APHQ-ViT: Post-Training Quantization with Average Perturbation Hessian Based Reconstruction for Vision Transformers

Zhuguanyu Wu[1,2], Jiayi Zhang[1,2], Jiaxin Chen[1,2]✉, Jinyang Guo[3], Di Huang[2], Yunhong Wang[1,2]✉
[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
[2]School of Computer Science and Engineering, Beihang University, Beijing, China
[3]School of Artificial Intelligence, Beihang University, Beijing, China
{goatwu, zhangjyi, jiaxinchen, jinyangguo, dhuang, yhwang}@buaa.edu.cn

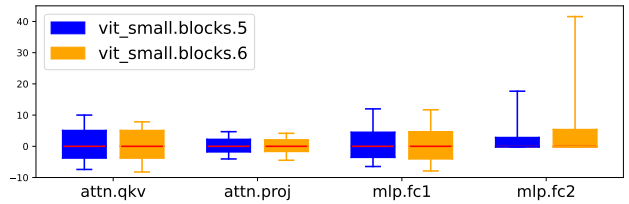
Abstract

Vision Transformers (ViTs) have become one of the most commonly used backbones for vision tasks. Despite their remarkable performance, they often suffer significant accuracy drops when quantized for practical deployment, particularly by post-training quantization (PTQ) under ultra-low bits. Recently, reconstruction-based PTQ methods have shown promising performance in quantizing Convolutional Neural Networks (CNNs). However, they fail when applied to ViTs, primarily due to the inaccurate estimation of output importance and the substantial accuracy degradation in quantizing post-GELU activations. To address these issues, we propose ***APHQ-ViT****, a novel PTQ approach based on importance estimation with Average Perturbation Hessian (APH). Specifically, we first thoroughly analyze the current approximation approaches with Hessian loss, and propose an improved average perturbation Hessian loss. To deal with the quantization of the post-GELU activations, we design an MLP Reconstruction (MR) method by replacing the GELU function in MLP with ReLU and reconstructing it by the APH loss on a small unlabeled calibration set. Extensive experiments demonstrate that APHQ-ViT using linear quantizers outperforms existing PTQ methods by substantial margins in 3-bit and 4-bit across different vision tasks. The source code is available at* https://github.com/GoatWu/APHQ-ViT.

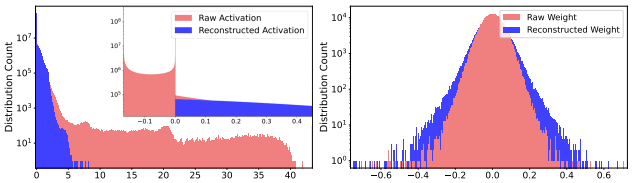
1. Introduction

The success of Transformer-based models in natural language processing (NLP) [7, 45] has inspired their application to various computer vision tasks, such as image classification [3, 11, 32], object detection [2, 6, 53, 57] and instance segmentation [42, 48, 51, 52, 56]. Due to their sophisticated architectures for representation learning, substantial memory usage and computational overhead make it a great challenge to deploy these models on resource-constrained devices [23].

Model quantization has recently emerged as a promising solution to reduce the computational cost of deep learning models. This technique converts the weights or activations from float-point precision to low bit-width, while preserving the original model architectures. Most current quantization approaches are generally categorized into two groups: quantization-aware training (QAT) [5, 12] and post-training quantization (PTQ) [15, 40]. QAT methods typically achieve superior accuracy compared to PTQ by performing end-to-end training on the full pretraining dataset. Nevertheless, they are often time-intensive and encounter substantial limitations when the original dataset is inaccessible. In contrast, PTQ methods are more applicable as they rely solely on a small unlabeled calibration dataset instead of requiring



(a) activation statistics of linear layers



(b) activation distribution of *mlp.fc2* (c) weight distribution of *mlp.fc2*

Figure 1. (a) shows the box plot of activations for each linear layer in *vit-small.blocks.5 and 6* by using the 0.99 quantile, highlighting the varying ranges of post-GELU activations. (b) and (c) display that the activation range of *fc2* is significantly reduced after MLP Reconstruction, while the weight range only exhibits slight changes.

✉ Corresponding Authors

access to the full training set. PTQ methods can be further divided into two categories, *i.e.*, the one that only involves calibration [27, 29, 35, 50], and the reconstruction-based one [20, 36, 49, 54], where the later generally achieves superior accuracy by introducing an efficient fine-tuning process. Despite their promising performance in quantizing CNNs [25, 30, 46], the reconstruction-based methods suffer from the following two limitations when applied to ViTs.

1) *Inaccurate estimation of output importance.* Representative reconstruction-based PTQ methods employ the block-reconstruction framework [25, 46], which fine-tunes the AdaRound [39] weights to ensure that the output of the quantized block closely matches the output of the original full-precision one. The mean squared error (MSE) between the quantized and original outputs is one of the most commonly used metrics to evaluate quantization quality. However, this approach is suboptimal, since it treats all output tokens and dimensions equally, overlooking the critical importance of the class token and the importance variations across channels in ViTs as shown in Sec. C of the *supplementary material*. Some works leverage the Hessian matrix based on Fisher information to explore the distinct importance [8, 25, 50], while fail to surpass the MSE loss due to the inaccurate approximation on the Hessian matrix.

2) *Performance degradation in quantizing post-GELU activation.* As shown in Fig. 1 (a), the quantization error in post-GELU activations stems from two primary factors. First, the activation distribution is highly imbalanced: negative activations are densely concentrated within a narrow interval $[-0.17, 0]$, while positive activations follow sparse distributions. Second, the activation range varies significantly, reaching up to 40 in certain layers. Some works have attempted to deal with the imbalanced activation distribution by a twin-uniform quantizer that employs separate scaling factors for positive and negative activations [50], or employ a hardware-friendly logarithmic quantizer with an arbitrary base [47]. However, they necessitate specialized hardware support for the quantizer, limiting their practicality in real-world applications.

To address the above issues, we propose a novel quantization approach dubbed APHQ-ViT for the post-training quantization of Vision Transformers. As illustrated in Fig. 2, to tackle the *inaccurate estimation of output importance*, we thoroughly investigate the current approximation methods with Hessian loss, and propose an improved average perturbation Hessian (APH) loss for block reconstruction. We show that applying APH to explore the importance of output can stabilize the reconstruction process, and further promote precision. To deal with *performance degradation in quantizing post-GELU activations*, we develop the MLP Reconstruction method (MR), by replacing the GELU activation function with ReLU. As shown in Fig. 1(b) and (c), MR not only reduces the activation range while maintaining the weight range, but also alleviates the imbalanced activation distributions, thus reducing the quantization error.

The main contributions of our work lie in three-fold:

1) We thoroughly analyze the limitations of existing Hessian guide quantization loss, and propose an improved Average Perturbation Hessian (APH) loss by mitigating the estimation deviations, which facilitates both the block-wise quantization reconstruction and MLP reconstruction.

2) We develop a novel MLP Reconstruction (MR) method by replacing the GELU activation function in MLP with ReLU, which simultaneously alleviates imbalanced activation distribution and significantly reduces the activation range, making the model more amenable to quantization.

3) We extensively conduct experiments on public datasets across various vision tasks in order to evaluate the performance of our method. Experimental results demonstrate that the proposed method, utilizing only linear quantizers, significantly outperforms the current state-of-the-art approaches with distinct Vision Transformer architectures, especially in the case of ultra-low bit quantization.

2. Related Work

Model quantization, which aims to map the floating-point weights and activations to lower bit widths, has become one of the most widely used techniques for accelerating the inference of deep learning models. It can be roughly divided into two categories: Post-Training Quantization (PTQ) and Quantization Aware Training (QAT). Among the quantization methods for Vision Transformers, QAT methods [5, 12, 18, 26] often achieve higher accuracy. However, QAT methods often require a large amount of training resources, limiting their universality. By contrast, PTQ methods only take a small calibration dataset to adjust quantization parameters, making them resource-efficient.

The PTQ methods can be further categorized into two groups, *i.e.*, the calibration-only methods that solely involve the calibration stage, and the reconstruction-based methods that additionally incorporate a reconstruction stage.

Calibration-only methods can efficiently obtain a quantized model. PTQ4ViT [50] employs a twin-uniform quantizer to reduce the activation quantization error, and adopts a Hessian guided loss to evaluate the effectiveness of different scaling factors. RepQ-ViT [27] decouples the quantization and inference processes, specifically addressing post-LayerNorm activations with significant inter-channel variations. NoisyQuant [31] reduces quantization error by adding a fixed uniform noisy bias to the values being quantized. IGQ-ViT [38] employs a group-wise activation quantizer to balance the inference efficiency and quantization accuracy. ERQ [55] introduces the GPTQ approach [14] to ViTs and proposes an activation quantization error reduction module to mitigate quantization errors, along with a derived proxy for output error to refine weight rounding. AdaLog [47] designs

a hardware-friendly arbitrary-base logarithmic quantizer to handle power-law activations and a progressive hyperparameter search algorithm. However, these methods still suffer substantial quantization loss under low-bit quantization.

Reconstruction-based methods often achieve quantized models with higher accuracy, by additionally employing quantization reconstruction. Numerous approaches have been developed for CNNs. AdaRound [39] adopts a refined weight rounding strategy to minimize the task loss, outperforming conventional rounding-to-nearest methods. BRECQ [25] improves performance by leveraging cross-layer dependencies through block-wise reconstruction. QDrop [46] employs random activation dropout during block reconstruction, facilitating obtaining smoother optimized weight distributions. Although effective for CNNs, these methods yield suboptimal results when applied to ViTs. I&S-ViT [54] employs a three-stage smooth optimization strategy to address the quantization inefficiency and ensure stable learning. DopQ-ViT selects optimal scaling factors to mitigate the impact of outliers and preserve quantization performance. OASQ [36] addresses outlier activations employing distinct granularities in the quantization reconstruction. Although these methods generally outperform calibration-only approaches, they still struggle to reach an acceptable performance under ultra-low bit quantization.

3. The Proposed Approach

As shown in Fig. 2, the proposed APHQ-ViT approach follows the block-wise quantization pipeline. In each block, we first perform MLP Reconstruction, followed by quantization reconstruction based on QDrop. The average perturbation Hessian loss is applied in both reconstructions to explore the distinct output importance. The overall pipeline of APHQ-ViT is summarized in Algorithm 1. The average perturbation Hessian loss and MLP Reconstruction are described in Sec. 3.2 and Sec. 3.3, respectively.

3.1. Preliminaries: Hessian in BRECQ

The Hessian guided metric proposed by BRECQ [25] stands out as one of the most prevalent metrics for evaluating the quantization quality of CNNs. It assumes that the de-quantized weight $\widehat{\boldsymbol{W}}$ can be represented as the original weight $\boldsymbol{W}$ perturbed by $\boldsymbol{\epsilon}$, *i.e.*, $\widehat{\boldsymbol{W}} = \boldsymbol{W} + \boldsymbol{\epsilon}$. The quality of quantization is measured by estimating the quantization loss through a Taylor expansion:

$$\mathbb{E}[\mathcal{L}(\widehat{\boldsymbol{W}})] - \mathbb{E}[\mathcal{L}(\boldsymbol{W})] \approx \boldsymbol{\epsilon}^\top \bar{\boldsymbol{J}}^{(\boldsymbol{W})} + \frac{1}{2}\boldsymbol{\epsilon}^\top \bar{\boldsymbol{H}}^{(\boldsymbol{W})}\boldsymbol{\epsilon}, \quad (1)$$

where $\bar{\boldsymbol{J}}^{(\boldsymbol{W})}$ and $\bar{\boldsymbol{H}}^{(\boldsymbol{W})}$ are the Jacobian and Hessian matrices w.r.t the weight $\boldsymbol{W}$, respectively.

Supposing the convergence of a pre-trained model to be quantized, existing works often drop the first-order term $\boldsymbol{\epsilon}^\top \bar{\boldsymbol{J}}^{(W)}$ and approximate the Hessian matrix with squared gradient, resulting in the quantization loss:

$$\mathbb{E}[\mathcal{L}(\widehat{\boldsymbol{W}})] - \mathbb{E}[\mathcal{L}(\boldsymbol{W})] \approx \sum_i \left((\widehat{\boldsymbol{O}}_i - \boldsymbol{O}_i) \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{O}_i} \right)^2, \quad (2)$$

where $\boldsymbol{O}$ is the output, and $\widehat{\boldsymbol{O}}$ is the de-quantized one of $\boldsymbol{O}$.

The above Hessian guided quantization loss adopts two approximations as in BRECQ: 1) the Hessian matrix is approximated by the Fisher Information Matrix (FIM) [13]; 2) the diagonal elements of FIM are approximated by the squared gradients w.r.t. the output. These approximations achieve high accuracy, when the task loss is the Cross-Entropy (CE) loss, and the model's predicted distribution aligns closely with the true data distribution. However, in practice, models are often unable to fit the true data distribution well, leading to inevitable approximation errors. Additionally, these approximations fail to generalize to tasks such as segmentation and object detection. As a consequence, when applied to ViTs, the loss in Eq. (2) is inferior to the MSE Loss in many ViT architectures as shown in Table 4.

Algorithm 1 APHQ-ViT for Block-wise Quantization.

Input: The full-precision model $\mathcal{M}$, the full-precision block $\mathcal{B}$ to be quantized, the calibration data $\mathcal{D}_{calib}$, and the loss function $\mathcal{L}$.

Calculate the Average Perturbation Hessian:

Compute the raw output $\boldsymbol{O}$ of $\mathcal{B}$ based on $\mathcal{D}_{calib}$.
Calculate the perturbed outputs $\boldsymbol{O}^+$ and $\boldsymbol{O}^-$.
Compute $f(\boldsymbol{O})$/$f(\boldsymbol{O}^+)$/$f(\boldsymbol{O}^-)$ by forward passing $\boldsymbol{O}$/$\boldsymbol{O}^+$/$\boldsymbol{O}^-$ through the remaining blocks of $\mathcal{M}$.
Calculate $\mathcal{L}(f(\boldsymbol{O}), f(\boldsymbol{O}^+))$ and $\mathcal{L}(f(\boldsymbol{O}), f(\boldsymbol{O}^-))$ and obtain $\bar{\boldsymbol{J}}^{(\boldsymbol{O}^+)}$ and $\bar{\boldsymbol{J}}^{(\boldsymbol{O}^-)}$ by backward propagation.
Calculate the average perturbation Hessian matrix $\bar{\boldsymbol{H}}$ based on Eq. (8).

MLP Reconstruction:

Replace the GELU activation of MLP by ReLU.
for $i = 0, \cdots, \text{max\_iter}$ **do**
    Calculate $\boldsymbol{O}_{\text{Direct}}$ and $\boldsymbol{O}_{\text{Clamp}}$ by Eqs. (11) and (12).
    Calculate $\mathcal{L}_{\text{Distill}}$ by Eq. (14).
    Perform backward propagation and update MLP.
end for

Quantization Reconstruction:

for $i = 0, \cdots, \text{max\_iter}$ **do**
    Calculate the quantized output $\widehat{O}$ by QDrop [46].
    Calculate $\mathcal{L}_{\text{APH}}$ based on Eq. (9).
    Perform backward propagation and update the AdaRound [39] weights in $\mathcal{B}$.
end for

Output: The quantized block $\widehat{\mathcal{B}}$.

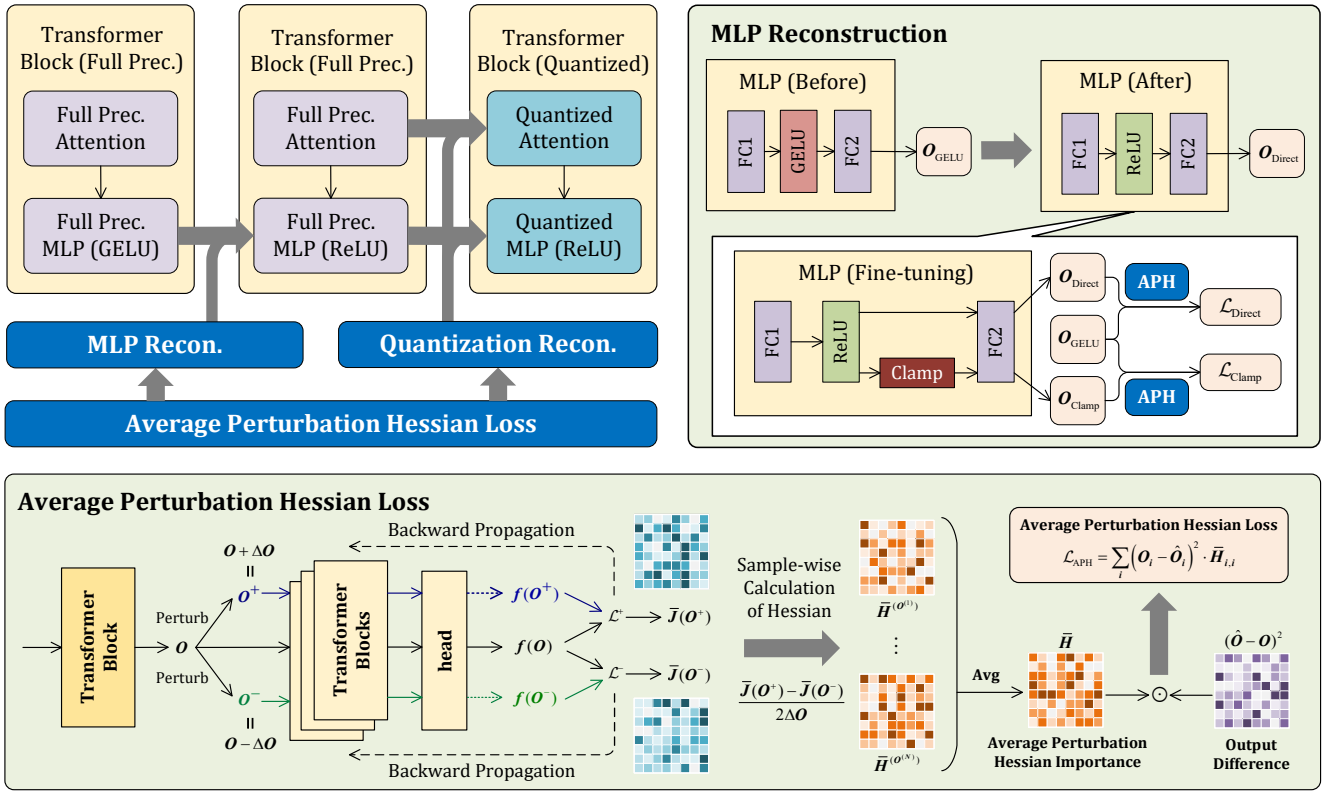


Figure 2. Framework overview of APHQ-ViT. In the block-wise quantization process, we first reconstruct the MLP layer, followed by quantization reconstruction, both of which are optimized by the proposed Average Perturbation Hessian (APH) loss. The MLP Reconstruction (MR) method replaces the GELU activation function with ReLU and reduces the post-GELU activation range. The detailed implementation of the APH loss is visualized at the bottom.

3.2. Average Perturbation Hessian Loss

To address the limitations of the Hessian guided loss in Eq. (2), we develop a perturbation based estimation method that only relies on two fundamental assumptions as below:

A.1) When performing Taylor series expansions for the loss, the third and higher-order derivatives can be omitted without significantly sacrificing the accuracy [10, 37, 39].

A.2) The influence of individual elements on the final output is assumed to be independent, allowing the use of the diagonal Hessian as a practical substitute for the computationally intensive full Hessian [10, 43].

It is worth noting that A.1) and A.2) are widely used in various model compression methods, including BRECQ, which further rely on additional, stronger assumptions to achieve their results.

We first extend the loss function to ensure compatibility across diverse tasks. Instead of the conventional CE loss, we regard quantization as a knowledge distillation process on a small unlabeled calibration dataset. This allows us to directly employ distillation loss to address different tasks. Specifically, for classification, we adopt the KL divergence between the output logits as the distillation loss. For two-stage object detection and instance segmentation, we combine the KL divergence from the classification head and the smooth L1 distance [16] from the regression head as the distillation loss. Compared to the CE Loss, these distillation losses share the following common characteristics: $\mathcal{L}(\hat{\boldsymbol{O}}, \boldsymbol{O}) \geq 0$, and $\mathcal{L}(\hat{\boldsymbol{O}}, \boldsymbol{O}) = 0$ if and only if $\boldsymbol{O} = \hat{\boldsymbol{O}}$. According to the extreme value theorem [21], if $\mathcal{L}$ is differentiable at $\hat{\boldsymbol{O}} = \boldsymbol{O}$, then we have:

$$\bar{\boldsymbol{J}}^{(\boldsymbol{O})} = \left.\frac{\partial \mathcal{L}(\hat{\boldsymbol{O}}, \boldsymbol{O})}{\partial \hat{\boldsymbol{O}}}\right|_{\hat{\boldsymbol{O}}=\boldsymbol{O}} = 0. \tag{3}$$

Based on Eq. (3), we treat the errors introduced by quantization or MLP Reconstruction as small perturbations denoted by $\boldsymbol{\epsilon}$, and perform a Taylor expansion as below:

$$\begin{aligned}\mathcal{L}(\boldsymbol{O}+\boldsymbol{\epsilon}) &= \mathcal{L}(\boldsymbol{O}) + \boldsymbol{\epsilon}^\top \bar{\boldsymbol{J}}^{(\boldsymbol{O})} + \frac{1}{2}\boldsymbol{\epsilon}^\top \bar{\boldsymbol{H}}^{(\boldsymbol{O})}\boldsymbol{\epsilon} + O(\|\boldsymbol{\epsilon}\|^3) \\ &= \frac{1}{2}\boldsymbol{\epsilon}^\top \bar{\boldsymbol{H}}^{(\boldsymbol{O})}\boldsymbol{\epsilon} + O(\|\boldsymbol{\epsilon}\|^3) \approx \frac{1}{2}\boldsymbol{\epsilon}^\top \bar{\boldsymbol{H}}^{(\boldsymbol{O})}\boldsymbol{\epsilon},\end{aligned} \tag{4}$$

where $\mathcal{L}(\boldsymbol{O}+\boldsymbol{\epsilon}), \mathcal{L}(\boldsymbol{O})$ are the abbreviations of $\mathcal{L}(\boldsymbol{O}+\boldsymbol{\epsilon}, \boldsymbol{O})$ and $\mathcal{L}(\boldsymbol{O}, \boldsymbol{O})$, respectively, $\bar{\boldsymbol{H}}^{(\boldsymbol{O})}$ is the Hessian matrix of

$\mathcal{L}$ w.r.t $\boldsymbol{O}$, and $O(\|\boldsymbol{\epsilon}\|^3)$ represents the sum of the third and higher-order derivatives. As depicted in Eq. (3), $\mathcal{L}(\boldsymbol{O})$ and $\bar{\boldsymbol{J}}^{(\boldsymbol{O})}$ are zeros, and $O(\|\boldsymbol{\epsilon}\|^3)$ is omitted according to A.1).

Based on A.2), we follow BRECQ by utilizing the block-diagonal Hessian and disregarding the inter-block dependencies. By definition, the diagonal elements of the Hessian matrix are the second partial derivatives of the loss function:

$$\bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O})} = \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{O}_i^2} = \frac{\partial}{\boldsymbol{O}_i}\left(\frac{\partial \mathcal{L}}{\boldsymbol{O}_i}\right). \tag{5}$$

For the $i-$th diagonal element, we perturb $\boldsymbol{O}$ by $\Delta \boldsymbol{O} = 10^{-6}$: $\boldsymbol{O}^+ = \boldsymbol{O} + \Delta \boldsymbol{O} \cdot \mathbf{1}$ and $\boldsymbol{O}^- = \boldsymbol{O} - \Delta \boldsymbol{O} \cdot \mathbf{1}$, where $\mathbf{1}$ equals 1 for all elements. Based on the mean value theorem [21], there exists an $\boldsymbol{O}'$ between $\boldsymbol{O}^-$ and $\boldsymbol{O}^+$ such that

$$\bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O}')} = \frac{\bar{\boldsymbol{J}}_i^{(\boldsymbol{O}^+)} - \bar{\boldsymbol{J}}_i^{(\boldsymbol{O}^-)}}{2 \cdot \Delta \boldsymbol{O}_i}, \tag{6}$$

where $\bar{\boldsymbol{J}}^{(\boldsymbol{O}^+)}$ and $\bar{\boldsymbol{J}}^{(\boldsymbol{O}^-)}$ are the Jacobian matrices at $\boldsymbol{O}^+$ and $\boldsymbol{O}^-$, which are computed through backward-propagation. As the perturbation $\Delta \boldsymbol{O}$ is small enough, we approximate $\bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O})}$ by $\bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O}')}$. The perturbation Hessian loss is thus formulated as below:

$$\mathcal{L}_{\mathrm{PH}} = \sum_i \left(\widehat{\boldsymbol{O}}_i - \boldsymbol{O}_i\right)^2 \cdot \bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O})}. \tag{7}$$

It is worth noting that using distinct Hessians for different samples may lead to an unstable training process. To address this issue, we compute the average Hessian across all samples and utilize the mean value to formulate the final reconstruction loss as below:

$$\bar{\boldsymbol{H}}_{i,i} = \frac{1}{N} \sum_{n=1}^{N} \bar{\boldsymbol{H}}_{i,i}^{(\boldsymbol{O}^{(n)})}, \tag{8}$$

$$\mathcal{L}_{\mathrm{APH}} = \sum_i \left(\widehat{\boldsymbol{O}}_i - \boldsymbol{O}_i\right)^2 \cdot \bar{\boldsymbol{H}}_{i,i}, \tag{9}$$

where $\boldsymbol{O}^{(n)}$ is the output of the $n-$th sample, and N is the sample size.

Ideally, $\mathcal{L}_{\mathrm{APH}}$ and $\mathcal{L}_{\mathrm{PH}}$ have the following properties.

Theorem 3.1. *The expectation of the APH loss is consistent with that of the PH loss,* i.e., $\mathbb{E}[\mathcal{L}_{\mathrm{APH}}] = \mathbb{E}[\mathcal{L}_{\mathrm{PH}}]$, *under certain independence assumptions.*

Theorem 3.2. *When utilizing mini-batch gradient descent, the variance of the gradient of the quantization parameter θ w.r.t. the APH loss is smaller than that of the PH loss under certain independence assumptions:*

$$\mathrm{Var}\left[\frac{\partial \mathcal{L}_{\mathrm{APH}}}{\partial \theta}\right] \leq \mathrm{Var}\left[\frac{\partial \mathcal{L}_{\mathrm{PH}}}{\partial \theta}\right]. \tag{10}$$

We refer to Sec. A.1 and Sec. A.2 of the *supplementary material* for detailed proof. Theorems 3.1 and 3.2 imply that the gradient of the APH loss is an unbiased estimation on that of the PH loss, while effectively reducing its variance, under certain independence assumptions. As claimed in [22, 24], lower gradient variance results in faster convergence and improved training stability. Therefore, our proposed APH loss is expected to outperform the PH loss.

Compared to the Hessian in BRECQ, our method only requires one additional forward and backward pass, while maintaining the same training complexity. As a result, the extra computational overhead is negligible. The key advantages of our method lie in two-fold: 1) APH is deduced directly from the definition, thus eliminating errors introduced by the Fisher Information Matrix; 2) APH is theoretically generalizable to other tasks besides classification, such as object detection and segmentation.

3.3. MLP Reconstruction

As depicted in Sec. 1, quantizing post-GELU activations in ViTs incurs two significant challenges: 1) the post-GELU activation distribution is highly imbalanced, *i.e.*, concentrating within the narrow interval (-0.17, 0], which leads to approximation errors during quantization [50]. 2) the activation range of post-GELU activations varies substantially.

In this section, we propose an MLP Reconstruction method to address the above two issues simultaneously. To deal with the imbalanced distribution, we replace all GELU activation functions in MLP with ReLU. Subsequently, we perform the feature knowledge distillation [19], and reconstruct MLP individually. Specifically, for each MLP, we obtain its original input and output using the unlabeled data. By following Sec. 3.2, we compute the average perturbation Hessian to determine the output importance. Thereafter, we replace the MLP activation function with ReLU and utilize the Hessian importance to calculate the weighted L_2 distance between the output with ReLU and the original one with GELU, formulated as below:

$$\mathcal{L}_{\mathrm{Direct}} = \left(\boldsymbol{O}_{\mathrm{GELU}} - \boldsymbol{O}_{\mathrm{Direct}}\right)^2 \odot \bar{\boldsymbol{H}}, \tag{11}$$

where $\boldsymbol{O}_{\mathrm{Direct}} = \mathrm{FC2}(\mathrm{ReLU}(\mathrm{FC1}(\boldsymbol{X})))$ is the output of reconstructed MLP with ReLU for input $\boldsymbol{X}$, $\boldsymbol{O}_{\mathrm{GELU}}$ is the output of the original MLP with GELU, and $\bar{\boldsymbol{H}}$ is the average perturbation Hessian.

The reason ReLU can be used as a replacement for GELU lies in the fact that, in deeper Transformers, ReLU may suffer from the dying ReLU problem [33], which is why GELU is typically used during training. However, as described in [34], neural networks with ReLU activations also theoretically possess universal approximation capabilities. In this paper, the MLP module are reconstructed individually for each layer, which is of shallow depth, thus avoiding the dying ReLU problem. This enables the network to achieve

Table 1. Comparison of the top-1 accuracy (%) on the ImageNet dataset with different quantization bit-widths. Here 'Opt.' means whether or not using an optimize-based PTQ method, 'PSQ' refers to 'Post-Softmax Quantizer', and 'PGQ' refers to 'Post-GELU Quantizer'. '*' indicates that the results are reproduced by using the official code. 'TUQ', 'MPQ', 'GUQ', 'SULQ', and 'TanQ' are the abbreviations of 'Twin-Uniform Quantizer' in PTQ4ViT, 'Matthew-effect Preserving Quantizer' in APQ-ViT, 'Groupwise Uniform Quantizer' in IGQ-ViT, 'Shift-Uniform-Log2 Quantizer' in I&S-ViT, and 'Tangent Quantizer' in DopQ-ViT, respectively.

Method	Opt.	PSQ	PGQ	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Prec.	-	-	-	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
PTQ4ViT [50]	×	TUQ	TUQ	3/3	0.10	0.10	3.50	0.10	31.06	28.69	20.13
RepQ-ViT [27]	×	$\log\sqrt{2}$	Uniform	3/3	0.10	0.10	0.10	0.10	0.10	0.10	0.10
AdaLog [47]	×	AdaLog	AdaLog	3/3	13.88	37.91	31.56	24.47	57.47	64.41	69.75
I&S-ViT [54]	✓	SULQ	Uniform	3/3	45.16	63.77	41.52	55.78	73.30	74.20	69.30
DopQ-ViT [49]	✓	TanQ	Uniform	3/3	54.72	65.76	44.71	59.26	74.91	74.77	69.63
QDrop* [46]	✓	Uniform	Uniform	3/3	38.31	73.79	46.69	52.55	74.32	74.11	75.28
APHQ-ViT(Ours)	✓	Uniform	Uniform	3/3	**63.17**	**76.31**	**55.42**	**68.76**	**76.31**	**76.10**	**78.14**
PTQ4ViT [50]	×	TUQ	TUQ	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT [8]	×	MPQ	Uniform	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
RepQ-ViT [27]	×	$\log\sqrt{2}$	Uniform	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
ERQ [55]	×	$\log\sqrt{2}$	Uniform	4/4	68.91	76.63	60.29	72.56	78.23	80.74	82.44
IGQ-ViT [38]	×	GUQ	GUQ	4/4	73.61	79.32	62.45	74.66	79.23	80.98	83.14
AdaLog [47]	×	AdaLog	AdaLog	4/4	72.75	79.68	63.52	72.06	78.03	80.77	82.47
I&S-ViT [54]	✓	SULQ	Uniform	4/4	74.87	80.07	65.21	75.81	79.97	81.17	82.60
DopQ-ViT [49]	✓	TanQ	Uniform	4/4	75.69	80.95	65.54	75.84	80.13	81.71	83.34
QDrop* [46]	✓	Uniform	Uniform	4/4	67.62	82.02	64.95	74.73	79.64	81.03	82.79
OASQ [36]	✓	Unifrom	Uniform	4/4	72.88	76.59	66.31	76.00	78.83	81.02	82.46
APHQ-ViT(Ours)	✓	Uniform	Uniform	4/4	**76.07**	**82.41**	**66.66**	**76.40**	**80.21**	**81.81**	**83.42**

expressive capability comparable to that by using GELU.

To address the activation range issue, we design an alternative clamp loss to constrain the range effectively. Specifically, we compute the p-th percentile of all positive values and restrict the activations within this p-th percentile. The clipped output is formulated as:

$$\begin{aligned} \boldsymbol{A}_{\mathrm{FC2}} &= \mathrm{ReLU}(\mathrm{FC1}(\boldsymbol{X})), \\ \boldsymbol{O}_{\mathrm{clamp}} &= \mathrm{FC2}(\mathrm{clamp}(\boldsymbol{A}_{\mathrm{FC2}},\ \mathrm{Quantile}_p(\boldsymbol{A}_{\mathrm{FC2}}))). \end{aligned} \tag{12}$$

Accordingly, the clipped reconstruction loss is written as:

$$\mathcal{L}_{\mathrm{Clamp}} = (\boldsymbol{O}_{\mathrm{GELU}} - \boldsymbol{O}_{\mathrm{clamp}})^2 \odot \boldsymbol{H}. \tag{13}$$

The MLP Reconstruction loss is finally formulated as:

$$\mathcal{L}_{\mathrm{Distill}} = \mathcal{L}_{\mathrm{Direct}} + \alpha \cdot \mathcal{L}_{\mathrm{Clamp}}, \tag{14}$$

where α is a trade-off hyperparameter fixed as $\alpha = 2$. It is important to note that $\mathcal{L}_{\mathrm{Direct}}$ cannot be omitted. By solely using $\mathcal{L}_{\mathrm{Clamp}}$ leads to vanishing gradients for hard-clipped activations. In regions where activations are hard-clipped, the gradients tend to be zero, hindering the effective update of the MLP parameters. By incorporating $\mathcal{L}_{\mathrm{Direct}}$ that leverages unclamped activations, the gradient vanishing issue is mitigated, thus facilitating effective learning.

4. Experimental Results and Analysis

4.1. Experimental Setup

Datasets and Models. For the classification task, we evaluate our method on ImageNet [41] with representative Vision Transformer architectures, including ViT [11], DeiT [44] and Swin [32]. For object detection and instance segmentation, we evaluate on COCO [28] by utilizing the Mask R-CNN [17] and Cascade Mask R-CNN [1] frameworks based on the Swin backbones.

Implementation Details. All pretrained full-precision Vision Transformers are obtained from the timm library[1]. The pretrained detection and segmentation models are obtained from MMDetection [4]. Following existing works [25, 36, 46, 54], we randomly select 1024 unlabeled images from ImageNet and 256 unlabeled images from COCO as the calibration sets for classification and object detection, respectively. We adopt channel-wise uniform quantizers for weight quantization and layer-wise uniform quantizers for activation quantization, including the attention map. We follow the hyper-parameter settings as used in QDrop [46] by setting the batch size, learning rate for activation quantization, learning rate for tuning weight, the maximal iteration

[1]https://github.com/huggingface/pytorch-image-models

Table 2. Quantization results (%) on COCO for the object detection and instance segmentation tasks. Here, 'Baseline' refers to the results by using only uniform quantizers for calibration. * and † indicate that the results are re-produced by using the official code.

Method	Opt.	PSQ	W/A	Mask R-CNN				Cascade Mask R-CNN			
				Swin-T		Swin-S		Swin-T		Swin-S	
				AP^b	AP^m	AP^b	AP^m	AP^b	AP^m	AP^b	AP^m
Full-Precision	-	-	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
Baseline*	×	Uniform	4/4	34.6	34.2	40.8	38.6	45.9	40.2	47.9	41.6
RepQ-ViT [27]	×	$\log\sqrt{2}$	4/4	36.1	36.0	$44.2_{42.7}$†	40.2	47.0	41.1	49.3	43.1
ERQ [55]	×	$\log\sqrt{2}$	4/4	36.8	36.6	43.4	40.7	47.9	42.1	50.0	43.6
I&S-ViT [54]	✓	SULQ	4/4	37.5	36.6	43.4	40.3	48.2	42.0	**50.3**	43.6
DopQ-ViT [49]	✓	TanQ	4/4	37.5	36.5	43.5	40.4	48.2	42.1	**50.3**	**43.7**
QDrop* [46]	✓	Uniform	4/4	36.2	35.4	41.6	39.2	47.0	41.3	49.0	42.5
APHQ-ViT (Ours)	✓	Uniform	4/4	**38.9**	**38.1**	**44.1**	**41.0**	**48.9**	**42.7**	**50.3**	**43.7**

number in both MLP Reconstruction and QDrop reconstruction as 32, 4e-5, 1e-3 and 20000, respectively. In addition, we set the percentile $p = 0.99$ in Eq. (12).

4.2. Quantization Results on ImageNet

We first compare our method to the state-of-the-art approaches for post-training quantization of ViTs on ImageNet: 1) the calibration-only methods including PTQ4ViT [50], APQ-ViT [8], RepQ-ViT [27], ERQ [55], IGQ-ViT [38] and AdaLog [47]; and 2) the reconstruction-based methods including DopQ-ViT [49], QDrop [46] and OASQ [36].

As summarized in Table 1, for 4-bit quantization, some of the compared methods suffer a remarkable degradation in accuracy due to severe quantization loss of weights and activations. However, the performance of the proposed APHQ-ViT remains competitive compared to the full-precision models and consistently outperforms existing methods. As for 3-bit quantization, calibration-only methods yield an extremely low performance (*e.g.* 0.1%) in most scenarios. Reconstruction-based methods like DopQ-ViT and I&S-ViT also suffer significant accuracy loss on models that are challenging to quantize (*e.g.* ViT-S and DeiT-T). By contrast, APHQ-ViT maintains more stable accuracy when reducing the precision from 32 bits to 3 bits. It surpasses the second-best method, DopQ-ViT, by 10.71% when using the DeiT-T backbone and achieves an average improvement of 7.21%.

4.3. Quantization Results on COCO

We further evaluate our method on COCO for object detection and instance segmentation. As shown in Table 2, the baseline method, which employs only uniform quantizers and QDrop, achieves lower accuracy compared to other calibration-only and reconstruction-based methods that utilize specific quantizers. By employing the APH loss and MLP Reconstruction, our method achieves results on par with or superior to those using specific quantizers.

Table 3. Ablation results w.r.t the top-1 accuracy (%) of the proposed main components on ImageNet with the W3/A3 setting.

Method	ViT-S	ViT-B	DeiT-T	DeiT-S	Swin-S
Full-Prec.	81.39	84.54	72.21	79.85	81.80
QDrop	38.31	73.79	46.69	52.55	74.11
+APH	59.11	76.05	53.82	67.40	75.44
+APH +MR	**63.17**	**76.31**	**55.42**	**68.76**	**76.10**

4.4. Ablation Studies

Effect of the Main Components. We first evaluate the effectiveness of the proposed Average Perturbation Hessian (APH) loss and the MLP Reconstruction (MR) method. As displayed in Table 3, applying the APH loss on QDrop reconstruction significantly promotes the top-1 accuracy across distinct Vision Transformer architectures. Specifically, the accuracy is improved by 20.80%, 14.85%, and 7.13% when using ViT-S, DeiT-S, and DeiT-T on W3/A3, respectively. MLP Reconstruction consistently boosts the accuracy when combined with the APH loss.

Average Perturbation Hessian. To validate the effectiveness of the proposed APH loss, we compare it with the alternative representative quantization loss, including the MSE loss [46] and the BRECQ based Hessian (BH) loss. We further compare it with the original Perturbation Hessian (PH) without averaging. As shown in Table 4, the PH loss outperforms other quantization losses in most ViT architectures, and the APH loss further improves the accuracy.

MLP Reconstruction. We separately reconstruct the MLP module, *i.e.*, performing MLP Reconstruction one by one without utilizing QDrop reconstruction. As summarized in Table 5, except for a performance drop of over 1% on DeiT-T, the accuracy loss on other models is less than 0.5%. On ViT-B, the accuracy even surpasses that of the full-precision

Table 4. Ablation results w.r.t the top-1 accuracy (%) of the proposed Perturbation Hessian, compared to other losses on ImageNet with the W3/A3 setting. "BH", "PH" and "APH" denote "BRECQ-based Hessian", "Perturbation Hessian" and "Average Perturbation Hessian", respectively.

Method	ViT-S	ViT-B	DeiT-T	DeiT-S	Swin-S
Full-Prec.	81.39	84.54	72.21	79.85	83.23
MSE [46]	38.31	73.79	46.69	52.55	74.11
BH [25]	54.33	66.62	49.27	63.72	75.20
PH	55.14	72.80	52.25	66.12	75.40
APH	**59.11**	**76.05**	**53.82**	**67.40**	**75.44**

Table 5. Ablation results w.r.t the top-1 accuracy (%) of the proposed MLP Reconstruction method on ImageNet.

Method	ViT-S	ViT-B	DeiT-T	DeiT-S	Swin-S
Full-Prec.	81.39	84.54	72.21	79.85	83.23
MLP Recon.	80.90	84.84	71.07	79.38	83.12

model by adopting the MR method.

We provide more ablation results in Sec. B of the *supplementary material*.

4.5. Analysis of Inference Efficiency on MR

MLP Reconstruction replaces the GELU activation function with ReLU. Unlike GELU, which incurs additional computational overhead, ReLU can be folded into the preceding linear layer. As a consequence, the proposed MR method not only promotes quantization accuracy but also accelerates inference. Since quantization below 8 bits typically requires specialized hardware [9, 25, 55], we benchmark the quantized model at W8A8 on an Intel i5-12400F CPU. As shown in Table 6, 8-bit quantization generally achieves a 1.4 to 1.6 times speedup. By replacing GELU with ReLU via MR, we further improve inference efficiency.

4.6. Discussion on Training Efficiency

The MLP Reconstruction method in APHQ-ViT introduces additional training overhead. However, the extra training cost is acceptable. As shown in Table 7, our method incurs less training overhead, compared to QAT methods such as LSQ. Furthermore, our approach requires only 1024 unlabeled images as a calibration set, eliminating fine-tuning on the entire dataset, as is typically required by QAT methods.

5. Conclusion

In this paper, we propose a novel post-training quantization approach dubbed APHQ-ViT for Vision Transformers. We first demonstrate that the current Hessian guided loss adopts an inaccurate estimated Hessian matrix, and present an improved Average Perturbation Hessian (APH) loss. Based on APH, we develop an MLP Reconstruction method that simultaneously replaces the GELU activation function with ReLU and significantly reduces the activation range. Extensive experimental results show the effectiveness of our approach across various Vision Transformer architectures and vision tasks, including image classification, object detection, and instance segmentation. Notably, compared to the state-of-the-art methods, APHQ-ViT achieves an average improvement of 7.21% on ImageNet with 3-bit quantization using only uniform quantizers.

Table 6. Comparison of latency and throughput of ViTs under W8A8 quantization to full-precision models. "AF" indicates the adopted activation function. "Lat." refers to the model latency (in milliseconds). "TP" stands for the throughput (in images per second). "SR" is the speedup rate.

Model	AF	Bits	Lat.	TP	SR
DeiT-T	GELU	32	30.93	32.08	×1
	GELU	8	22.34	44.76	× 1.40
	ReLU	8	**20.66**	**48.40**	× **1.51**
DeiT-S	GELU	32	100.03	9.97	×1
	GELU	8	63.89	15.65	× 1.57
	ReLU	8	**58.40**	**17.12**	× **1.72**
DeiT-B	GELU	32	346.93	2.88	×1
	GELU	8	217.96	4.59	× 1.59
	ReLU	8	**198.80**	**5.03**	× **1.75**
Swin-S	GELU	32	255.56	3.90	×1
	GELU	8	180.42	5.54	× 1.42
	ReLU	8	**171.88**	**5.82**	× **1.49**
Swin-B	GELU	32	411.07	2.43	×1
	GELU	8	282.28	3.54	× 1.44
	ReLU	8	**264.38**	**3.78**	× **1.54**

Table 7. Comparison of the training time cost and accuracy (%) under W3/A3 by using distinct quantization methods on a single Nvidia RTX 4090 GPU.

Model	Method	PTQ	Data Size	Time Cost	Acc.
DeiT-S	LSQ [12]	×	1280 K	∼170 h	77.3
	QDrop [46]	✓	1024	47 min	52.6
	APHQ-ViT	✓	1024	62 min	68.8
Swin-S	LSQ [12]	×	1280 K	∼450 h	80.6
	QDrop [46]	✓	1024	130 min	74.1
	APHQ-ViT	✓	1024	170 min	76.1

Acknowledgments

This work was partly supported by the Beijing Municipal Science and Technology Project (No. Z231100010323002), the National Natural Science Foundation of China (Nos. 62202034,62176012,62022011,62306025), the Beijing Natural Science Foundation (No. 4242044), the Aeronautical Science Foundation of China (No. 2023Z071051002), CCF-Baidu Open Fund, the Research Program of State Key Laboratory of Virtual Reality Technology and Systems, and the Fundamental Research Funds for the Central Universities.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 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 *ECCV*, pages 213–229, 2020. 1

[3] Chun-Fu (Richard) Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, pages 347–356, 2021. 1

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, I Pierce, Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 1, 2

[6] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic DETR: end-to-end object detection with dynamic attention. In *ICCV*, pages 2968–2977, 2021. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 1

[8] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *ACM MM*, pages 5380–5388, 2022. 2, 6, 7

[9] Peiyan Dong, Lei Lu, Chao Wu, Cheng Lyu, Geng Yuan, Hao Tang, and Yanzhi Wang. Packqvit: Faster sub-8-bit vision transformers via full and packed quantization on the mobile. In *NeurIPS*, 2023. 8

[10] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: hessian aware quantization of neural networks with mixed-precision. In *ICCV*, pages 293–302, 2019. 4

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 6

[12] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *ICLR*, 2020. 1, 2, 8

[13] Fisher and Ronald Aylmer. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922. 3

[14] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: accurate post-training quantization for generative pre-trained transformers. In *ICLR*, 2023. 2

[15] 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

[16] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015. 4

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 6

[18] Yefei He, Zhenyu Lou, Luoming Zhang, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Bivit: Extremely compressed binary vision transformers. In *ICCV*, 2023. 2

[19] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[20] Haocheng Huang, Jiaxin Chen, Jinyang Guo, Ruiyi Zhan, and Yunhong Wang. TCAQ-DM: timestep-channel adaptive quantization for diffusion models. In *AAAI*, 2025. 2

[21] Stewart James. *Calculus: Early Transcendentals*. Cengage Learning, 2015. 4, 5

[22] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, 2013. 5

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

[24] Lihua Lei and Michael I. Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *International Conference on Artificial Intelligence and Statistics*, 2017. 5

[25] 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. In *ICLR*, 2021. 2, 3, 6, 8

[26] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. In *NeurIPS*, 2022. 2

[27] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *ICCV*, pages 17227–17236, 2023. 2, 6, 7

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 6

[29] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. In *IJCAI*, pages 1173–1179, 2022. 2

[30] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *CVPR*, 2023. 2

[31] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *CVPR*, pages 20321–20330, 2023. 2

[32] 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 *ICCV*, pages 9992–10002, 2021. 1, 6

[33] Lu Lu, Yeonjong Shin, Yanhui Su, and George E. Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019. 5

[34] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *NeurIPS*, 2017. 5

[35] Chengtao Lv, Hong Chen, Jinyang Guo, Jinyang Guo, Jinyang Guo, Yifu Ding, and Xianglong Liu. PTQ4SAM: post-training quantization for segment anything. In *CVPR*, 2024. 2

[36] Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Outlier-aware slicing for post-training quantization in vision transformer. In *ICML*, 2024. 2, 3, 6, 7

[37] James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, pages 2408–2417, 2015. 4

[38] Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *CVPR*, 2024. 2, 6, 7

[39] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *ICML*, pages 7197–7206, 2020. 2, 3, 4

[40] Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks. *arXiv preprint arXiv:2205.07877*, 2022. 1

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[42] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7242–7252, 2021. 1

[43] Becker Suzanna and Lecun Yann. Improving the convergence of back-propagation learning with second-order methods. In *Connectionist Models Summer School*, 1989. 4

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 6

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1

[46] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2022. 2, 3, 6, 7, 8

[47] Zhuguanyu Wu, Jiaxin Chen, Hanwen Zhong, Di Huang, and Yunhong Wang. Adalog: Post-training quantization for vision transformers with adaptive logarithm quantizer. In *ECCV*, 2024. 2, 6, 7

[48] Zhiyong Xu, Weicun Zhang, Tianxiang Zhang, Zhifang Yang, and Jiangyun Li. Efficient transformer for remote sensing image segmentation. *Remote. Sens.*, 13(18):3585, 2021. 1

[49] Lianwei Yang, Haisong Gong, and Qingyi Gu. Dopq-vit: Towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *arXiv preprint arXiv:2408.03291*, 2024. 2, 6, 7

[50] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV*, pages 191–207, 2022. 2, 5, 6, 7

[51] Yanan Zhang, Jiaxin Chen, and Di Huang. Cat-det: Contrastively augmented transformer for multimodal 3d object detection. In *CVPR*, 2022. 1

[52] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. 1

[53] Hanwen Zhong, Jiaxin Chen, Yutong Zhang, Di Huang, and Yunhong Wang. Transforming vision transformer: Towards efficient multi-task asynchronous learner. In *NeurIPS*, 2024. 1

[54] Yunshan Zhong, Jiawei Hu, Mingbao Lin, Mengzhao Chen, and Rongrong Ji. I&s-vit: An inclusive & stable method for pushing the limit of post-training vits quantization. *arXiv preprint arXiv:2311.10126*, 2023. 2, 3, 6, 7

[55] Yunshan Zhong, Jiawei Hu, You Huang, Yuxin Zhang, and Rongrong Ji. ERQ: Error reduction for post-training quantization of vision transformers. In *ICML*, 2024. 2, 6, 7, 8

[56] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. Octr: Octree-based transformer for 3d object detection. In *CVPR*, 2023. 1

[57] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1