AIQViT: Architecture-Informed Post-Training Quantization for Vision Transformers

Runqing Jiang[1], **Ye Zhang**[1], **Longguang Wang**[2], **Pengpeng Yu**[1], **Yulan Guo**[1,*]
[1]Shenzhen Campus, Sun Yat-sen University
[2]Aviation University of Air Force
{jiangrq3,yupp5}@mail2.sysu.edu.cn, zhangy2658@mail.sysu.edu.cn, wanglongguang15@nudt.edu.cn, guoyulan@sysu.edu.cn

Abstract

Post-training quantization (PTQ) has emerged as a promising solution for reducing the storage and computational cost of vision transformers (ViTs). Recent advances primarily target at crafting quantizers to deal with peculiar activations characterized by ViTs. However, most existing methods underestimate the information loss incurred by weight quantization, resulting in significant performance deterioration, particularly in low-bit cases. Furthermore, a common practice in quantizing post-Softmax activations of ViTs is to employ logarithmic transformations, which unfortunately prioritize less informative values around zero. This approach introduces additional redundancies, ultimately leading to suboptimal quantization efficacy. To handle these, this paper proposes an innovative PTQ method tailored for ViTs, termed AIQViT (**A**rchitecture-**I**nformed Post-training **Q**uantization for **ViT**s). First, we design an architecture-informed low-rank compensation mechanism, wherein learnable low-rank weights are introduced to compensate for the degradation caused by weight quantization. Second, we design a dynamic focusing quantizer to accommodate the unbalanced distribution of post-Softmax activations, which dynamically selects the most valuable interval for higher quantization resolution. Extensive experiments on five vision tasks, including image classification, object detection, instance segmentation, point cloud classification, and point cloud part segmentation, demonstrate the superiority of AIQViT over state-of-the-art PTQ methods.

Introduction

Thanks to the powerful scalability and superior ability in building long-range dependencies, vision transformers (ViTs) have accomplished cutting-edge performances in various vision tasks, such as image classification (Dosovitskiy et al. 2020; Touvron et al. 2021), image object detection (Carion et al. 2020; Li et al. 2023a), and point cloud analysis (Zhao et al. 2021; Guo et al. 2021). However, the computational intensity and substantial memory requirements of off-the-shelf ViTs pose significant barriers to their widespread deployment on resource-constrained devices. To this end, a number of model compression techniques have been studied, including network pruning (Fang et al. 2023; Jiang et al. 2022), knowledge distillation (Jiang et al. 2023; Zhao et al. 2022), and model quantization (Nagel et al. 2022; Xiao et al. 2023), to reduce the storage and computational cost of complicated models while preserving their performances.

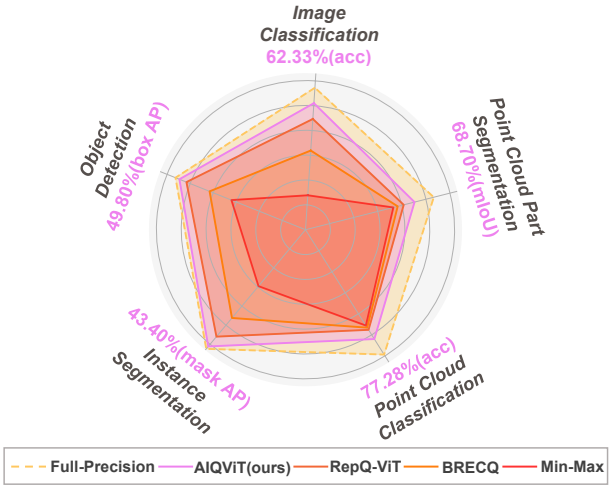

Figure 1: The performances of different approaches on different tasks (including image classification, object detection, instance segmentation, point cloud classification, and point cloud part segmentation), where the W4/A4 setting is used for quantization. Best viewed in color.

Among the model compression techniques, model quantization has emerged as a widely adopted paradigm, which compresses models by reducing the bit-width of weights and activations. Some quantization-aware training methods (Lin et al. 2015; Nagel et al. 2022) require a retraining process to recover accuracy, which is computationally expensive and time-consuming. Moreover, they typically rely on the full training datasets, which may not be feasible due to privacy problems. To handle these, post-training quantization (PTQ) has gained increasing popularity in recent years. Compared with the quantization-aware training methods, PTQ-based methods (Li et al. 2020; Wei et al. 2021) only invocate a small subset of the original training data for calibration, providing a more viable option for model compression.

Early PTQ-based works (Li et al. 2020; Wei et al. 2021)

*Corresponding Author

on computer vision tasks primarily target at compressing convolutional neural networks (CNNs), and have obtained remarkable performances. However, it remains challenging to apply PTQ to ViT models due to their unique components (e.g., Softmax) different from those of CNNs, especially in low-bit cases. To this end, some methods (Liu et al. 2021b; Yuan et al. 2022; Li et al. 2023b) have explored PTQ techniques for ViTs, where specific feature distributions (i.e., post-Softmax and post-LayerNorm) are taken into account. Unfortunately, these works still show limited performances in low-bit cases. This is because the parameter space of full-precision models is not inherently aligned with their quantized counterparts, resulting in substantial quantization errors in weights. Besides, the commonly adopted logarithmic operations tend to preserve precision for zero-around values, which may involve intensive redundant information (Meng et al. 2022), disturbing the quantization efficacy.

To address these challenges, this paper proposes AIQViT (**A**rchitecture-**I**nformed Post-training **Q**uantization for **ViT**s), a PTQ method tailored for ViTs across different tasks as presented in Figure 1. First, we propose an architecture-informed low-rank compensation mechanism to alleviate the degradation induced by weight quantization. Specifically, we introduce learnable weights for each linear layer to remedy the information loss caused by weight quantization. To reduce training cost and prevent overfitting, these weights are designed in low-rank formats, with their ranks determined using network architecture search. Second, we design a DFQ (**D**ynamic **F**ocusing **Q**uantizer) to handle the post-Softmax activations of ViTs. To be specific, DFQ learns to identify the most valuable interval of post-Softmax activations and subsequently applies standard uniform quantization instead of logarithmic operations within this interval, and thus obtains better quantization efficiency.

In summary, our main contributions are listed as follows:

- We propose AIQViT, which is composed of an architecture-informed low-rank compensation mechanism and a dynamic focusing quantizer, to quantize ViTs in a post-training manner, even in low-bit cases.
- We develop an architecture-informed low-rank compensation mechanism to compensate for the information loss caused by the weight quantization of ViT models.
- We design a DFQ to deal with the unbalanced distribution of post-Softmax activations without using logarithmic operations, achieving higher quantization efficiency and better performances.
- Extensive experiments on five tasks (including image classification, object detection, instance segmentation, point cloud classification, and point cloud part segmentation) with multiple ViT variants validate the superiority of AIQViT against several state-of-the-art PTQ methods.

Related Works

Vision Transformers

Transformer, which arises from natural language processing tasks, has sparked great interest in the field of computer vision (Yan et al. 2023). ViT (Dosovitskiy et al. 2020) pioneeringly introduces a pure transformer architecture on image classification by splitting images into sequences of patches, and attains excellent performance compared to some advanced convolutional neural networks. Later, DeiT (Touvron et al. 2021) is proposed, where transformer can be efficiently trained on mid-size image datasets. Inspired by these, some efforts attempt to apply ViTs to other vision tasks, such as object detection (Carion et al. 2020; Fang et al. 2021) and point cloud understanding (Guo et al. 2021; Zhao et al. 2021). For instance, DETR (Carion et al. 2020) regards object detection as a set prediction task, in which transformers are used for capturing the relationship between objects. After that, YOLOS (Fang et al. 2021) uses an attention-only architecture and sequence-to-sequence strategy for object detection. Additionally, PCT (Guo et al. 2021) is proposed as the first transformer-based backbone for point cloud understanding. PCT takes each point as a token and performs vector attention between points in a local neighbor set, and attains promising performance on point cloud classification and point cloud part segmentation tasks. Generally, the impressive accomplishments of ViTs heavily depend on substantial computation and storage overhead, hindering their applications on devices with limited memory and computation resources. In this paper, we are concerned with the problem of producing quantized ViTs with low-bit weights and activations in a post-training manner.

Model Quantization

One promising solution to compress complex models and accelerate inference is model quantization. Model quantization decreases the bit-width of weights and activations to alleviate the memory footprint and the computational overhead. Early works (Lin et al. 2015; Nagel et al. 2022) commonly adopt the paradigm of quantization-aware training (QAT) to employ retraining on the entire training data for higher accuracy after quantization. Despite their effectiveness, these methods inevitably suffer from privacy problems and massive time overhead, particularly for large-scale ViTs.

In contrast to QAT, post-training quantization (PTQ), which crafts quantized models without expensive retraining, has attracted increasing attention in model compression. Adaround (Nagel et al. 2020) suggests that the naive round-to-nearest is not optimal for quantization, and designs an adaptive rounding strategy to lower the quantization error. After that, BRECQ (Li et al. 2020) uses basic building blocks in neural networks to perform reconstruction, obtaining impressive image classification accuracy with 4-bit ResNet. Despite their achievements on CNNs, they show limited performances on ViTs. Thus, many works (Lin et al. 2022; Li et al. 2023b) attempt to apply PTQ to ViT models. Due to the uneven activation distributions of post-LayerNorm, RepQ-ViT (Li et al. 2023b) uses channel-wise quantization to alleviate the severe inter-channel variations, and then uses reparameterization skills to convert scale factors into layer-wise formats. However, these methods typically rely on log2-based quantizers to deal with the post-Softmax activations, which focus on zero-around values, containing massive redundancy. Besides, these quantizers entail specialized operations to achieve efficiency (Lee et al. 2017; Lin et al. 2022).

Preliminaries

Overview of ViTs

ViTs are primarily composed of an embedding layer and some stacked transformer blocks, contributing to capturing long-range relationships hidden in different patches. Specifically, an input image I is firstly split into several patches and then fed into a linear layer to obtain its feature representation $X^{(0)} \in \mathbb{R}^{N \times D}$, where N denotes the number of patches and D denotes the embedding dimension. Subsequently, $X^{(0)}$ is sent into a transformer block consisting of a multi-head self-attention (MHSA) followed by a multi-layer perceptron (MLP) module. Mathematically, for the l-th transformer block, the above procedure is formulated as:

$$\tilde{X}^{(l-1)} = \text{MHSA}^{(l)}(\text{LayerNorm}(X^{(l-1)})) + X^{(l-1)} \quad (1)$$

$$X^{(l)} = \text{MLP}^{(l)}(\text{LayerNorm}(\tilde{X}^{(l-1)})) + \tilde{X}^{(l-1)}. \quad (2)$$

The MHSA module attends to all the patches by:

$$\text{Attn}_i = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{D_h}})V_i \quad (3)$$

$$\text{MHSA}(X) = \text{concat}(\text{Attn}_1, \text{Attn}_2, ..., \text{Attn}_H)W_o \quad (4)$$

where $Q_i = XW_i^q$, $K_i = XW_i^k$, $V_i = XW_i^v$ respectively represent the query, key and value for the i-th head. D_h denotes the feature dimension for each head. H is the number of attention heads. For readability, we omit the bias term. Then, the MLP encodes the patch features individually, which is defined as:

$$\text{MLP}(X) = \text{GELU}(XW_1)W_2 \quad (5)$$

where W_1 and W_2 denote the weights in linear layers.

Quantizers

Uniform Quantizer. It is widely used due to its broad compatibility across hardware platforms and is defined as:

$$x_q = \text{Quant} - \text{U}(x) = \text{clamp}(\lfloor \frac{x}{s} \rceil + z, 0, 2^k - 1) \quad (6)$$

where the x is a floating-point value and x_q is the quantized value. The s, z, and k represent the quantization scale, zero point, and bit-width, respectively. In line with previous works (Li et al. 2023b; Zhong et al. 2023), we employ channel-wise quantization for weights and layer-wise quantization for activations during inference.

Log2-Based Quantizer. It is specifically designed to address the challenge of non-negative long-tail activation quantization, which can be mathematically formulated as:

$$x_q = \text{Quant} - \text{LU}(x) = \text{clamp}(\lfloor -\log_2(\frac{x}{s}) \rceil, 0, 2^k - 1). \quad (7)$$

Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) (Hu et al. 2022) is broadly studied in the realm of large language models to achieve parameter-efficient fine-tuning. Mathematically, given a pre-trained weight matrix $W_0 \in \mathbb{R}^{m \times d}$, the process of LoRA is:

$$h = xW_0 + xBA \quad (8)$$

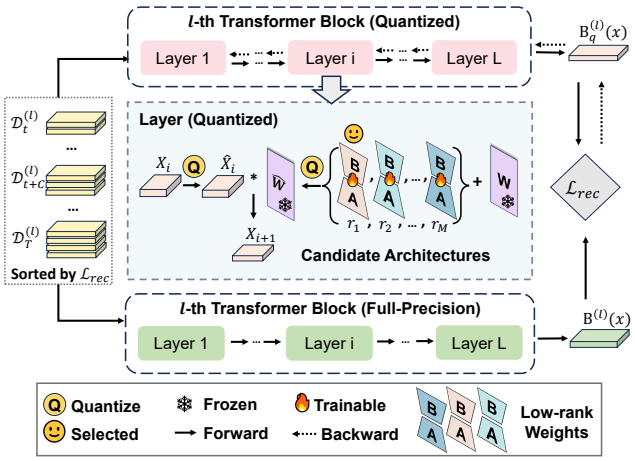

Figure 2: Overview of architecture-informed low-rank compensation. First, we employ differential architecture search to identify the optimal rank r from a candidate architecture set. Subsequently, we freeze the original weights and optimize the selected low-rank weights by minimizing the reconstruction loss between the full-precision block and the quantized block. During this process, the training set is incrementally expanded in a curriculum learning (CL) manner.

where $h \in \mathbb{R}^d$ is the hidden state corresponding to input $x \in \mathbb{R}^m$, $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times m}$, and the rank $r << \min(d, m)$. During training, only the A and B are updated by gradient descent while remaining W_0 frozen. After training, the BA is seamlessly merged into W_0 by assigning $W_0 = W_0 + BA$.

Method

Architecture-Informed Low-Rank Compensation

Unlike CNNs, ViTs are composed of a large number of fully connected (FC) layers, necessitating a significant amount of computing and storage resources. However, due to the more intricate architectures of ViTs, directly applying weight quantization for these layers may incur significant information loss, leading to inferior accuracy. To address this, some learnable weights are introduced for FC layers to compensate for information loss. These learnable weights are formulated in low-rank formats, effectively reducing optimization costs while preventing overfitting that may arise from limited data. The process is shown in Figure 2.

The Process of Low-Rank Compensation. Inspired by LoRA (Hu et al. 2022), we introduce learnable low-rank weights into quantization and reformulate Eq. (8):

$$h = \text{Quant-U}(x)\text{Quant-U}(W_0 + BA) \quad (9)$$

where we hold W_0 fixed and update A and B to minimize the following reconstruction loss:

$$\mathcal{L}_{rec}(\mathcal{D}_t^{(l)}, \Theta^{(l)}) = \mathbb{E}_{x \in \mathcal{D}_t^{(l)}} \left(\left\| \text{B}_q^{(l)}(x) - \text{B}^{(l)}(x) \right\|_\text{F} \right) \quad (10)$$

where $\mathcal{D}_t^{(l)}$ denotes the input dataset for the l-th block at t-th iteration, $\Theta^{(l)}$ denotes all the learnable low-rank weights for

the l-th block, $\|\cdot\|_F$ represents the Frobenius norm, $\mathrm{B}^{(l)}(x)$ and $\mathrm{B}_q^{(l)}(x)$ represent the outputs of the l-th full-precision block and quantized block, respectively. Thanks to the learnable low-rank weights, the quantized model is encouraged to learn a parameter space which is compatible for quantization, and the reconstruction error induced by weight quantization is significantly alleviated without incurring massive optimization overhead. During reconstruction, channel-wise quantization is used for post-LayerNorm activations. After that, we use scale reparameterization (Li et al. 2023b) to convert them into layer-wise quantization.

The Choice of Rank r. Since the choice of r has a profound impact on the performance of quantized models, we are motivated to automate the search for r. Specifically, we model the search of r as a network architecture search problem, and thus use efficient differentiable architecture search to handle this issue. Given a proposal set $\mathcal{S} = \{r_1, r_2, \cdots, r_M\}$ for the rank r, the forward pass of a fully-connected layer is:

$$y = \hat{x}W_0 + \sum_{r_j \in \mathcal{S}} \phi(\overline{\alpha}_{r_j} \hat{x} B_{r_j} A_{r_j}), \tag{11}$$

,

$$\overline{\alpha}_{r_j} = \frac{\exp(\alpha_{r_j})}{\sum_{i=1}^{M} \exp(\alpha_{r_i})} \tag{12}$$

where $\hat{x} = \text{Quant-U}(x)$ represents the quantized value of input x, and $B_{r_j} \in \mathbb{R}^{d \times r_j}$, $A_{r_j} \in \mathbb{R}^{r_j \times m}$ are candidate weights with respect to r_j. $\alpha = \{\alpha_{r_i}\}_{i=1}^{M}$ are a set of learnable parameters to control the importance of each architecture, and the $\phi(\cdot)$ denotes drop-path operation (Larsson, Maire, and Shakhnarovich 2016). Meanwhile, we divide the calibration set $\mathcal{D}$ into a training set $\mathcal{D}_{train}$ and a validation set $\mathcal{D}_{val}$, and define the optimization objective:

$$\min_{\alpha} \quad \mathcal{L}_{rec}(\mathcal{D}_{val}^{(l)}, \Theta_*^{(l)}) \tag{13}$$

$$\text{s.t. } \Theta_*^{(l)} = \arg\min_{\Theta^{(l)}} \mathcal{L}_{rec}(\mathcal{D}_{train}^{(l)}, \Theta^{(l)}). \tag{14}$$

The above bilevel optimization problem can be efficiently solved by approximate architecture gradient used in (Liu, Simonyan, and Yang 2018). After that, for each linear layer, the optimal rank is determined by $r^* = \arg\max_{r \in \mathcal{S}} \alpha_r$.

Note that, albeit our AIQViT and QLLM (Liu et al. 2023) use LoRA to compensate for quantization error, the differences between AIQViT and QLLM are significant. First, QLLM aims to quantize large language models while our AIQViT is designed for ViTs. Second, QLLM sets the value of r empirically. However, it is found that the rank r is critical for the final performance. Thus, we determine r by using network architecture search instead of manual settings.

Dynamic Focusing Quantizer

As indicated in Figure 3(a), the post-Softmax activations exhibit unbalanced distribution, which has become one of the central challenges in quantizing ViTs. To handle this, several methods (Li et al. 2023b; Lin et al. 2022) leverage the non-linearity of logarithmic operation and adopt log2-based quantizers to quantize the post-Softmax activations. Here, we visualize the execution process of a log2 quantizer in Figure 3(b). It can be observed that inputs with smaller values

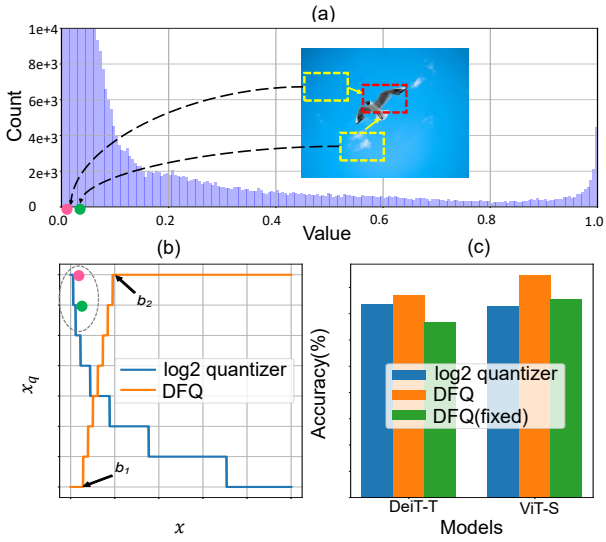

Figure 3: (a) Histogram of the first MHSA module's post-Softmax activations in DeiT-T. (b) log2 quantizer (in blue) and DFQ (in orange). (c) Results on ImageNet with W3/A3 quantization. "DFQ(fixed)" means all the layers use the same interval. Best viewed in color.

generally correspond to higher quantization resolution and vice versa. Such a schema indicates that the most valuable activations are located within a partial interval instead of the whole one. However, the log2 quantizer tends to preserve values around 0, which may contain a plethora of redundant information, disturbing the efficacy of quantization. Besides, the log2 quantizer keeps the interval fixed for each layer, which may not be the optimal solution. Based on the above insights, we design a DFQ (**D**ynamic **F**ocusing **Q**uantizer) to select the most valuable interval for each Softmax layer dynamically. To be specific, let $[b_1, b_2]$ be the selected interval where b_1 and b_2 are learnable parameters, which is:

$$x_q = \begin{cases} 0, & x < b_1 \\ \text{Quant-U}(x), & b_1 \le x \le b_2 \\ 2^k - 1, & x > b_2 \end{cases} \tag{15}$$

where $x \in [0, 1]$ is the activation after Softmax and k is the bit-width. By dynamically selecting b_1 and b_2, DFQ will focus on the most valuable interval and prioritize more bits accordingly. Then, we use a uniform quantizer for values in $[b_1, b_2]$, and directly convert the values in $[0, b_1)$ and $(b_2, 1]$ to 0 and $2^k - 1$, respectively. In this way, post-Softmax activations are quantized without logarithmic operations, and we find DFQ can achieve comparable results with log2-based quantizers, as illustrated in Figure 3(c).

Optimization Strategy

To minimize Eq. (10), we deviate from the standard training pipeline employed in BRECQ (Li et al. 2020). That is because we observe that arranging samples in a meaningful order yields better performances, especially in ultra-low

bit case (i.e., 3-bit). Inspired by curriculum learning, we optimize Eq. (10) by firstly feeding easier samples, and then gradually introducing the harder ones. To be specific, reconstruction loss is selected to gauge the hardness of each sample, and the training data for the t-th iteration is:

$$\mathcal{D}_t^{(l)} = \arg\min_{\hat{\mathcal{D}}:|\hat{\mathcal{D}}|\geq\lambda(t)\cdot|\mathcal{D}|} \mathcal{L}_{rec}(\hat{\mathcal{D}}, \Theta^{(l)}) \tag{16}$$

where $\mathcal{D}$ denotes the calibration data, the $|\mathcal{D}|$ and $|\hat{\mathcal{D}}|$ represent the size of $\mathcal{D}$ and $\hat{\mathcal{D}}$. $\lambda(t)$ is used to schedule the proportion of training samples at the t-th iteration, which is defined by a linear function (Wang, Chen, and Zhu 2022):

$$\lambda(t) = \min\left(1, \lambda_0 + \frac{1-\lambda_0}{T}\cdot t\right) \tag{17}$$

where λ_0 is the initial proportion of training samples which is set to 0.5 in our method, and T is the total number of iterations. Thus, the quantized model tends to learn high-confidence regions in the early training stage, mitigating the negative impact of outliers and preparing a more favorable parameter space for stable optimization.

Experiment

Experimental Setups

Datasets and Competing Methods. We adopt ImageNet (Krizhevsky, Sutskever, and Hinton 2012), ModelNet40 (Wu et al. 2015), and ShapeNetPart (Yi et al. 2016) for image classification, point cloud classification, and point cloud part segmentation respectively. The COCO (Lin et al. 2014) dataset is used to evaluate object detection and instance segmentation tasks. We use BRECQ (Li et al. 2020), QDrop (Wei et al. 2021), PTQ4ViT (Yuan et al. 2022), RepQ-ViT (Li et al. 2023b), APQ-ViT (Ding et al. 2022), Min-Max (Nagel et al. 2020) as competing methods.

Implementation Details. For image classification, we adopt the experimental settings from (Zhong et al. 2023) and randomly select a calibration set of 1,024 samples from the ImageNet dataset. We evaluate our method on a variety of models, including ViT (Dosovitskiy et al. 2020), DeiT (Touvron et al. 2021), and Swin Transformer (Liu et al. 2021a), with batch size set to 24. For object detection and instance segmentation, we employ Mask-RCNN (He et al. 2017) and Cascade Mask-RCNN (Cai and Vasconcelos 2018) on the COCO dataset, using a calibration set of 256 samples and a batch size of 1. For point cloud classification and part segmentation, we evaluate our method on Point Transformer (Zhao et al. 2021) with a batch size of 32 and a calibration set of 512 samples. The iteration numbers are set to 2,000 and 6,000 for network architecture search and calibration, respectively. We empirically set $\mathcal{S} = \{10, 20, 50, 100, 150\}$ for all model variants across all vision tasks.

Image Classification on ImageNet

Table 1 shows the test accuracy results obtained by various competing methods on ImageNet with different models and bit-widths. Notably, in the case of W3/A3 quantization, most competing methods exhibit unfavorable test accuracy on all model variants. For instance, RepQ-ViT achieves only 0.44% and 0.17% top-1 accuracy in ViT-S and ViT-B quantization, respectively, which is far from achieving practical usability. In contrast, BRECQ and QDrop perform slightly better than RepQ-ViT by leveraging block-wise reconstruction, which enables more accurate quantization parameters to be obtained. In the W4/A4 quantization setting, APQ-ViT achieves accuracy of 47.94% and 43.55% in DeiT-T and DeiT-S quantization, respectively, while RepQ-ViT improves test accuracy by over 9% owing to the application of scale reparameterization skills. Furthermore, in the W6/A6 quantization setting, all competing methods attain satisfied performance, with minimal performance gaps across different methods. Nonetheless, our AIQViT achieves the highest top-1 accuracy in most scenarios. For example, compared to the Full-Precision DeiT-B, our AIQViT achieves 81.40% test accuracy in the case of W6/A6 quantization, which corresponds to a mere 0.4% accuracy loss.

Object Detection and Instance Segmentation on COCO

In line with RepQ-ViT (Li et al. 2023b), we evaluate the performance of object detection and instance segmentation tasks on the COCO dataset, with the quantization results shown in Table 2. Notably, PTQ4ViT exhibits the worst performance across different quantization settings, which can be attributed to the loss of generality of its exquisite twin-scale mechanism when applied to more complex architectures. Furthermore, our experiments reveal that APQ-ViT achieves desired results when employing Swin-S as the backbone, but suffers from significant performance degradation when using Swin-T as the backbone, indicating a lack of robustness with respect to the choice of backbone. In contrast, our AIQViT demonstrates superior performance when adopting W4/A4 quantization on Mask R-CNN with Swin-S as the backbone, outperforming BRECQ by 4.6 box AP and 2.7 mask AP. Notably, for the W6/A6 quantization of Cascade Mask R-CNN with Swin-T as the backbone, our AIQViT obtains a box AP of 50.2 and a mask AP of 43.6, which is very close to its full-precision counterpart, with a mere 0.2 box AP and 0.1 mask AP gap.

Point Cloud Classification on ModelNet40

We also demonstrate the effectiveness of AIQViT on the 3D point cloud classification task. Specifically, we select Point Transformer (Zhao et al. 2021) as the backbone and validate our method on ModelNet40. The experimental results are shown in Table 3. As can be observed, the Min-Max obtains the worst performances on both W4/A4 and W6/A6 quantization, which is mainly due to the inaccurate quantization parameters estimated by min-max criteria. Besides, our AIQViT outperforms BRECQ by 5% in W4/A4 quantization, indicating that AIQViT can be well applied to 3D point cloud classification tasks.

Point Cloud Part Segmentation on ShapeNetPart

For the 3D point cloud part segmentation tasks, the superiority of AIQViT is validated on the ShapeNetPart dataset with mean intersection over union (mIoU) used as the evaluation

Method	REC	Bits.(W/A)	DeiT-T	DeiT-S	DeiT-B	ViT-S	ViT-B	Swin-S	Swin-B
Full-Precision	-	32/32	72.21	79.85	81.80	81.39	84.54	83.23	85.27
RepQ-ViT	×	3/3	1.15	4.85	7.23	0.44	0.17	1.22	4.87
QDrop	✓	3/3	15.68	16.89	22.38	4.32	5.99	58.96	52.19
BRECQ	✓	3/3	14.12	8.26	12.87	1.86	0.14	7.32	1.21
AIQViT (ours)	✓	3/3	**38.51**	**55.36**	**66.15**	**41.32**	**43.68**	**71.42**	**63.01**
PTQ4ViT	✓	4/4	36.96	34.08	64.39	42.57	30.69	76.09	74.02
RepQ-ViT	×	4/4	57.43	69.03	75.61	65.05	68.48	79.45	78.32
APQ-ViT	✓	4/4	47.94	43.55	67.48	47.95	41.41	77.15	76.48
QDrop	✓	4/4	31.65	35.79	65.47	17.77	21.72	78.92	80.49
BRECQ	✓	4/4	51.60	54.31	62.96	63.90	61.54	76.63	74.15
AIQViT (ours)	✓	4/4	**62.33**	**72.75**	**79.19**	**70.63**	**74.15**	**80.93**	**81.22**
PTQ4ViT	✓	6/6	69.68	76.28	80.25	78.63	81.65	82.38	84.01
RepQ-ViT	×	6/6	70.76	78.90	81.27	**80.43**	83.62	82.79	**84.57**
APQ-ViT	✓	6/6	70.49	77.76	80.42	79.10	82.21	82.67	84.18
QDrop	✓	6/6	70.61	78.91	80.19	73.01	80.86	81.92	83.85
BRECQ	✓	6/6	69.70	73.09	75.67	79.61	77.19	77.66	78.03
AIQViT (ours)	✓	6/6	**70.78**	**78.98**	**81.40**	80.21	**83.68**	**82.81**	84.39

Table 1: Quantization results on ImageNet dataset. The top-1 accuracy (%) is reported as the evaluation metric. "Bits.(W/A)" represents the bit-width for weight and activation. "REC" means "Reconstruction".

Method	REC	Bits.(W/A)	Mask R-CNN				Cascade Mask R-CNN			
			w.Swin-T		w.Swin-S		w.Swin-T		w.Swin-S	
			AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-Precision	-	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT	✓	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT	✓	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
RepQ-ViT	×	4/4	36.1	36.0	44.2	40.2	47.0	41.4	49.3	43.1
BRECQ	✓	4/4	33.7	33.1	39.5	37.7	42.1	36.3	43.2	37.6
AIQViT (ours)	✓	4/4	**38.2**	**36.7**	44.1	**40.4**	**47.1**	**41.4**	**49.8**	**43.4**
PTQ4ViT	✓	6/6	5.8	6.8	6.5	6.6	14.7	13.6	12.5	10.8
APQ-ViT	✓	6/6	**45.4**	41.2	**47.9**	42.9	48.6	42.5	50.5	43.9
RepQ-ViT	×	6/6	45.1	41.2	47.8	43.0	50.0	43.5	51.4	44.6
BRECQ	✓	6/6	36.1	36.0	44.2	40.2	46.9	39.9	48.3	41.9
AIQViT (ours)	✓	6/6	45.3	**41.2**	47.5	**43.0**	**50.2**	**43.6**	**51.5**	**44.6**

Table 2: Quantization results on COCO dataset. "AP^{box}" means the box average precision for object detection, and "AP^{mask}" means the mask average precision for instance segmentation.

metric. Experimental results are provided in Table 4. As presented in Table 4, it can be easily observed that AIQViT consistently outperforms other competing methods on W4/A4 and W6/A6 quantization. Specifically, for the W4/A4 case, AIQViT outperforms BRECQ by a large margin. In the case of W6/A6 quantization, AIQViT achieves 76.99 c.mIoU and 81.76 i.mIoU, which is only 1.58 c.mIoU and 1.50 i.mIoU lower than the Full-Precision model.

Ablation Studies

To demonstrate the effectiveness of the key components in AIQViT, we conduct ablation studies on the ImageNet dataset with DeiT-T. For convenience, the architecture-informed low-rank compensation, dynamic focusing quantizer, and curriculum learning strategy are abbreviated as AILoC, DFQ, and CL, respectively. Quantitive experimental results are detailed in Table 5. Note that, when DFQ is excluded, a uniform quantizer is employed for post-Softmax activations. The results indicate that AIQViT obtains the best results when all the variants are used. Specifically, compared with the vanilla (all variants are excluded), AILoC improves the test accuracy by 15.31%, 10.80%, and 7.61% for W3/A3, W4/A4, and W6/A6 quantization, respectively, confirming the effectiveness of the low-rank compensation mechanism used in AILoC. Besides, AIQViT suffers from an 11.93% accuracy drop when DFQ is absent, claiming the superiority of DFQ in handling low-bit cases. We also observe that the CL strategy brings more significant improvement for low-bit quantization than high-bit quantization. This can be attributed to the fact that the low-bit model benefits more from

Method	REC	Bits.(W/A)	mAcc	OA
Full-Precision	-	32/32	89.67	92.38
Min-Max	×	4/4	70.60	73.99
RepQ-ViT	×	4/4	72.61	77.98
BRECQ	✓	4/4	71.58	76.68
AIQViT(ours)	✓	4/4	**77.28**	**82.66**
Min-Max	×	6/6	83.24	86.86
RepQ-ViT	×	6/6	85.99	89.17
BRECQ	✓	6/6	85.27	88.99
AIQViT(ours)	✓	6/6	**87.31**	**90.60**

Table 3: Results on ModelNet40. "mACC" and "OA" are the mean of class-wise and overall accuracy (%), respectively.

Method	REC	Bits.(W/A)	c.mIoU	i.mIoU
Full-Precision	-	32/32	78.57	83.26
Min-Max	×	4/4	61.49	66.05
RepQ-ViT	×	4/4	65.78	68.43
BRECQ	✓	4/4	63.26	66.86
AIQViT(ours)	✓	4/4	**68.27**	**73.75**
Min-Max	×	6/6	72.88	76.98
RepQ-ViT	×	6/6	74.58	79.93
BRECQ	✓	6/6	73.14	79.79
AIQViT(ours)	✓	6/6	**76.99**	**81.76**

Table 4: Results on ShapeNetPart. "c.mIoU" and "i.mIoU" respectively denote the category mIoU and instance mIoU.

Method			Bits.(W/A)		
AILoC	DFQ	CL	3/3	4/4	6/6
×	×	×	14.70	46.58	61.24
✓	×	×	30.01	57.38	68.85
×	✓	×	21.43	50.28	63.64
×	×	✓	15.48	47.11	61.28
✓	✓	×	38.01	62.03	70.58
✓	×	✓	26.58	57.23	68.67
×	✓	✓	23.28	56.39	68.25
✓	✓	✓	**38.51**	**62.33**	**70.78**

Table 5: Top-1 accuracy (%) of DeiT-T on ImageNet.

the smoother optimization objective used in CL.

To validate the superiority of architectures searched in AILoC, we conduct experiments on ImageNet with DeiT-T and DeiT-S. The results are provided in Table 6. AIQViT with automatic r consistently performs better than those with fixed r. This is mainly due to the differentiable architecture search, which brings more suitable architectures for network quantization. For DeiT-S, in the cases of W4/A4 and W6/A6 quantization, models with $r = 20$ outperform those with $r = 100$ by 1.0% and 0.3%, which show that directly increasing r can not ensure better performances.

Figure 4(a) illustrates the quantization intervals learned

Model	Method	Bits.(W/A)		
		3/3	4/4	6/6
DeiT-T	r=20	33.78	59.61	68.99
	r=100	36.87	61.18	69.92
	Auto r	**38.51**	**62.33**	**70.78**
DeiT-S	r=20	50.58	69.73	77.31
	r=100	53.64	68.73	77.01
	Auto r	**55.36**	**72.75**	**78.98**

Table 6: Results of DeiT-T on ImageNet with different r. "Auto r" means setting r by network architecture search.

by DFQ. The results reveal that different layers correspond to distinct intervals, with a notable trend that shallow layers tend to have larger intervals compared to deep layers. We also investigate the impact of calibration dataset size. As shown in Figure 4(b), W4/A4 quantization exhibits better robustness to the calibration size compared to W3/A3 quantization. When only 256 samples are used, both W4/A4 and W3/A3 quantization yield subpar results, which can be attributed to the overfitting led by the limited data.

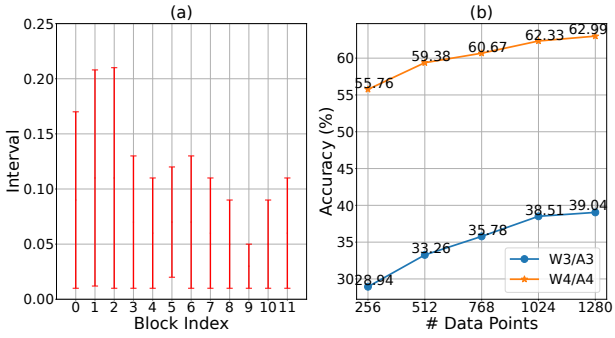

Figure 4: Visualization of the learned intervals and the influences of calibration data size. (a) Visualization of learned intervals for DeiT-T with W4/A4 quantization. (b) Effect of # data points for DeiT-T quantization on ImageNet.

Conclusion

This paper proposes AIQViT, a post-training quantization method tailored for ViTs. AIQViT employs an architecture-informed low-rank compensation mechanism, which uses the network architecture search and the CL strategy for rank calculation and stable optimization, respectively. Additionally, a simple yet effective DFQ is proposed to address the unbalanced distributions of post-Softmax activations instead of the less efficient logarithmic operations, and thus further improves the quantization efficiency. Experiments on five vision tasks demonstrate that the cumbersome ViTs can be compressed into their low-bit equivalents without compromising performances. In future works, we plan to develop novel PTQ methods for large models, hopefully extending these models to mobile applications.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. U20A20185, 62372491), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103, 2023B1515120087), the Shenzhen Science and Technology Program (No. RCYX20200714114641140).

References

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6154–6162.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*, 213–229.

Ding, Y.; Qin, H.; Yan, Q.; Chai, Z.; Liu, J.; Wei, X.; and Liu, X. 2022. Towards Accurate Post-Training Quantization for Vision Transformer. In *Proceedings of the ACM International Conference on Multimedia*, 5380–5388.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Fang, G.; Ma, X.; Song, M.; Mi, M. B.; and Wang, X. 2023. DepGraph: Towards Any Structural Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 16091–16101.

Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; and Liu, W. 2021. You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection. In *Advances in Neural Information Processing Systems*, 26183–26197.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. PCT: Point Cloud Transformer. *Computational Visual Media*, 7: 187–199.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jiang, R.; Yan, Y.; Xue, J.-H.; Chen, S.; Wang, N.; and Wang, H. 2023. Knowledge Distillation Meets Label Noise Learning: Ambiguity-Guided Mutual Label Refinery. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Jiang, R.; Yan, Y.; Xue, J.-H.; Wang, B.; and Wang, H. 2022. When Sparse Neural Network Meets Label Noise Learning: A Multistage Learning Framework. *IEEE Transactions on Neural Networks and Learning Systems*, 35: 2208–2222.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*.

Larsson, G.; Maire, M.; and Shakhnarovich, G. 2016. FractalNet: Ultra-Deep Neural Networks without Residuals. In *International Conference on Learning Representations*.

Lee, E. H.; Miyashita, D.; Chai, E.; Murmann, B.; and Wong, S. S. 2017. LogNet: Energy-Efficient Neural Networks Using Logarithmic Computation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 5900–5904.

Li, F.; Zhang, H.; Xu, H.; Liu, S.; Zhang, L.; Ni, L. M.; and Shum, H.-Y. 2023a. Mask DINO: Towards a Unified Transformer-based Framework for Object Detection and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3041–3050.

Li, Y.; Gong, R.; Tan, X.; Yang, Y.; Hu, P.; Zhang, Q.; Yu, F.; Wang, W.; and Gu, S. 2020. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. In *International Conference on Learning Representations*.

Li, Z.; Xiao, J.; Yang, L.; and Gu, Q. 2023b. Repq-ViT: Scale Reparameterization for Post-Training Quantization of Vision Transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, 17227–17236.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, 740–755.

Lin, Y.; Zhang, T.; Sun, P.; Li, Z.; and Zhou, S. 2022. FQ-ViT: Post-Training Quantization for Fully Quantized Vision Transformer. In *International Joint Conference on Artificial Intelligence*, 1173–1179.

Lin, Z.; Courbariaux, M.; Memisevic, R.; and Bengio, Y. 2015. Neural Networks with Few Multiplications. arXiv:1510.03009.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations*.

Liu, J.; Gong, R.; Wei, X.; Dong, Z.; Cai, J.; and Zhuang, B. 2023. QLLM: Accurate and Efficient Low-Bitwidth Quantization for Large Language Models. In *International Conference on Learning Representations*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021a. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE International Conference on Computer Vision*, 10012–10022.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021b. Post-Training Quantization for Vision Transformer. In *Advances in Neural Information Processing Systems*, 28092–28103.

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12309–12318.

Nagel, M.; Amjad, R. A.; van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. In *International Conference on Machine Learning*, 7197–7206.

Nagel, M.; Fournarakis, M.; Bondarenko, Y.; and Blankevoort, T. 2022. Overcoming Oscillations in Quantization-Aware Training. In *International Conference on Machine Learning*, 16318–16330.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training Data-Efficient Image Transformers & Distillation Through Attention. In *International Conference on Machine Learning*, 10347–10357.

Wang, X.; Chen, Y.; and Zhu, W. 2022. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 4555–4576.

Wei, X.; Gong, R.; Li, Y.; Liu, X.; and Yu, F. 2021. QDrop: Randomly Dropping Quantization for Extremely Low-bit Post-Training Quantization. In *International Conference on Learning Representations*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920.

Xiao, Y.; Liu, A.; Zhang, T.; Qin, H.; Guo, J.; and Liu, X. 2023. RobustMQ: Benchmarking Robustness of Quantized Models. *Visual Intelligence*, 1(1): 30.

Yan, P.; Liu, X.; Zhang, P.; and Lu, H. 2023. Learning Convolutional Multi-level Transformers for Image-based Person Re-identification. *Visual Intelligence*, 1(1): 24.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *ACM Transactions on Graphics*, 35(6): 1–12.

Yuan, Z.; Xue, C.; Chen, Y.; Wu, Q.; and Sun, G. 2022. PTQ4ViT: Post-Training Quantization for Vision Transformers with Twin Uniform Quantization. In *European Conference on Computer Vision*, 191–207.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled Knowledge Distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11953–11962.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point Transformer. In *Proceedings of the IEEE International Conference on Computer Vision*, 16259–16268.

Zhong, Y.; Hu, J.; Lin, M.; Chen, M.; and Ji, R. 2023. I&S-ViT: An Inclusive & Stable Method for Pushing the Limit of Post-Training ViTs Quantization. arXiv:2311.10126.