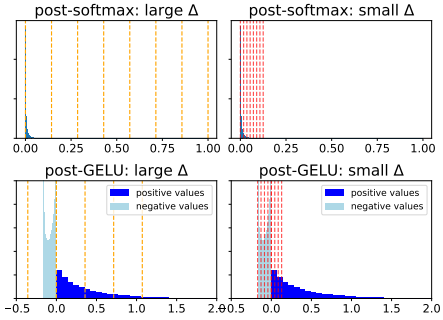

Fig. 3: Demonstration of different scaling factors to quantize the post-softmax and post-GELU activation values.

The same as [27], we use cosine distance as the metric to calculate the distance. We make the search spaces of Δ_A and Δ_B by linearly dividing $[\alpha\frac{A_{max}}{2^{k-1}}, \beta\frac{A_{max}}{2^{k-1}}]$ and $[\alpha\frac{B_{max}}{2^{k-1}}, \beta\frac{B_{max}}{2^{k-1}}]$ to n candidates, respectively. A_{max} and B_{max} are the maximum absolute value of A and B. α and β are two parameters to control the search range. We alternatively search for the optimal scaling factors Δ_A^* and Δ_B^* in the search space. Firstly, Δ_B is fixed, and we search for the optimal Δ_A to minimize distance$(O, \hat{O})$. Secondly, Δ_A is fixed, and we search for the optimal Δ_B to minimize distance$(O, \hat{O})$. Δ_A and Δ_B are alternately optimized for several rounds.

The values of A and B are collected using unlabeled calibration images. We search for the optimal scaling factors of activation or weight layer-by-layer. However, the base PTQ results in more than 1% accuracy drop on quantized vision transformer in our experiments.

3.2 Twin Uniform Quantization

The activation values in CNNs are usually considered Gaussian distributed. Therefore, most PTQ quantization methods are based on this assumption to determine the scaling factor. However, we observe the distributions of post-softmax values and post-GELU values are quite special as shown in Fig. 2. Specifically, (1) The distribution of activations after softmax is very unbalanced, in which most values are very close to zero and only a few values are close to one. (2) The values after the GELU function have a highly asymmetric distribution, in which the unbounded positive values are large while the negative values have a very small distribution range. As shown in Fig. 3, we demonstrate the quantization points of the uniform quantization using different scaling factors.

For the values after softmax, a large value means that there is a high correlation between the two patches, which is important in the self-attention mechanism. A larger scaling factor can reduce the quantization error of these large

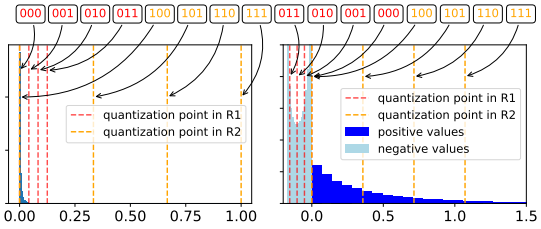

Fig. 4: Demonstration of the 3-bit twin uniform quantization on post-softmax values (left) and post-GELU values (right). We annotate the binary values for different quantization points.

values, which causes smaller values to be quantized to zero. While a small scaling factor makes the large values quantized to small values, which significantly decreases the intensity of attention between two patches. For the values after GELU, it is difficult to quantify both positive and negative values well with symmetric uniform quantization. Non-uniform quantization [10] can be used to solve the problem. It can set the quantization points according to the distribution, ensuring the overall quantization error is small. However, most hardware devices cannot efficiently process the non-uniform quantized values. Acceleration can be achieved only on specially designed hardware.

We propose the twin uniform quantization, which can be efficiently processed on existing hardware devices including CPUs and GPUs. As shown in Fig. 4, twin uniform quantization has two quantization ranges, R1 and R2, which are controlled by two scaling factors Δ_{R1} and Δ_{R2}, respectively. The k-bit twin uniform quantization is be formulated as:

$$\mathrm{T}_k(x, \Delta_{\mathrm{R1}}, \Delta_{\mathrm{R2}}) = \begin{cases} \Psi_{k-1}(x, \Delta_{\mathrm{R1}}), x \in \mathrm{R1} \\ \Psi_{k-1}(x, \Delta_{\mathrm{R2}}), otherwise \end{cases} . \tag{4}$$

For values after softmax, the values in $\mathrm{R1} = [0, 2^{k-1}\Delta^s_{\mathrm{R1}})$ can be well quantified by using a small Δ^s_{R1}. To avoid the effect of calibration dataset, we keeps Δ^s_{R2} fixed to $1/2^{k-1}$. Therefore, $\mathrm{R2} = [0, 1]$ can cover the whole range, and large values can be well quantified in R2. For activation values after GELU, negative values are located in $\mathrm{R1} = [-2^{k-1}\Delta^g_{\mathrm{R1}}, 0]$ and positive values are located in $\mathrm{R2}=[0, 2^{k-1}\Delta^g_{\mathrm{R2}}]$. We also keep Δ^g_{R1} fixed to make R1 just cover the entire range of negative numbers. Since different quantization parameters are used for positive and negative values respectively, the quantization error can be effectively reduced. When calibrating the network, we search for the optimal Δ^s_{R1} and Δ^g_{R2}.

The uniform symmetric quantization uses the k bit signed integer data format. It consists of one sign bit and $k-1$ bits representing the quantity. In order to efficiently store the twin-uniform-quantized values, we design a new data format. The most significant bit is the range flag to represent which range is used (0 for R1, 1 for R2). The other $k-1$ bits compose an unsigned number to represent

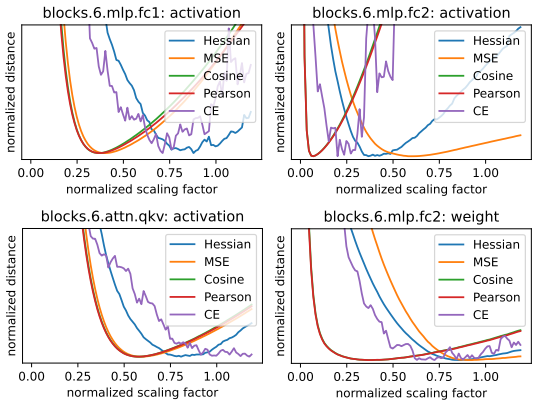

Fig. 5: The distance between the layer outputs before and after quantization and the change of task loss (CE) under different scaling factors on ViT-S/224. The x-axis is the normalized scaling factor by dividing $\frac{A_{max}}{2^{k-1}}$ or $\frac{B_{max}}{2^{k-1}}$.

the quantity. Because the sign of values in the same range is the same, the sign bit is removed.

Data in different ranges need to be multiplied and accumulated in matrix multiplication. In order to efficiently process with the twin-uniform-quantized values on CPUs or GPUs, we constrain the two ranges with $\Delta_{\mathrm{R2}} = 2^m \Delta_{\mathrm{R1}}$, where m is an unsigned integer. Assuming a_q is quantized in R1 and b_q is quantized in R2, the two values can be aligned:

$$a_q \times \Delta_{\mathrm{R1}} + b_q \times \Delta_{\mathrm{R2}} = (a_q + b_q \times 2^m)\Delta_{\mathrm{R1}}. \tag{5}$$

We left shift b_q by m bits, which is the same as multiplying the value by 2^m. The shift operation is very efficient on CPUs or GPUs. Without this constraint, multiplication is required to align the scaling factor, which is much more expensive than shift operations.

3.3 Hessian Guided Metric

Next, we will analyze the metrics to determine the scaling factors of each layer. Previous works [5,27,16] greedily determine the scaling factors of inputs and weights layer by layer. They use various kinds of metrics, such as MSE and cosine distance, to measure the distance between the original and the quantized outputs. The change in the internal output is considered positively correlated with the task loss, so it is used to calculate the distance.

We plot the performance of different metrics in Fig. 5. We observe that MSE, cosine distance, and Pearson correlation coefficient are inaccurate compared with task loss (cross-entropy) on vision transformers. The optimal scaling factors based on them are not consistent with that based on task loss. For instance, on blocks.6.mlp.fc1:activation, they indicate that a scaling factor around

$0.4\frac{A_{max}}{2^{k-1}}$ is the optimal one, while the scaling factor around $0.75\frac{A_{max}}{2^{k-1}}$ is the optimal according to the task loss. Using these metrics, we get sub-optimal scaling factors, causing the accuracy degradation. The distance between the last layer's output before and after quantization can be more accurate in PTQ. However, using it to determine the scaling factors of internal layers is impractical because it requires executing the network many times to calculate the last layer's output, which consumes too much time.

To achieve high accuracy and quick quantization at the same time, we propose to use the Hessian guided metric to determine the scaling factors. In the classification task, the task loss is $L = \mathrm{CE}(\hat{y}, y)$, where CE is cross-entropy, $\hat{y}$ is the output of the network, and y is the ground truth[5]. When we treat the weights as variables, the expectation of loss is a function of weight $\mathbb{E}[L(W)]$. The quantization brings a small perturbation ϵ on weight $\hat{W} = W + \epsilon$. We can analyze the influence of quantization on task loss by Taylor series expansion.

$$\mathbb{E}[L(\hat{W})] - \mathbb{E}[L(W)] \approx \epsilon^T \bar{g}^{(W)} + \frac{1}{2}\epsilon^T \bar{H}^{(W)} \epsilon, \tag{6}$$

where $\bar{g}^{(W)}$ is the gradients and $\bar{H}^{(W)}$ is the Hessian matrix. The target is to find the scaling factors to minimize the influence: $\min_{\Delta}(\mathbb{E}[L(\hat{W})] - \mathbb{E}[L(W)])$. Based on the layer-wise reconstruction method in [14], the optimization can be approximated[6] by:

$$\min_{\Delta} \mathbb{E}[(\hat{O}^l - O^l)^T diag\big((\frac{\partial L}{\partial O_1^l})^2, \ldots, (\frac{\partial L}{\partial O_{|O^l|}^l})^2\big)(\hat{O}^l - O^l)], \tag{7}$$

where O^l and $\hat{O}^l$ are the outputs of the l-th layer before and after quantization, respectively. As shown in Fig. 5, the optimal scaling factor indicated by Hessian guided metric is closer to that indicated by task loss (CE). Although it still has a gap with the task loss, Hessian guided metric significantly improves the performance. For instance, on blocks.6.mlp.fc1:activation, the optimal scaling factor indicated by Hessian guided metric has less influence on task loss than other metrics.

3.4 PTQ4ViT Framework

To achieve fast quantization and deployment, we develop an efficient post-training quantization framework for vision transformers, PTQ4ViT. Its flow is described in Algorithm 1. It supports the twin uniform quantization and Hessian guided metric. There are two quantization phases. 1) The first phase is to collect the output and the gradient of the output in each layer before quantization. The outputs of the l-th layer O^l are calculated through forward propagation on the calibration dataset. The gradients $\frac{\partial L}{\partial O_1^l}, \ldots, \frac{\partial L}{\partial O_a^l}$ are calculated through backward

[5] The ground truth y is not available in PTQ, so we use the prediction of floating-point network y_{FP} to approximate it.

[6] The derivation of it is in Appendix.

Algorithm 1: Searches for the optimal scaling factors of each layer.

for l *in* 1 *to* L **do**
  forward-propagation $O^l \leftarrow A^l B^l$;
end
for l *in* L *to* 1 **do**
  backward-propagation to get $\frac{\partial L}{\partial O^l}$;
end
for l *in* 1 *to* L **do**
  initialize $\Delta^*_{B^l} \leftarrow \frac{B^l_{max}}{2^{k-1}}$;
  generate search spaces of Δ_{A^l} and Δ_{B^l};
  for $r = 1$ *to* $\#Round$ **do**
    search for $\Delta^*_{A^l}$ using Eq. (7);
    search for $\Delta^*_{B^l}$ using Eq. (7);
  end
end

propagation. 2) The second phase is to search for the optimal scaling factors layer by layer. Different scaling factors in the search space are used to quantize the activation values and weight values in the l-th layer. Then the output of the layer $\hat{O}^l$ is calculated. We search for the optimal scaling factor Δ^* that minimizes Eq. (7).

In the first phase, we need to store O^l and $\frac{\partial L}{\partial O^l}$, which consumes a lot of GPU memory. Therefore, we transfer these data to the main memory when they are generated. In the second phase, we transfer O^l and $\frac{\partial L}{\partial O^l}$ back to GPU memory and destroy them when the quantization of l-th layer is finished. To make full use of the GPU parallelism, we calculate $\hat{O}^l$ and the influence on loss for different scaling factors in batches.

4 Experiments

In this section, we first introduce the experimental settings. Then we will evaluate the proposed methods on different vision transformer architectures. At last, we will take an ablation study on the proposed methods.

4.1 Experiment Settings

For post-softmax quantization, the search space of Δ^s_{R1} is $[\frac{1}{2^k}, \frac{1}{2^{k+1}}, ..., \frac{1}{2^{k+10}}]$. The search spaces of scaling factors for weight and other activations are the same as that of base PTQ (Sec. 3.1). We set $alpha = 0$, $beta = 1.2$, and $n = 100$. The search round $\#Round$ is set to 3. We experiment on the ImageNet classification task [19]. We randomly select 32 images from the training dataset as calibration images. The ViT models are provided by timm [26].

We quantize all the weights and inputs for the fully-connect layers including the first projection layer and the last prediction layer. We also quantize the two

Table 1: Top-1 Accuracy of Quantized Vision Transformers. The result in the bracket is the accuracy drop from floating-point networks. W6A6 means weights are quantized to 6-bit and activations are quantized to 6-bit. The default patch size is 16x16. ViT-S/224/32 means the input resolution is 224x224 and the patch size is 32x32.

Model	FP32	Base PTQ		PTQ4ViT	
		W8A8	W6A6	W8A8	W6A6
ViT-S/224/32	75.99	73.61(2.38)	60.14(15.8)	75.58(0.41)	71.90(4.08)
ViT-S/224	81.39	80.46(0.91)	70.24(11.1)	81.00(0.38)	78.63(2.75)
ViT-B/224	84.54	83.89(0.64)	75.66(8.87)	84.25(0.29)	81.65(2.89)
ViT-B/384	86.00	85.35(0.64)	46.88(39.1)	85.82(0.17)	83.34(2.65)
DeiT-S/224	79.80	77.65(2.14)	72.26(7.53)	79.47(0.32)	76.28(3.51)
DeiT-B/224	81.80	80.94(0.85)	78.78(3.01)	81.48(0.31)	80.25(1.55)
DeiT-B/384	83.11	82.33(0.77)	68.44(14.6)	82.97(0.13)	81.55(1.55)
Swin-T/224	81.39	80.96(0.42)	78.45(2.92)	81.24(0.14)	80.47(0.91)
Swin-S/224	83.23	82.75(0.46)	81.74(1.48)	83.10(0.12)	82.38(0.84)
Swin-B/224	85.27	84.79(0.47)	83.35(1.91)	85.14(0.12)	84.01(1.25)
Swin-B/384	86.44	86.16(0.26)	85.22(1.21)	86.39(0.04)	85.38(1.04)

input matrices for the matrix multiplications in self-attention modules. We use different quantization parameters for different self-attention heads. The scaling factors for W^Q, W^K, and W^V are different. The same as [16], we don't quantize softmax and normalization layers in vision transformers.

4.2 Results on ImageNet Classification Task

We choose different vision transformer architectures, including ViT [7], DeiT [22], and Swin [15]. The results are demonstrated in Tab. 1. From this table, we observe that base PTQ results in more than 1% accuracy drop on some vision transformers even at the 8-bit quantization. PTQ4ViT achieves less than 0.5% accuracy drop with 8-bit quantization. For 6-bit quantization, base PTQ results in high accuracy drop (9.8% on average) while PTQ4ViT achieves a much smaller accuracy drop (2.1% on average).

We observe that the accuracy drop on Swin is not as significant as ViT and DeiT. The prediction accuracy drops are less than 0.15% on the four Swin transformers at 8-bit quantization. The reason may be that Swin computes the self-attention locally within non-overlapping windows. It uses a smaller number of patches to calculate the self-attention, reducing the unbalance after post-softmax values. We also observe that larger vision transformers are less sensitive to quantization. For instance, the accuracy drops of ViT-S/224/32, ViT-S/224, ViT-B/224, and ViT-B/384 are 0.41, 0.38, 0.29, and 0.17 at 8-bit quantization and 4.08, 2.75, 2.89, and 2.65 at 6-bit quantization, respectively. The reason may be that the larger networks have more weights and generate more activations, making them more robust to the perturbation caused by quantization.

Table 2: Results of different PTQ methods. #ims means the number of calibration images. MP means mixed precision. BC means bias correction.

Model	Method	Bit-width	#ims	Size	Top-1
DeiT-S/224 79.80	EasyQuant [27]	W8A8	1024	22.0	76.59
	Liu [16]	W8A8	1024	22.0	77.47
	Liu [16]	W8A8 (MP)	1024	22.2	78.09
	PTQ4ViT	W8A8	32	22.0	**79.47**
	EasyQuant [27]	W6A6	1024	16.5	73.26
	Liu [16]	W6A6	1024	16.5	74.58
	Liu [16]	W6A6 (MP)	1024	16.6	75.10
	PTQ4ViT	W6A6	32	16.5	**76.28**
DeiT-B 81.80	EasyQuant [27]	W8A8	1024	86.0	79.36
	Liu [16]	W8A8	1024	86.0	80.48
	Liu [16]	W8A8 (MP)	1024	86.8	81.29
	PTQ4ViT	W8A8	32	86.0	**81.48**
	EasyQuant [27]	W6A6	1024	64.5	75.86
	Liu [16]	W6A6	1024	64.5	77.02
	Liu [16]	W6A6 (MP)	1024	64.3	77.47
	PTQ4ViT	W6A6	32	64.5	**80.25**
	Liu [16]	W4A4 (MP)	1024	43.6	**75.94**
	PTQ4ViT	W4A4	32	43.0	60.91
	PTQ4ViT+BC	W4A4	32	43.0	64.39

Tab. 2 demonstrates the results of different PTQ methods. EasyQuant [27] is a popular post-training method that alternatively searches for the optimal scaling factors of weight and activation. However, the accuracy drop is more than 3% at 8-bit quantization. Liu et al. [16] proposed using the Pearson correlation coefficient and ranking loss are used as the metrics to determine the scaling factors, which increases the Top-1 accuracy. Since the sensitivity of different layers to quantization is not the same, they also use the mixed-precision technique, achieving good results at 4-bit quantization. At 8-bit quantization and 6-bit quantization, PTQ4ViT outperforms other methods, achieving more than 1% improvement in prediction accuracy on average. At 4-bit quantization, the performance of PTQ4ViT is not good. Although bias correction [17] can improve the performance of PTQ4ViT, the result at 4-bit quantization is lower than the mixed-precision of Liu et al. This indicates that mixed-precision is important for quantization with lower bit-width.

4.3 Ablation Study

Next, we take ablation study on the effect of the proposed twin uniform quantization and Hessian guided metric. The experimental results are shown in Tab. 3. As we can see, the proposed methods improve the top-1 accuracy of quantized vi-

Table 3: Ablation study of the effect of the proposed twin uniform quantization and Hessian guided metric. We mark a ✓ if the proposed method is used.

Model	Hessian Guided	Softmax Twin	GELU Twin	Top-1 Accuracy W8A8	W6A6
ViT-S/224 81.39				80.47	70.24
	✓			80.93	77.20
	✓	✓		81.11	78.57
	✓		✓	80.84	76.93
		✓	✓	79.25	74.07
	✓	✓	✓	81.00	78.63
ViT-B/224 84.54				83.90	75.67
	✓			83.97	79.90
	✓	✓		84.07	80.76
	✓		✓	84.10	80.82
		✓	✓	83.40	78.86
	✓	✓	✓	84.25	81.65
ViT-B/384 86.00				85.35	46.89
	✓			85.42	79.99
	✓	✓		85.67	82.01
	✓		✓	85.60	82.21
		✓	✓	84.35	80.86
	✓	✓	✓	85.89	83.19

sion transformers. Specifically, using the Hessian guided metric alone can slightly improve the accuracy at 8-bit quantization, and it significantly improves the accuracy at 6-bit quantization. For instance, on ViT-S/224, the accuracy improvement is 0.46% at 8-bit while it is 6.96% at 6-bit. And using them together can further improve the accuracy.

Based on the Hessian guided metric, using the twin uniform quantization on post-softmax activation or post-GELU activation can improve the performance. We observe that using the twin uniform quantization without the Hessian guided metric significantly decreases the top-1 accuracy. For instance, the top-1 accuracy on ViT-S/224 achieves 81.00% with both Hessian guided metric and twin uniform quantization at 8-bit quantization, while it decreases to 79.25% without Hessian guided metric, which is even lower than basic PTQ with 80.47% top-1 accuracy. This is also evidence that the metric considering only the local information is inaccurate.

5 Conclusion

In this paper, we analyzed the problems of post-training quantization for vision transformers. We observed both the post-softmax activations and the post-GELU activations have special distributions. We also found that the common

quantization metrics are inaccurate to determine the optimal scaling factor. To solve these problems, we proposed the twin uniform quantization and a Hessian-guided metric. They can decrease the quantization error and improve the prediction accuracy at a small cost. To enable the fast quantization of vision transformers, we developed an efficient framework, PTQ4ViT. The experiments demonstrated that we achieved near-lossless prediction accuracy on the ImageNet classification task, making PTQ acceptable for vision transformers.

Acknowledgements This work is supported by National Key R&D Program of China (2020AAA0105200), NSF of China (61832020, 62032001, 92064006), Beijing Academy of Artificial Intelligence (BAAI), and 111 Project (B18001).

6 Appendix

6.1 Derivation of Hessian guided metric

Hessian guided metric introduces as small an increment on task loss $L = CE(\hat{y}, y)$ as possible, in which $\hat{y}$ is the prediction of the quantized model and y is the ground truth. Here y is approximated by the prediction of the floating-point model y_{FP}, since no labels of input data are available in PTQ.

Quantization introduces a small perturbation ϵ on weight W, whose effect on task loss $\mathbb{E}[L(W)]$ could be analyzed with Taylor series expansion,

$$\mathbb{E}[L(\hat{W})] - \mathbb{E}[L(W)] \approx \epsilon^T \bar{g}^{(W)} + \frac{1}{2}\epsilon^T \bar{H}^{(W)} \epsilon. \tag{8}$$

Since the pretrained model has converged to a local optimum, The gradients $\bar{g}^{(W)}$ is close to zero and could be ignored. The Hessian matrix $\bar{H}^{(W)}$ on weight could be computed by

$$\frac{\partial^2 L}{\partial w_i \partial w_j} = \frac{\partial}{\partial w_j}(\sum_{k=1}^{m} \frac{\partial L}{\partial O_k}\frac{\partial O_k}{\partial w_i}) = \sum_{k=1}^{m} \frac{\partial L}{\partial O_k}\frac{\partial^2 O_k}{\partial w_i \partial w_j} + \sum_{k,l=1}^{m} \frac{\partial O_k}{\partial w_i}\frac{\partial^2 L}{\partial O_k \partial O_l}\frac{\partial O_l}{\partial w_j}. \tag{9}$$

$O = W^T X \in R^m$ is the output of the layer, and $\frac{\partial^2 O_k}{\partial w_i \partial w_j} = 0$. So the first term of Eq. (9) is zero, and $\bar{H}^{(W)} = J_O(W)^T \bar{H}^{(O)} J_O(W)$. Therefore, Eq. (8) could be further written as,

$$\mathbb{E}[L(\hat{W})] - \mathbb{E}[L(W)] \approx \frac{1}{2}(J_O(W)\epsilon)^T \bar{H}^{(O)} J_O(W)\epsilon \approx \frac{1}{2}(\hat{O} - O)^T \bar{H}^{(O)}(\hat{O} - O) \tag{10}$$

Following Liu et al.[14], we use the Diagonal Fisher Information Matrix to substitute $\bar{H}^{(O)}$. The optimization is formulated as:

$$\min_{\Delta_W} \mathbb{E}[(\hat{O} - O)^T \mathrm{diag}((\frac{\partial L}{\partial O_1})^2, \cdots, (\frac{\partial L}{\partial O_m})^2)(\hat{O} - O)]. \tag{11}$$

References

1. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 7948–7956 (2019), https://proceedings.neurips.cc/paper/2019/hash/c0a62e133894cdce435bcb4a5df1db2d-Abstract.html 2, 5
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html 1
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12346, pp. 213–229. Springer (2020). https://doi.org/10.1007/978-3-030-58452-8_13, https://doi.org/10.1007/978-3-030-58452-8_13 3
4. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I., Srinivasan, V., Gopalakrishnan, K.: PACT: parameterized clipping activation for quantized neural networks. CoRR **abs/1805.06085** (2018), http://arxiv.org/abs/1805.06085 2, 5
5. Choukroun, Y., Kravchik, E., Yang, F., Kisilev, P.: Low-bit quantization of neural networks for efficient inference. In: 2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019. pp. 3009–3018. IEEE (2019). https://doi.org/10.1109/ICCVW.2019.00363, https://doi.org/10.1109/ICCVW.2019.00363 2, 5, 8
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423 1
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=YicbFdNTTy 1, 3, 11
8. Fang, J., Shafiee, A., Abdel-Aziz, H., Thorsley, D., Georgiadis, G., Hassoun, J.: Post-training piecewise linear quantization for deep neural networks. In: ECCV (2020) 4

9. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. CoRR **abs/2103.13630** (2021), https://arxiv.org/abs/2103.13630 2
10. Guo, Y., Yao, A., Zhao, H., Chen, Y.: Network sketching: Exploiting binary structure in deep cnns. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 4040–4048. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.430, http://doi.ieeecomputersociety.org/10.1109/CVPR.2017.430 7
11. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D.: A survey on visual transformer. CoRR **abs/2012.12556** (2020), https://arxiv.org/abs/2012.12556 1, 4
12. Huang, L., Tan, J., Liu, J., Yuan, J.: Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXV. Lecture Notes in Computer Science, vol. 12370, pp. 17–33. Springer (2020). https://doi.org/10.1007/978-3-030-58595-2_2, https://doi.org/10.1007/978-3-030-58595-2_2 4
13. Jia, D., Han, K., Wang, Y., Tang, Y., Guo, J., Zhang, C., Tao, D.: Efficient vision transformers via fine-grained manifold distillation. CoRR **abs/2107.01378** (2021), https://arxiv.org/abs/2107.01378 4
14. Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., Gu, S.: BRECQ: pushing the limit of post-training quantization by block reconstruction. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=POWv6hDd9XH 9, 14
15. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. CoRR **abs/2103.14030** (2021), https://arxiv.org/abs/2103.14030 1, 4, 11
16. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer pp. 28092–28103 (2021), https://proceedings.neurips.cc/paper/2021/hash/ec8956637a99787bd197eacd77acce5e-Abstract.html 2, 4, 5, 8, 11, 12
17. Nagel, M., van Baalen, M., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 1325–1334. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00141, https://doi.org/10.1109/ICCV.2019.00141 12
18. Prato, G., Charlaix, E., Rezagholizadeh, M.: Fully quantized transformer for machine translation. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020. Findings of ACL, vol. EMNLP 2020, pp. 1–14. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.1, https://doi.org/10.18653/v1/2020.findings-emnlp.1 5
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y, https://doi.org/10.1007/s11263-015-0816-y 10

20. Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Q-BERT: hessian based ultra low precision quantization of BERT. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 8815–8821. AAAI Press (2020), https://aaai.org/ojs/index.php/AAAI/article/view/6409 5
21. Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Tao, D.: Patch slimming for efficient vision transformers. CoRR **abs/2106.02852** (2021), https://arxiv.org/abs/2106.02852 4
22. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (2021), http://proceedings.mlr.press/v139/touvron21a.html 11
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017), https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html 1, 3
24. Wang, H., Zhu, Y., Adam, H., Yuille, A.L., Chen, L.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 5463–5474. Computer Vision Foundation / IEEE (2021), https://openaccess.thecvf.com/content/CVPR2021/html/Wang_MaX-DeepLab_End-to-End_Panoptic_Segmentation_With_Mask_Transformers_CVPR_2021_paper.html 4
25. Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. CoRR **abs/2102.12122** (2021), https://arxiv.org/abs/2102.12122 4
26. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861 10
27. Wu, D., Tang, Q., Zhao, Y., Zhang, M., Fu, Y., Zhang, D.: Easyquant: Post-training quantization via scale optimization. CoRR **abs/2006.16669** (2020), https://arxiv.org/abs/2006.16669 5, 6, 8, 12
28. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII. Lecture Notes in Computer Science, vol. 11212, pp. 373–390. Springer (2018). https://doi.org/10.1007/978-3-030-01237-3_23, https://doi.org/10.1007/978-3-030-01237-3_23 2, 5