Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

An adaptive outlier correction quantization method for vision Transformers

Zheyang LI[†1,2], Chaoxiang LAN[2], Kai ZHANG[2], Wenming TAN[2], Ye REN[2], Jun XIAO[†‡1]

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

[2]*Hikvision Research Institute, Hangzhou 310051, China*

[†]E-mail: lizheyang@zju.edu.cn; junx@cs.zju.edu.cn

Received Nov. 11, 2024; Revision accepted July 8, 2025; Crosschecked Aug. 6, 2025

Abstract: Transformers have demonstrated considerable success across various domains but are constrained by their significant computational and memory requirements. This poses challenges for deployment on resource-constrained devices. Quantization, as an effective model compression method, can significantly reduce the operational time of Transformers on edge devices. Notably, Transformers display more substantial outliers than convolutional neural networks, leading to uneven feature distribution among different channels and tokens. To address this issue, we propose an adaptive outlier correction quantization (AOCQ) method for Transformers, which significantly alleviates the adverse effects of these outliers. AOCQ adjusts the notable discrepancies in channels and tokens across three levels: operator level, framework level, and loss level. We introduce a new operator that equivalently balances the activations across different channels and insert an extra stage to optimize the activation quantization step on the framework level. Additionally, we transfer the imbalanced activations across tokens and channels to the optimization of model weights on the loss level. Based on the theoretical study, our method can reduce the quantization error. The effectiveness of the proposed method is verified on various benchmark models and tasks. Surprisingly, DeiT-Base with 8-bit post-training quantization (PTQ) can achieve 81.57% accuracy with a 0.28 percentage point drop while enjoying 4× faster runtime. Furthermore, the weights of Swin and DeiT on several tasks, including classification and object detection, can be post-quantized to ultra-low 4 bits, with a minimal accuracy loss of 2%, while requiring nearly 8× less memory.

Key words: Transformer; Model compression and acceleration; Post-training quantization; Outlier
https://doi.org/10.1631/FITEE.2400994 **CLC number:** TP391

1 Introduction

Transformer-based architectures (Vaswani et al., 2017) have shown great power in natural language processing (NLP) tasks (Choi et al., 2018; Devlin et al., 2019). Increasingly, vision Transformers (ViTs) have also achieved competitive performance on many computer vision (CV) tasks including image classification, object detection, object segmentation, and other vision tasks recently (Carion et al., 2020; Chen ZS et al., 2021; Dosovitskiy et al., 2021; Graham et al., 2021; Yuan L et al., 2021; Touvron et al., 2021a; Dong et al., 2022; Liu Z et al., 2022b; Yu et al., 2022). Transformers consist of a number of blocks containing multi-head self-attention (MHSA) and feed-forward networks (FFNs), which enables the extraction of highly discriminative features. However, these Transformer-based models are notable for their substantial computational intensity and extensive memory requirements, posing significant challenges for deployment on resource-constrained devices (Alam et al., 2023; Chitty-Venkata et al.,

‡ Corresponding author
ORCID: Zheyang LI, https://orcid.org/0000-0002-0229-8707; Jun XIAO, https://orcid.org/0000-0003-0303-134X

2023). Consequently, there is an urgent industry requirement to compress and accelerate these Transformer-based models to facilitate broader application.

Much effort has been invested in facilitating the deployment of Transformers, including pruning (Han S et al., 2015; Zheng et al., 2022), distillation (Touvron et al., 2021b), quantization (Yao et al., 2022), and the direct design of more efficient Transformers (Choromanski et al., 2021; Yang et al., 2022). Among these methods, quantization employs low-bit precision for weight and activation values without altering the model architecture, making it particularly suitable for carefully designed efficient Transformers. There are primarily two types of quantization methods: post-training quantization (PTQ) and quantization-aware training (QAT) (Chitty-Venkata et al., 2023). Unlike the QAT method, which necessitates the entire training dataset, PTQ only requires unlabeled calibration images, thereby enabling rapid quantization and deployment. Consequently, our focus is on PTQ methods to compress Transformers. Besides, quantization and other techniques (Han S et al., 2015; Touvron et al., 2021b; Yang et al., 2022) are complementary for model acceleration.

Previous studies, such as ranking-aware (Liu ZH et al., 2021), adopt a ranking loss to make the order of the self-attention results after quantization as consistent as possible. Fully quantized vision Transformer (FQ-ViT) (Lin et al., 2022) observes serious inter-channel variation in LayerNorm inputs and extreme non-uniform distributions in attention maps, and thus uses power-of-two factor (PTF) and log-int-softmax (LIS) to reduce the performance degradation. Scale reparameterization for post-training quantization of vision Transformers (RepQ-ViT) (Li ZK et al., 2023) uses channel-wise quantization to deal with the imbalance between channels and uses a $\log\sqrt{2}$ quantizer to compress the power-law features of the latter. For inference, RepQ-ViT reparametrizes the scale factors to the layer-wise and $\log\sqrt{2}$ quantizer with minimal computations. However, we find that not only the channel variation but also the token variation is the key to limiting the accuracy of the quantized Transformers.

To mitigate the adverse effects caused by these issues, we propose an adaptive outlier correction quantization (AOCQ) method for Transformers. It reduces the imbalance of channels and tokens in three levels, operator level, framework level, and loss level, as shown in Fig. 1. Specifically, it introduces a per-channel public factor to deal with the activations before the add operator to balance the large channel discrepancy. Next, based on the BRECQ (Li YH et al., 2021) framework, we insert an extra stage to optimize the activation quantization step, which helps alleviate the effect of extreme outliers. Besides, during the quantization optimization process, a norm layer of the token dimension is equipped with the loss to achieve the balance. Further, we theoretically prove that such operations make the loss landscape smoother and the quantization error can be reduced.

To sum up, our contributions are listed as follows:

1. We revisit the fully quantized Transformers and highlight that the key problem is the huge discrepancy in channels and tokens of the Transformers, which leads to significant accuracy degradation.

2. We propose AOCQ, a simple but effective post-training quantization algorithm, which reduces the imbalance of channels and tokens at three levels: operator level, framework level, and loss level. The theoretical analysis proves that the quantization error can be reduced obviously.

3. We test the proposed framework on various tasks and benchmarks, consistently obtaining significant improvements over state-of-the-art (SOTA) post-training quantization methods.

2 Related works

2.1 ViT

Transformers initially showed great success in sequence modeling and neural machine translation (Vaswani et al., 2017). Owing to the high flexibility, Transformers have been applied to multi-modal and speech tasks. Carion et al. (2020) trained an end-to-end Transformer-based detector DEtection TRansformer (DETR), achieving results comparable to convolutional neural networks (CNNs). Meanwhile, Dosovitskiy et al. (2021) proposed a pure Transformer to treat image patches as sequences for image classification. Besides, Transformers have been applied to a range of computer vision (CV) problems, including object detection (Zhu et al., 2021; Li F et al., 2022; Liu SL et al., 2022; Zhang H et al.,

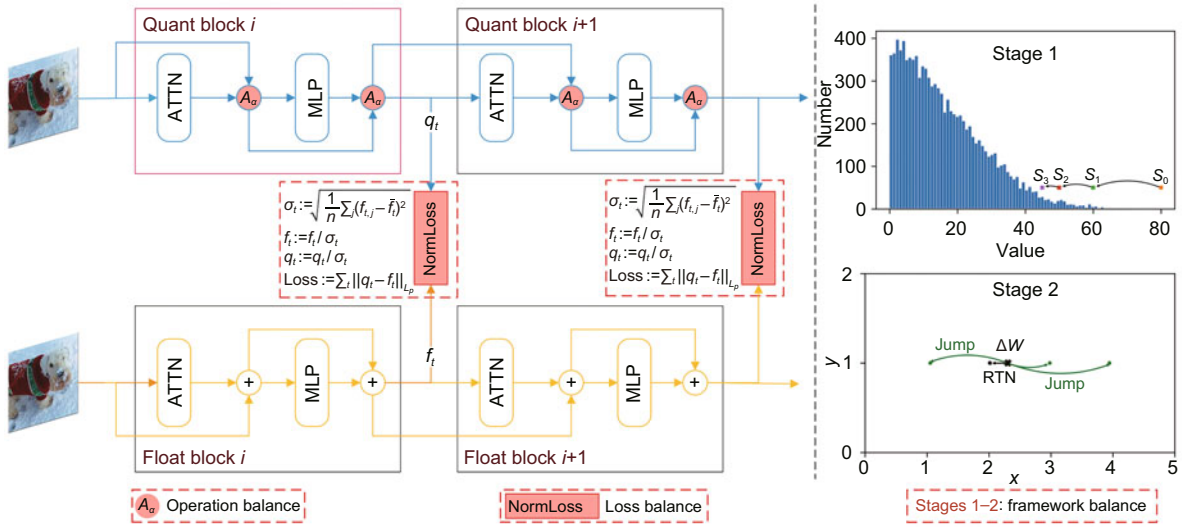

Fig. 1 Framework of the AOCQ. AOCQ deals with the outliers adaptively in three levels: operator level, framework level, and loss level. It uses Add$_\alpha$ and LN$_\alpha$ to diminish the imbalance between channels. A two-stage optimization method has been used to optimize quantization scale and the rounding direction. In the first stage, the quantization scale s_0 of activation is iteratively refined to s_4 through successive optimization steps. After that, the rounding direction of the weights is optimized in the second stage. The token norm function suppresses the extreme tokens in the block-wise reconstruction loss

2023), semantic segmentation (Li F et al., 2023), image processing (Liang et al., 2021), and video understanding (Liu Z et al., 2022a). In the following years, ViT models sprang up like mushrooms in the following years. Token-to-Token (T2T) Transformer (Yuan L et al., 2021) is used to improve the local relations between adjacent tokens before entering the backbone network. A family of shifted window-based Transformers, including Swin Transformer (Liu Z et al., 2021) and Swin Transformer V2 (Liu Z et al., 2022b), uses local shifted window-based attention to construct a hierarchical architecture and achieve strong baselines in many vision tasks. Ding et al. (2022) proposed a new framework, dual vision Transformer (DaViT), which combines mixed blocks of channel attention and spatial attention. Inception Transformer (Si et al., 2022) captures both the high and low frequencies in inputs by applying the Inception mixer of parallel convolution modules and self-attention modules. Hierarchical vision Transformer (HiVit) (Zhang XS et al., 2023) comes up with a plain and efficient Transformer backbone and shows a clear advantage on several benchmarks. Recently, faster vision Transformer (FasterViT) (Hatamizadeh et al., 2024) and FLatten Transformer (Han DC et al., 2023) have made efforts to make the attention more efficient by the hierarchical attention and depthwise attention. In conclusion, deploying various Transformers on resource-limited devices is still urgent in the industry.

2.2 QAT and PTQ

Model compression is an effective strategy for reducing the development costs of deep neural networks (DNNs). These approaches contain pruning (Han S et al., 2015; Zheng et al., 2022), distillation (Hinton et al., 2015; Sanh et al., 2019), quantization (Yao et al., 2022), efficient architecture design (Chen MH et al., 2021; Choromanski et al., 2021), Transformer function and kernel optimization (Hong et al., 2023), and hardware acceleration optimization (Ham et al., 2020; Qu et al., 2022). Quantization is an effective and promising way to accelerate neural networks. There are two types of quantization methods, QAT methods (Choi et al., 2018; Zafrir et al., 2019) and PTQ methods (Yuan ZH et al., 2022).

QAT improves the accuracy of the quantized model by finetuning on the training datasets with a straight-through estimator (STE) (Bengio et al., 2013), which is used to approximate the gradient. Similarly, Gong et al. (2019) introduced a differentiable tanh function to simulate the gradients in the training process. Learned step size quantization (LSQ) (Esser et al., 2020) pays attention to

optimizing the quantization intervals, achieving good performance on several benchmarks.

PTQ methods quantize networks with a small number of unlabeled images, which is significantly faster. Notably, Nagel et al. (2020) proposed AdaRound to treat the rounding task as a quadratic unconstrained binary optimization problem by approximating the task loss with a Taylor series expansion. BRECQ (Li YH et al., 2021) uses the Fisher information matrix (FIM) to assign each pre-activation with an importance measure during reconstruction, which achieves a 4-bit ResNet of accuracy 76.29, with only a 0.7 drop. Quantization has been a popular method for model compression, especially for CNNs, in the past years.

2.3 PTQ for ViTs

Several quantization methods (Li YH et al., 2021; Liu ZH et al., 2021; Li ZK et al., 2022, 2024) have been applied to ViTs to achieve faster inference. However, the outliers in Transformers are larger than those in CNNs, making it more challenging to quantize the models. Liu ZH et al. (2021) firstly proposed a PTQ method to quantize the ViT. A ranking loss was proposed to keep the relative order of the self-attention results after quantization. Besides, the models were not fully quantized, and some parts need to retain floating-point units in the hardware, leading to hardware inefficiency. FQ-ViT (Lin et al., 2022) proposes PTF and LIS to handle the serious inter-channel variation, which achieves fully-quantized ViTs. However, PTF and LIS are still unfriendly for some resource-limited devices. PTQ4ViT (Yuan ZH et al., 2022) introduces a twin uniform quantization method and proposes a Hessian guided metric to evaluate different scaling factors, which achieves better performance. RepQ-ViT (Li ZK et al., 2023) adopts a $\log\sqrt{2}$ quantizer to diminish the imbalance between channels. Actually, based on the observation, we find that both channel variation and the token variation have an influence on the accuracy of the quantized Transformers.

3 Method

3.1 Outliers in the Transformer

A standard Transformer consists of a series of blocks which contain MHSA and FFN. The attention can be formulated as

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V}, \quad (1)$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the query, key, and value matrix, respectively. d is the hidden size of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. MHSA contains multiple heads, each of which concurrently computes attention operations. All heads are concatenated depthwise and linearly transformed to the output by a fully connected (FC) layer.

The MHSA can be formulated as

$$\begin{cases}\text{MHSA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=[\text{head}_1,\text{head}_2,...,\text{head}_h]\boldsymbol{W}^{\mathrm{O}},\\ \text{head}_i=\text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^{\mathrm{Q}},\boldsymbol{K}\boldsymbol{W}_i^{\mathrm{K}},\boldsymbol{V}\boldsymbol{W}_i^{\mathrm{V}}),\end{cases} \quad (2)$$

where $\boldsymbol{W}^{\mathrm{Q}},\boldsymbol{W}^{\mathrm{K}},\boldsymbol{W}^{\mathrm{V}}$, and $\boldsymbol{W}^{\mathrm{O}}$ are the corresponding weight parameters. head_i represents the i^{th} head.

Following MHSA, FFN is constructed by stacking two FC layers with activation functions, such as rectified linear unit (ReLU) or Gaussian error linear unit (GeLU). The FFN can be formulated as

$$\text{FFN}(\boldsymbol{x})=\phi(\boldsymbol{x}\boldsymbol{W}_1+\boldsymbol{b}_1)\boldsymbol{W}_2+\boldsymbol{b}_2, \quad (3)$$

where $\boldsymbol{x}$ is the input feature and ϕ stands for the activation function. $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$ represent the weights and biases of different FC layers, respectively. As a result, the whole block consists of an alternative MHSA and FFN. Besides, LayerNorm (LN) (Ba et al., 2016) is adopted to normalize activations, which helps avoid extreme outliers.

The whole Transformer block can be written as

$$\begin{cases}\boldsymbol{z}=\text{MHSA}(\text{LN}(\boldsymbol{x}))+\boldsymbol{x},\\ \boldsymbol{y}=\text{FFN}(\text{LN}(\boldsymbol{z}))+\boldsymbol{z}.\end{cases} \quad (4)$$

We found that as the depth increases, both the activation values and their variances in the Transformer will gradually increase, as shown in Fig. 2. In addition, this phenomenon has been mentioned in other related work (Shen et al., 2020). Lemma 1 theoretically supports this phenomenon. The increased activation values and their variances bring great challenge to quantize Transformers. The direct quantization results of different depths' Transformers are shown in Table 1. The more Transformer block model contains, the larger the accuracy drop.

Lemma 1 Let l be the depth of the model, d_l stands for the channel dimension, $\boldsymbol{W}^{\mathrm{Q}}$ and $\boldsymbol{W}^{\mathrm{K}}$ be

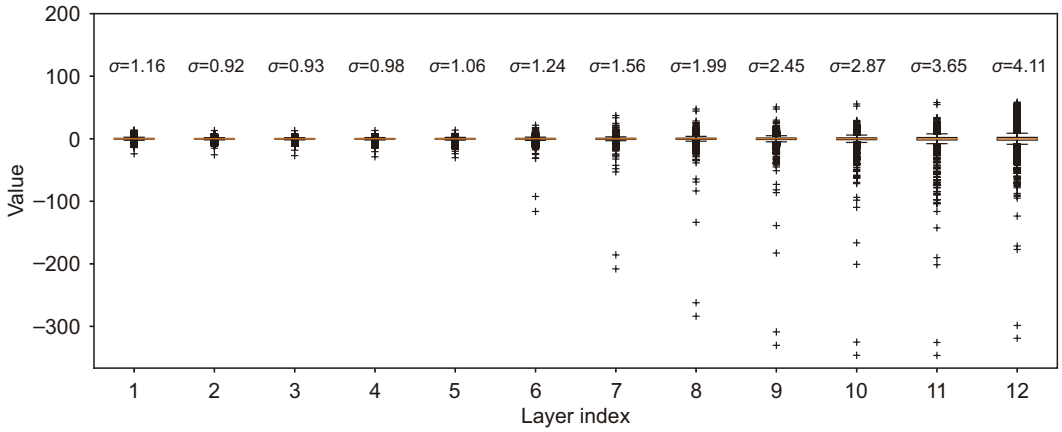

Fig. 2 Boxplot of different layers' activations. As the network deepens, the average and maximum activation values also become larger and larger

Table 1 Quantization results for Swin Transformers without any tricks

Model	Number of blocks	Top-1 accuracy with FP32 (%)	Top-1 accuracy with W8A8 (%)	δ
Swin-Tiny	12	81.35	80.73	–0.62
Swin-Small	24	83.30	72.33	–10.97
Swin-Base	36	83.60	46.13	–37.47

All the operations including FC, softmax, matmul, add, LN, and GeLU have been quantized. The outliers in deeper models have a worse impact on the performance. FP32 means that the weights and activations of the model remain in a full-precision form. W8A8 means that the weights and activations are both quantized to 8 bits. δ means the accuracy drop of W8A8 compared to FP32

initialized independently, and $\boldsymbol{W}^{\mathrm{V}} \sim N(0,1)$. Moreover, for the Post-LN Transformer, we have

$$E(||\boldsymbol{X}_{(l,i)}^{\mathrm{post}}||_2^2) = \frac{3}{2}d_l, \forall l > 0, i, \tag{5}$$

where $E(||\boldsymbol{X}_{(l,i)}^{\mathrm{post}}||_2^2)$ denotes the expectation of the output features $\boldsymbol{X}_{(l,i)}^{\mathrm{post}}$. For the Pre-LN Transformer, we have

$$(1+\frac{1}{2}l)d_l \leq E(||\boldsymbol{X}_{(l,i)}^{\mathrm{pre}}||_2^2) \leq (1+\frac{3}{2}l)d_l, \forall l > 0, i. \tag{6}$$

That is to say, as the depth increases, both the activation values and their variances of Transformers will gradually increase. The proof of Post-LN Transformer is obvious. Therefore, we focus on the Pre-LN Transformer.

Proof For simplicity, the computation process of Pre-LN Transformer can be listed as

$$\begin{cases} \boldsymbol{x}_1 = \mathrm{LN}(\boldsymbol{x}_0), \\ \boldsymbol{x}_2 = \mathrm{MHSA}(\boldsymbol{x}_1, \mathbf{bias}), \\ \boldsymbol{x}_3 = \boldsymbol{x}_0 + \boldsymbol{x}_2, \\ \boldsymbol{x}_4 = \mathrm{LN}(\boldsymbol{x}_3), \\ \boldsymbol{x}_5 = \mathrm{FFN}(\boldsymbol{x}_4), \\ \boldsymbol{x}_6 = \boldsymbol{x}_3 + \boldsymbol{x}_5. \end{cases} \tag{7}$$

Let $\boldsymbol{x}_i = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ \ldots, \boldsymbol{t}_n)$, n be the token number, and $\boldsymbol{t}_i$ be a token. We have

$$\begin{aligned} ||\boldsymbol{x}_1||_2^2 &= \sum_{i=1}^{d} (\boldsymbol{t}_{1,i} - \boldsymbol{\mu}_1)^2 / \sigma_1^2, \\ &= \frac{d}{\sigma_1^2}\sigma_1^2 = d, \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_1$ and σ_1^2 are the mean and std of $\boldsymbol{x}_1$, which are calculated by LN. d means the channel dimension. Similarly,

$$E(||\boldsymbol{x}_4||_2^2) = d. \tag{9}$$

As $\boldsymbol{W}^{\mathrm{Q}}$ and $\boldsymbol{W}^{\mathrm{K}}$ are initialized independently, the values of softmax input features are similar and the value of the softmax output is equal to $1/n$. We have

$$\begin{aligned} \boldsymbol{x}_2 &= \mathrm{MHSA}(\boldsymbol{x}_1, \mathbf{bias}), \\ &= \frac{1}{n}(1, 1, ..., 1)\boldsymbol{x}_1\boldsymbol{W}^{\mathrm{V}}, \\ &= \frac{1}{n}\sum \boldsymbol{t}_i \boldsymbol{W}^{\mathrm{V}}. \end{aligned} \tag{10}$$

At the same time, when $\boldsymbol{W} \sim N(0,1)$, it can be

arrived at

$$\begin{aligned}
&E(||\boldsymbol{tW}||_2^2)\\
&=E\left(\sum_i^d(\boldsymbol{tW}_i)^2\right)\\
&=E\left(\sum_i\sum_j\boldsymbol{tW}_i\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{t}^{\mathrm{T}}\right)\\
&=E\left(\sum_i\boldsymbol{tW}_i\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{t}^{\mathrm{T}}+\sum_{i\neq j}\sum_j\boldsymbol{tW}_i\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{t}^{\mathrm{T}}\right)\\
&=\sum_i E(\boldsymbol{tW}_i\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{t}^{\mathrm{T}})\\
&=\sum_i\boldsymbol{t}E(\boldsymbol{W}_i\boldsymbol{W}_i^{\mathrm{T}})\boldsymbol{t}^{\mathrm{T}}\\
&=\sum_i^d\frac{1}{d}\boldsymbol{tt}^{\mathrm{T}}=||\boldsymbol{t}||_2^2.
\end{aligned}\tag{11}$$

According to Eq. (11), it can be calculated as

$$\begin{aligned}
E(||\boldsymbol{x}_2||_2^2)&=E\left(\left\|\left(\frac{1}{n}\sum\boldsymbol{t}_i\right)\boldsymbol{W}^{\mathrm{V}}\right\|_2^2\right)\\
&=E\left(\left\|\frac{1}{n}\sum\boldsymbol{t}_i\right\|_2^2\right)\\
&\leq\frac{1}{n}\sum E(||\boldsymbol{t}_i||_2^2)\\
&=\frac{1}{n}\sum E(||\boldsymbol{t}_i||_2^2)=d.
\end{aligned}\tag{12}$$

Because we assume that the token is independent and the parameters' expectation is 0, we have

$$\begin{aligned}
E(\boldsymbol{x}_2\boldsymbol{x}_0^{\mathrm{T}})&=\frac{1}{n}E\left(\sum_{i=1}^n\boldsymbol{t}_i\boldsymbol{W}^{\mathrm{V}}\boldsymbol{x}_0^{\mathrm{T}}\right)\\
&=\frac{1}{n}\sum_{i=1}^n E(\boldsymbol{t}_i\boldsymbol{W}^{\mathrm{V}}\boldsymbol{x}_0^{\mathrm{T}})\\
&=\frac{1}{n}\sum_{i=1}^n E(E(\boldsymbol{t}_i\boldsymbol{W}^{\mathrm{V}}\boldsymbol{x}_0^{\mathrm{T}})|\boldsymbol{x}_0^{\mathrm{T}})\\
&=0.
\end{aligned}\tag{13}$$

Therefore,

$$\begin{aligned}
E(||\boldsymbol{x}_3||_2^2)&=E(||\boldsymbol{x}_0+\boldsymbol{x}_2||_2^2)\\
&=E(||\boldsymbol{x}_0||_2^2)+E(||\boldsymbol{x}_2||_2^2)+2E(\boldsymbol{x}_2\boldsymbol{x}_0^{\mathrm{T}})\\
&\leq E(||\boldsymbol{x}_0||_2^2)+d.
\end{aligned}\tag{14}$$

That means

$$E(||\boldsymbol{x}_0||_2^2)\leq E(||\boldsymbol{x}_3||_2^2)\leq E(||\boldsymbol{x}_0||_2^2)+d.\tag{15}$$

We can obtain the following equation from the same process:

$$E(||\boldsymbol{x}_6||_2^2)=E(||\boldsymbol{x}_3||_2^2)+\frac{d}{2}.\tag{16}$$

Combining the above formulations, we obtain

$$E(||\boldsymbol{x}_0||_2^2)+\frac{d}{2}\leq E(||\boldsymbol{x}_6||_2^2)\leq E(||\boldsymbol{x}_0||_2^2)+\frac{3}{2}d.\tag{17}$$

This completes the proof.

Furthermore, there are serious inter-channel and inter-token variations in ViTs, which brings unacceptable quantization errors, as shown in Fig. 3. The large range of inter-channel and inter-token variations may make Transformer extract more powerful representation. However, preserving the feature representation and minimizing the quantization error are the key challenges for PTQ.

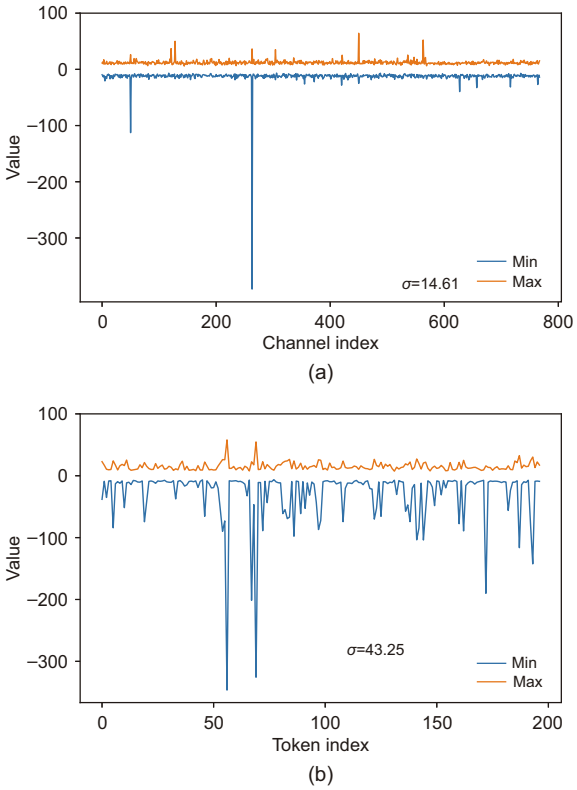

Fig. 3 Inter-channel variation in Transformers (a) and inter-token variation in Transformers (b), demonstrating that the outliers exist in different channels and tokens

3.2 Operation balance

In this subsection, we introduce two new operators named Add_α and LN_α, where the add operation

and LN operation are equipped with a per-channel factor α. The factors are used to deal with the inter-channel variation directly, which are extracted from the channel-imbalanced features. Applying Add_α and LN_α, the activations become smooth and can be quantized more easily, as shown in Fig. 4. As a result, the quantization error is reduced obviously.

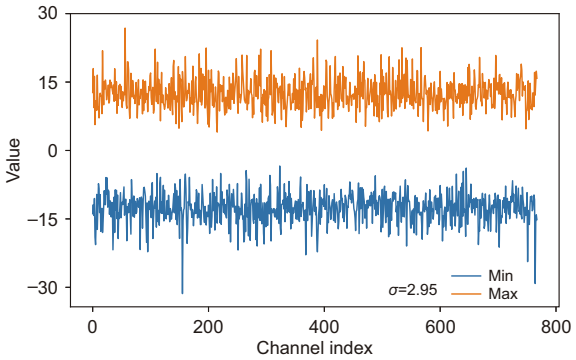

Fig. 4 The activations with Add_α. The comparison of Fig. 3a demonstrates that the outliers in different channels have been diminished largely with Add_α, where std is reduced from 14.61 to 2.95

In Fig. 5, we plot the density distributions of activations. This visualization clearly shows that the original distribution exhibits dual-peak characteristics across channels. Given that quantization must trade-off range and precision, simultaneously considering these disparate peaks inherently leads to significant quantization error. After applying our channel balancing method, the visualization demonstrates a more uniform distribution. Specifically, the quantization error is reduced from 0.753 to 0.109, effectively demonstrating the method's efficacy.

Concretely, Add_α and LN_α can be written as

$$\text{Add}_\alpha(\boldsymbol{x}_1, \boldsymbol{x}_2) = a\boldsymbol{x}_1 + b\boldsymbol{x}_2 + c, \tag{18}$$

$$\text{LN}_\alpha(\boldsymbol{x}) = \text{LN}(e\boldsymbol{x}), \tag{19}$$

where a, b, c, and e are calculated according to the calibration set.

As a result, the whole calculation process of the block can be listed as

$$\begin{cases} \boldsymbol{z} = \text{Add}_\alpha(\text{MHSA}(\text{LN}_\alpha(\boldsymbol{x})), \boldsymbol{x}), \\ \boldsymbol{y} = \text{Add}_\alpha(\text{FFN}(\text{LN}_\alpha(\boldsymbol{z})), \boldsymbol{z}). \end{cases} \tag{20}$$

In conclusion, it is mathematically equivalent to Eq. (4), as shown in the following proof:

Proof For simplicity, the eltwise-add operation in MHSA can be written as

$$\boldsymbol{y} = \boldsymbol{x}_1 + \boldsymbol{x}_2, \tag{21}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ stand for the inputs, while $\boldsymbol{y}$ is the output. Equipped with Add_α, the new calculating process is

$$\boldsymbol{x}_1^\alpha = \frac{\boldsymbol{x}_1 - c_1}{a_1}, \tag{22}$$

$$\boldsymbol{x}_2^\alpha = \frac{\boldsymbol{x}_2 - c_2}{a_2}, \tag{23}$$

$$\begin{aligned} \boldsymbol{y} &= \text{Add}_\alpha(\boldsymbol{x}_1^\alpha, \boldsymbol{x}_2^\alpha) \\ &= a\boldsymbol{x}_1^\alpha + b\boldsymbol{x}_2^\alpha + c \\ &= a\frac{\boldsymbol{x}_1 - c_1}{a_1} + b\frac{\boldsymbol{x}_2 - c_2}{a_2} + c. \end{aligned} \tag{24}$$

Note $\boldsymbol{x}_1^\alpha$ and $\boldsymbol{x}_2^\alpha$ are smoother than $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ at the channel level. We quantize $\boldsymbol{x}_1^\alpha$ and $\boldsymbol{x}_2^\alpha$ instead of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. As a result, $\boldsymbol{x}_1^\alpha$ and $\boldsymbol{x}_2^\alpha$ lead to lower quantization error. a_1, a_2, and c are the channel-level common factors. Let $a = a_1$, $b = a_2$, and $c = c_1 + c_2$, it can be transformed as

$$\begin{aligned} \boldsymbol{y} &= a_1\frac{\boldsymbol{x}_1 - c_1}{a_1} + a_2\frac{\boldsymbol{x}_2 - c_2}{a_2} + c_1 + c_2 \\ &= \boldsymbol{x}_1 + \boldsymbol{x}_2. \end{aligned} \tag{25}$$

It proved that the result in Eq. (25) is equal to that in Eq. (21). Similarly, LN in the Transformer can be written as

$$\boldsymbol{y} = \frac{\boldsymbol{x}_1 - E(\boldsymbol{x}_1)}{\sqrt{\text{Var}(\boldsymbol{x}_1)}}, \tag{26}$$

where Var represents the variance.

With LN_α, the new calculating process is

$$\boldsymbol{x}_1^\alpha = \frac{\boldsymbol{x}_1 - c_1}{a_1}, \tag{27}$$

$$\begin{aligned} \boldsymbol{y} &= \text{LN}_\alpha(\boldsymbol{x}_1^\alpha) \\ &= \text{LN}(a\boldsymbol{x}_1^\alpha) \\ &= \frac{a\boldsymbol{x}_1^\alpha - E(a\boldsymbol{x}_1^\alpha)}{\text{Var}(a\boldsymbol{x}_1^\alpha)} \\ &= \frac{a\frac{\boldsymbol{x}_1 - c_1}{a_1} - E(a\frac{\boldsymbol{x}_1 - c_1}{a_1})}{\text{Var}(a\frac{\boldsymbol{x}_1 - c_1}{a_1})}. \end{aligned} \tag{28}$$

Let $a = a_1$. It can be

$$\begin{aligned} \boldsymbol{y} &= \frac{a_1\frac{\boldsymbol{x}_1 - c_1}{a_1} - E(a_1\frac{\boldsymbol{x}_1 - c_1}{a_1})}{\text{Var}(a_1\frac{\boldsymbol{x}_1 - c_1}{a_1})} \\ &= \frac{\boldsymbol{x}_1 - c - E(\boldsymbol{x}_1 - c)}{\text{Var}(\boldsymbol{x}_1 - c)} \\ &= \frac{\boldsymbol{x}_1 - E(\boldsymbol{x}_1)}{\text{Var}(\boldsymbol{x}_1)}. \end{aligned} \tag{29}$$

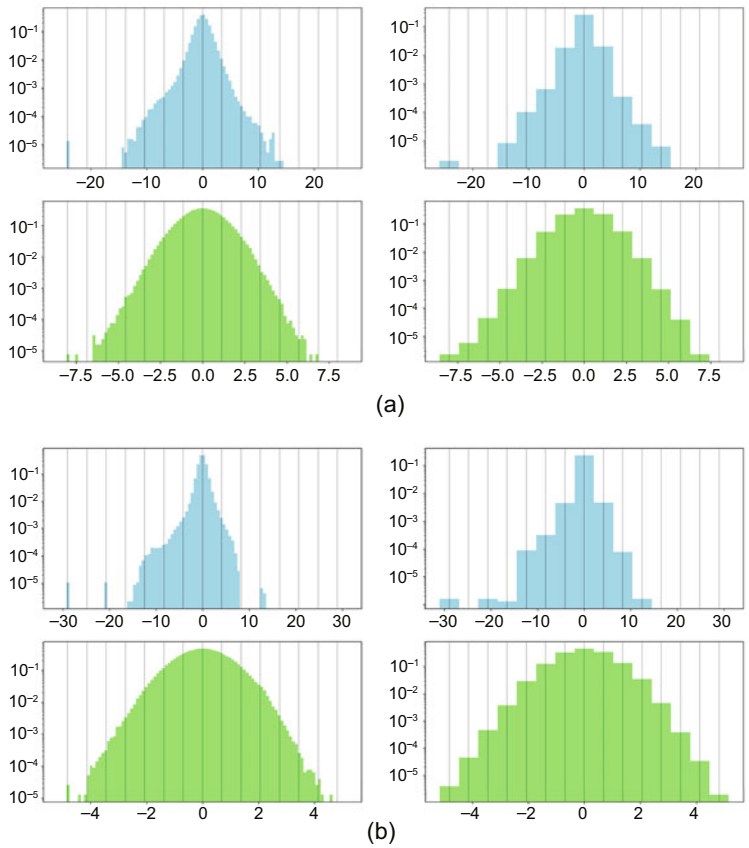

Fig. 5 Comparison of the density distribution: (a) BasePTQ; (b) our method. The x-axis represents the value interval, while the y-axis represents the count of activations falling within each interval. The left and right columns represent the original and quantized distributions, respectively

It shows that Eq. (29) is equal to Eq. (26). Similarly, we quantize $\boldsymbol{x}_1^{\alpha}$ instead of $\boldsymbol{x}_1$, to achieve a better result.

This completes the proof.

As shown in Fig. 1, Add_{α} and LN_{α} serve as drop-in replacements for the original Add and LN operations respectively, maintaining performance parity with the original models. The per-channel factors can be simply derived using either per-channel maximum values or computed based on the distribution-aware methods.

3.3 Framework balance

We propose a two-stage optimization method that more effectively suppresses the impact of outliers. Specifically, in the first stage, we construct a block and optimize the quantization scale of the activation values within the block. The optimal scale is automatically determined based on the minimum loss, which will suppress the impact of some outliers. As depicted in Fig. 1, the quantization scale s_0 is iteratively refined to s_4 through successive optimization steps. In the second stage, we optimize the rounding direction of the weights following BRECQ based on the reconstruction loss. The details of the two-stage optimization approach can be seen in Fig. 1. Compared with BRECQ, the two-stage optimization framework can deal with the adverse effects of outliers.

3.4 Loss balance

In the block-wise reconstruction loss, outliers greatly affect the optimization direction of the loss. After using two new operators (Add_{α} and LN_{α}) and a two-stage optimization approach, outliers are mainly concentrated between various tokens. Those tokens with very large values have the greatest impact on the final loss, thereby dominating the overall optimization direction. This results in a large number of tokens with seemingly small values not playing a

role in the optimization. To balance the imbalance between different tokens, we introduce normalization between tokens in the calculation of the loss, which is called norm-loss. The calculation process of norm-loss has also been shown in Fig. 1.

Note that norm-loss is applied only during the quantization reconstruction phase. It is not used during the actual inference. Our objective with norm-loss is to consider the importance of all tokens for more balanced quantization parameter learning. Furthermore, we visualize the attention maps derived using the norm-loss. As shown in Fig. 6, the attention map generated by quantization with norm-loss closely resembles the original counterpart.

Furthermore, we take advantage of the condition number of FIM to prove the effectiveness of our method theoretically. The condition number represents how easy it is to optimize the corresponding layer (LeCun et al., 1990, 2012). However, the global FIM is difficult due to the high memory and computational costs. Under mild conditions, the FIM is approximately computed by using the Kronecker

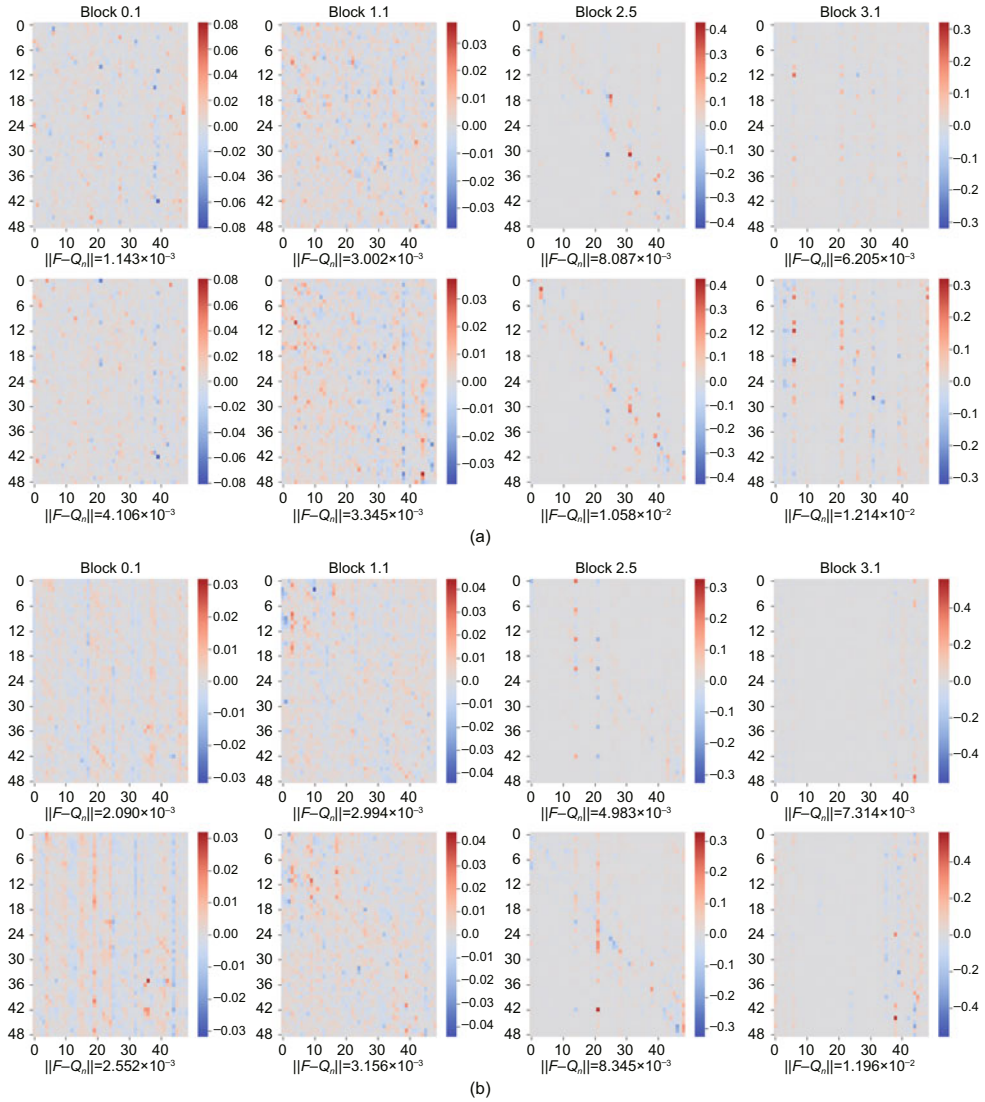

Fig. 6 Attention map errors: (a) ImageNet; (b) CIFAR-10. The attention map represents the scores between query and key vectors. We visualized the divergence of attention maps between the original model and the different quantized models. A smaller divergence indicates a higher similarity. Besides, deeper layers exhibit lower error levels in attention maps, critically impacting the final performance. When comparing CIFAR-10 and ImageNet tasks, attention maps on CIFAR-10 demonstrate smaller deviations. This aligns with the superior quantization performance observed in CIFAR-10, showing less quantization degradation in accuracy

product (K-FAC) (Grosse and Martens, 2016; Ba et al., 2017; Huang et al., 2020). As a result, the full FIM can be replaced by a block diagonal matrix, $\boldsymbol{F} = \mathrm{diag}(F_1, F_2, \ ..., F_k)$. F_k can be computed as

$$\begin{aligned} F_k &= E(\mathrm{vec}(\boldsymbol{W}_\mathrm{g})\mathrm{vec}(\boldsymbol{W}_\mathrm{g})^\mathrm{T}) \\ &= E(\mathrm{vec}(\boldsymbol{g}\boldsymbol{x}^\mathrm{T})\mathrm{vec}(\boldsymbol{g}\boldsymbol{x}^\mathrm{T})^\mathrm{T}) \\ &= E((\boldsymbol{x}\times\boldsymbol{g})(\boldsymbol{x}\times\boldsymbol{g})^\mathrm{T}) \\ &= E((\boldsymbol{x}\times\boldsymbol{x}^\mathrm{T})(\boldsymbol{g}\times\boldsymbol{g}^\mathrm{T})) \\ &\approx E(\boldsymbol{x}\boldsymbol{x}^\mathrm{T})E(\boldsymbol{g}\boldsymbol{g}^\mathrm{T}), \end{aligned} \tag{30}$$

where $\boldsymbol{x}$ is the k^{th} input and $\boldsymbol{g}$ means the gradient of $\boldsymbol{x}$. $\boldsymbol{W}_\mathrm{g}$ is the gradient of $\boldsymbol{W}$. After the Fisher information matrix is computed, the eigenvalue and the condition number can be calculated.

As shown in Fig. 7, in the shallow layers of the network, the condition number difference using norm-loss is not significant due to the non-significant outliers in the activation values. However, as the depth of the network increases, the abnormalities in the activation values become more pronounced. As a result, the condition number of norm-loss significantly decreases. This indicates that using norm-loss can effectively reduce the adverse impact brought by outliers.

4 Experiments

In this section, we present results on ViTs for image classification and object detection. We demonstrate our methods on several tasks and various models including DeiT family models and Swin family models. In the end, ablation studies have been conducted on the proposed methods.

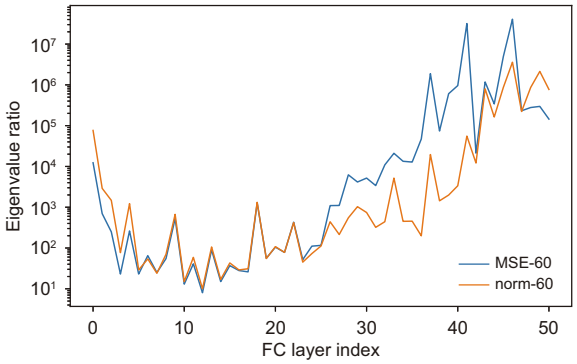

Fig. 7 Condition number of different FC layers in the Transformer. The orange and blue lines represent the condition number with and without the norm-loss, respectively. In the deep layers of the network (layer number within 28–40), the condition number of norm-loss significantly decreases compared with that of regular loss, which means that it is more easily optimized. References to color refer to the online version of this figure

4.1 Experimental settings

ImageNet-1K is one of the most commonly used image classification dataset that consists of 1000 classes. We randomly sample 1000 training images from ImageNet as the calibration data, and use the validation set to evaluate the performance for the classification task. Similarly, COCO has been the preferred dataset to verify the detection performance. All layers' weights and activations, including the first projection layer and the last prediction layer, have been quantized. Similar to Lin et al. (2022), softmax and layer normalization layers in ViTs have been quantized. In the block-wise reconstruction process, the number of steps has been set to 1000. The input resolution is set to 224×224 for classification, while the input resolution is 1333×800 for object detection. The adopted model is available from the official address, including Swin, DeiT, and Faster R-CNN (Ren et al., 2015).

4.2 Comparison with SOTA methods

The proposed method has been compared with other PTQ methods, including PTQ4ViT, BasePTQ (Yuan ZH et al., 2022), and FQ-ViT (Lin et al., 2022).

4.2.1 Image classification

Different ViT architectures including DeiT and Swin have been evaluated on ImageNet. The results are shown in Table 2. From the table, we observe that our method with 4-bit quantization outperforms all other methods across models of different scales. With weight 8-bit quantization, PTQ4ViT achieves the best performance on Swin-Base, Swin-Small, and DeiT-small, while our method attains the best on Swin-Tiny, DeiT-Tiny, and DeiT-Base. Note that PTQ4ViT ignores the quantization of add operations and our method, FQ-ViT is fully quantized. With 8-bit quantization, AOCQ's performance drop is generally within 0.7 percentage points (PPs) on ViTs, and with 4-bit quantization, the performance drop is within 2.1 PPs. We notice that the larger the model, the smaller the performance drop, indicating that larger models are less sensitive to quantization

Table 2 Comparison of the performance of the proposed PTQ method AOCQ with other SOTA methods for image classification on ImageNet

Model	Method	W/A	Model size (M)	Accuracy (%)	δ	W/A	Model size (M)	Accuracy (%)	δ
Swin-Tiny		32/32	112	81.35		32/32	112	81.35	
	BasePTQ*	8/8	28	80.96	–0.39	4/8	14	66.41	–14.94
	PTQ4ViT*	8/8	28	81.24	–0.11	4/8	14	77.96	–3.39
	FQ-ViT	8/8	28	80.51	–0.84	4/8	14	78.23	–3.12
	AOCQ (ours)	8/8	28	**81.25**	**–0.10**	4/8	14	**80.87**	**–0.48**
Swin-Small		32/32	195	83.20		32/32	195	83.20	
	BasePTQ*	8/8	49	82.75	–0.45	4/8	24	78.43	–4.77
	PTQ4ViT*	8/8	49	**83.10**	**–0.10**	4/8	24	80.25	–2.95
	FQ-ViT	8/8	49	82.71	–0.49	4/8	24	81.62	–1.58
	AOCQ (ours)	8/8	49	83.00	–0.20	4/8	24	**82.80**	**–0.40**
Swin-Base		32/32	345	83.60		32/32	345	83.60	
	BasePTQ*	8/8	86	83.32	–0.28	4/8	43	78.43	–5.17
	PTQ4ViT*	8/8	86	**83.56**	**–0.06**	4/8	43	80.25	–3.35
	FQ-ViT	8/8	86	82.97	–0.63	4/8	43	81.62	–1.98
	AOCQ (ours)	8/8	86	83.50	–0.10	4/8	43	**83.42**	**–0.18**
DeiT-Tiny		32/32	22	72.21		32/32	22	72.21	
	BasePTQ*	8/8	5.6	71.28	–0.93	4/8	2.8	56.58	–15.63
	PTQ4ViT*	8/8	5.6	71.57	–0.64	4/8	2.8	66.70	–5.51
	FQ-ViT	8/8	5.6	71.61	–0.60	4/8	2.8	65.78	–6.43
	AOCQ (ours)	8/8	5.6	**71.74**	**–0.47**	4/8	2.8	**70.18**	**–2.03**
DeiT-Small		32/32	86	79.85		32/32	86	79.85	
	BasePTQ*	8/8	22	77.65	–2.20	4/8	11	64.62	–15.23
	PTQ4ViT*	8/8	22	**79.47**	**–0.38**	4/8	11	77.03	–2.82
	FQ-ViT	8/8	22	79.17	–0.68	4/8	11	75.65	–4.20
	AOCQ (ours)	8/8	22	79.19	–0.66	4/8	11	**78.60**	**–1.25**
DeiT-Base		32/32	338	81.85		32/32	338	81.85	
	BasePTQ*	8/8	84	80.94	–0.91	4/8	42	74.27	–7.58
	PTQ4ViT*	8/8	84	81.48	–0.37	4/8	42	79.59	–2.26
	FQ-ViT	8/8	84	81.20	–0.65	4/8	42	79.36	–2.49
	AOCQ(ours)	8/8	84	**81.57**	**–0.28**	4/8	42	**81.48**	**–0.37**

The input resolution is set to 224×224. W/A means weight/activation. * means that the quantization is only used for matrix multiplication, where add and other operations maintain float. M stands for million. BasePTQ is from Yuan ZH et al. (2022). The best results are in bold, and the second-best results are underlined

and their performance is more stable. Comparing Swin and DeiT, we find that DeiT experiences a greater performance drop. The reason may be that Swin computes the self-attention locally within non-overlapping windows and the activation range is smaller.

Additionally, we conduct experiments on the downstream task using the CIFAR-10 dataset. Experimental results demonstrate that our method achieves a 1.84 PPs gain over the BasePTQ method in Table 3. This demonstrates the efficacy of our approach on downstream datasets.

4.2.2 Object detection

To further verify the generalizability of our method, we also conducted experiments on the object detection task. Table 4 shows that our method has a performance drop of 0.4 PPs when compressed by 4× (8-bit quantization), and the performance only drops by 0.7 PPs when compressed by 8× (4-bit quantization). At the same time, it can be observed that the improvement of the proposed method is more significant at 4-bit quantization. All the above results demonstrate the effectiveness of our method.

4.3 Ablation studies

We conducted ablation studies on our method, including various components of AOCQ and the hyperparameters used within the method, e.g., the numbers of steps and samples in the reconstruction process. These experiments facilitate a deeper investigation and prove the effectiveness of our method.

4.3.1 Effect of different components

In this subsection, we individually verified the effectiveness of the three strategies (operation balance, framework balance, and loss balance) on classification and detection tasks. From Table 5, it shows that adopting 4-bit quantization directly results in a performance drop of more than 14 PPs. By merely employing the strategy of operation balance, the performance can be significantly improved by 12.52 PPs. Furthermore, by adopting a two-step quantization strategy, the performance is restored, reaching 80.56%. Finally, by introducing the strategy of loss balance, the performance achieves 80.87%, with the performance drop compared to the float performance being only 0.48 PPs. Experimental results on detection in Table 6 also demonstrate the effect of our method.

4.3.2 The numbers of steps and samples in the reconstruction process

In AOCQ, it is necessary to sample images to adjust the quantization hyperparameter α and the quantization weight $\boldsymbol{W}$. During the second stage, it can be observed that the more iterations are performed, the better the accuracy achieves. From Table 7, it is evident that when the number of steps $\geq$ 1000, the improvement in performance diminishes. Similarly, Fig. 8 indicates that when the number of steps surpasses 1000, the loss approaches convergence. Therefore, in our experiments, it is recommended to set the number of iterations to 1000.

Besides, the number of samples has an effect

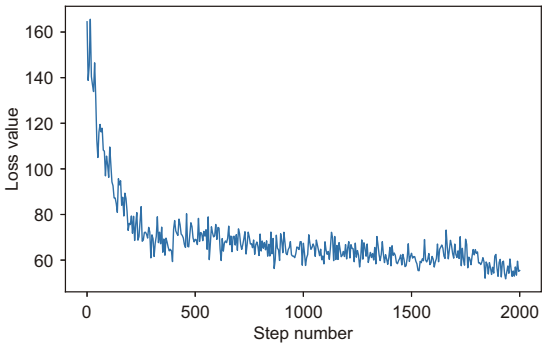

Fig. 8 The loss of different numbers of steps. The loss decreases as the step number increases

Table 3 Comparison of the performance of the proposed PTQ method AOCQ with other SOTA methods for image classification on CIFAR-10

Model	Method	W/A	Model size (M)	Top-1 accuracy (%)
Swin-Tiny		32/32	112	97.20
	BasePTQ*	4/8	28	95.21
	AOCQ (ours)	4/8	28	97.05

W/A means weight/activation

Table 4 Comparison of the performance of the proposed PTQ method AOCQ with other methods for object detection on COCO

Model	Method	W/A	Model size (M)	mAP (%)
Swin-Tiny		32/32	172	45.5
	BasePTQ*	8/8	43	44.9
	AOCQ (ours)	8/8	43	45.1
	BasePTQ*	4/8	22	38.8
	AOCQ (ours)	4/8	22	44.8

The input resolution is set to 1333 × 800. W/A means weight/activation. * means that the quantization is only used for matrix multiplication, where add and other operations maintain float. mAP represents the mean average precision

Table 5 The effect of different components of AOCQ on ImageNet

Model	Weight	Activation	Operation balance	Framework balance	Loss balance	Top-1 accuracy (%)
Swin-Tiny	32	32				81.35
	4	8				66.41
	4	8	✓			78.93
	4	8	✓	✓		80.56
	4	8	✓	✓	✓	80.87

✓ means that the method has been adopted

Table 6 The effect of different components of AOCQ on COCO

Model	Weight	Activation	Operation balance	Framework balance	Loss balance	mAP (%)
Faster R-CNN Swin-Tiny	32	32				45.5
	4	8				33.8
	4	8	✓			43.1
	4	8	✓	✓		44.2
	4	8	✓	✓	✓	44.8

✓ means that the method has been adopted

Table 7 The effect of the number of steps

Model	Number of steps	Weight	Activation	Top-1 accuracy (%)
Swin-Tiny	Baseline	32	32	81.35
	0	4	8	78.93
	100	4	8	80.34
	200	4	8	80.55
	500	4	8	80.78
	1000	4	8	80.87
	2000	4	8	80.86

The results are evaluated on ImageNet. The number of samples is set to 256

Table 8 The effect of the number of samples

Model	Number of samples	Weight	Activation	Top-1 accuracy (%)
Swin-Tiny	Baseline	32	32	81.35
	128	4	8	80.87
	256	4	8	80.86
	512	4	8	80.84
	1024	4	8	80.87

The results are evaluated on ImageNet. The number of steps is set to 1000

on the results. Table 8 shows that 128 samples are enough to achieve the competitive results.

4.3.3 Loss in the reconstruction process

In this subsection, various methods have been explored to balance the tokens in the loss function. From Table 9, it can be observed that token normalization yields the best performance among these methods. The common L_2 norm ($\alpha = 2$) tends to amplify the imbalance between tokens, highlighting the significance of outliers. When using the L_1 norm ($\alpha = 1$), the impact of outliers is diminished, which consequently improves the performance. However, when $\alpha < 1$, the loss does not converge. The result illustrates the effectiveness of balancing tokens in loss optimization.

Table 9 The effect of loss balance

Model	Loss	W/A	Top-1 accuracy (%)
Swin		32/32	81.35
	L_2 ($\alpha = 2$)	4/8	80.56
	L_1 ($\alpha = 1$)	4/8	80.62
	$\alpha = 0.5$	4/8	NaN
	Token norm	4/8	80.87

W/A means weight/activation. NaN stands for "not a number," which appears during the training and means that training cannot continue. $||\cdot||^{\alpha}$ is the default loss function, where $\alpha = 2$ is known as L_2. In contrast, the token norm is adopted in our approach

4.4 Visualization

Gradient-weighted class activation mapping (Grad-CAM) is a highly effective technology for offering visual clarifications of model decisions, which makes the model more transparent and easier to interpret. To give a clear illustration of the AOCQ quantization method's impact, we visualized the activation maps from both the full-precision model and the quantized model with Grad-CAM. As can be seen in Fig. 9, the target in the image shows basically the same heat maps, but there may be tiny differences in the background, like the upper left part of the leopard image in Fig. 9b. This indicates that our method using channel and token has a minimal effect on the target.

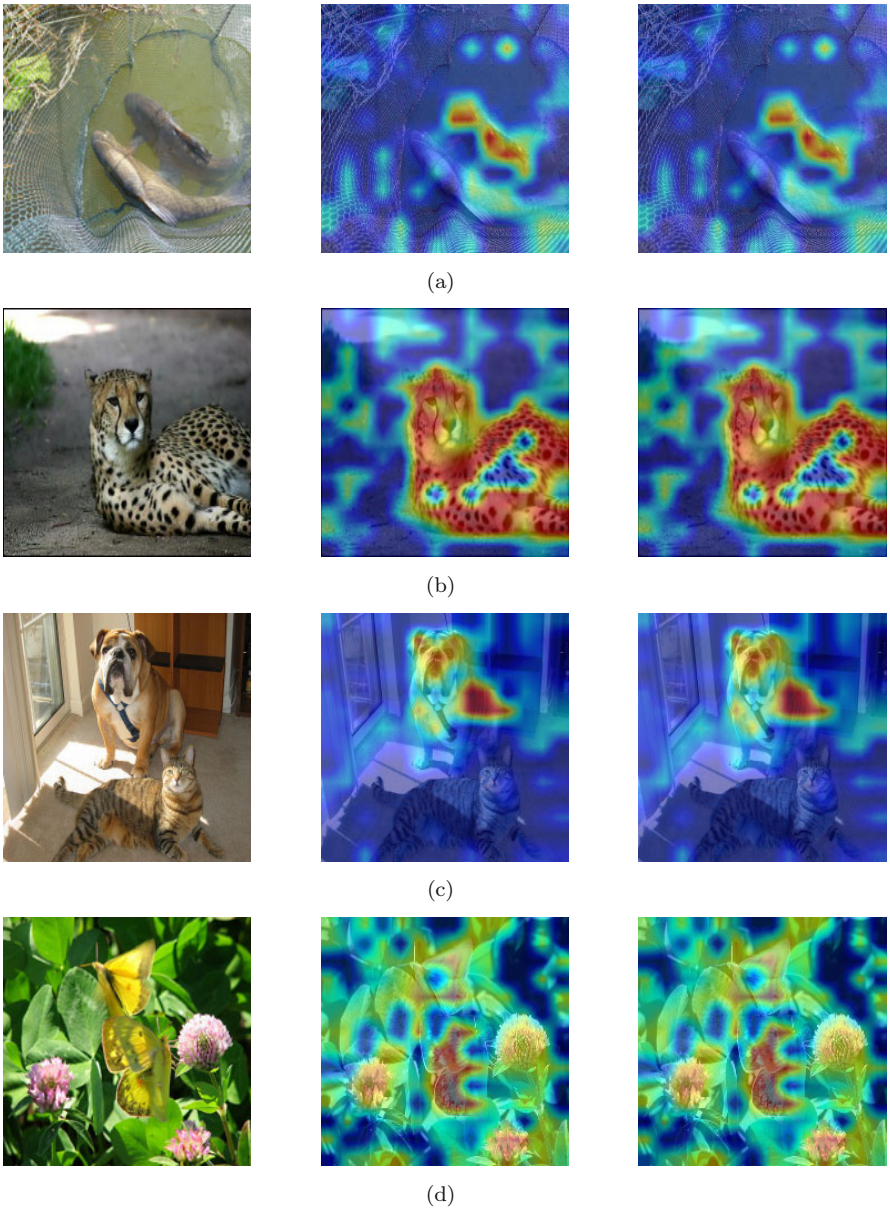

Fig. 9 Gradient-weighted class activation maps: (a) fish; (b) leopard; (c) dog; (d) butterfly. We use Grad-CAM to visualize the attention maps of the ViT-Base model. The left is the original input image. The middle is the Grad-CAM on FP32 model and the right is the Grad-CAM on W4A8 quantized model with our method. All images are resized to 224 × 224 resolutions, and the output of the last layer is used for visualization

5 Conclusions

In this paper, we analyze the issue of outliers in Transformers and theoretically demonstrate that this phenomenon is caused by the structural design of the Transformer itself. To address this problem, a new method (i.e., AOCQ) has been proposed, which balances the impact of outliers from the operator level, framework level, and loss level. Based on the analysis of the condition number of the FIMFisher information matrix, we theoretically prove the effectiveness of our method. Experimentally, various tasks, including classification and object detection, have been validated. Specifically, AOCQ achieves a decrease of 0.7 PPs in the performance on Swin Transformer and DeiT models with 4× compression. We hope that this solution will facilitate the deployment of Transformers in the industry.

In the future, the application of large language models (LLMs) based on the Transformer architecture will become increasingly widespread. The number of channels and the depth of blocks in LLMs are greater compared to those of ViTs, which means that the impact of outliers will be even more significant. This implies that the challenge of quantizing LLMs will be greater. We will explore the role of our approach within LLMs in the future, with the hope of accelerating the development of LLMs.

Contributors

Zheyang LI carried out the molecular genetic studies, participated in the sequence alignment, and drafted the paper. Chaoxiang LAN and Kai ZHANG participated in the design of the study and performed the statistical analysis. Jun XIAO, Ye REN, and Wenming TAN conceived the study, participated in the design and coordination, and helped organize the paper. Zheyang LI and Jun XIAO revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the study are publicly available. The ImageNet-1K dataset can be accessed at http://www.image-net.org/. The COCO 2017 dataset can be accessed at https://cocodataset.org/.

References

Alam N, Kolawole S, Sethi S, et al., 2023. Vision Transformers for mobile applications: a short survey. https://arxiv.org/abs/2305.19365

Ba J, Grosse R, Martens J, 2017. Distributed second-order optimization using Kronecker-factored approximations. Proc 5th Int Conf on Learning Representations, p.1-17.

Ba JL, Kiros JR, Hinton GE, 2016. Layer normalization. https://arxiv.org/abs/1607.06450

Bengio Y, Leonard N, Courville A, 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. https://arxiv.org/abs/1308.3432

Carion N, Massa F, Synnaeve G, et al., 2020. End-to-end object detection with Transformers. Proc 16th European Conf on Computer Vision, p.213-229. https://doi.org/10.1007/978-3-030-58452-8_13

Chen MH, Peng HW, Fu JL, et al., 2021. AutoFormer: searching Transformers for visual recognition. Proc IEEE/CVF Int Conf on Computer Vision, p.12250-12260. https://doi.org/10.1109/ICCV48922.2021.01205

Chen ZS, Xie LX, Niu JW, et al., 2021. Visformer: the vision-friendly Transformer. Proc IEEE/CVF Int Conf on Computer Vision, p.569-578. https://doi.org/10.1109/ICCV48922.2021.00063

Chitty-Venkata KT, Mittal S, Emani M, et al., 2023. A survey of techniques for optimizing Transformer inference. *J Syst Archit*, 144:102990. https://doi.org/10.1016/j.sysarc.2023.102990

Choi J, Wang Z, Venkataramani S, et al., 2018. PACT: parameterized clipping activation for quantized neural networks. https://arxiv.org/abs/1805.06085

Choromanski KM, Likhosherstov V, Dohan D, et al., 2021. Rethinking attention with performers. Proc 9th Int Conf on Learning Representations, p.1-38.

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional Transformers for language understanding. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.4171-4186. https://doi.org/10.18653/v1/N19-1423

Ding MY, Xiao B, Codella N, et al., 2022. DaViT: dual attention vision Transformers. Proc 17th European Conf on Computer Vision, p.74-92. https://doi.org/10.1007/978-3-031-20053-3_5

Dong XY, Bao JM, Chen DD, et al., 2022. CSWin Transformer: a general vision Transformer backbone with cross-shaped windows. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12114-12124. https://doi.org/10.1109/CVPR52688.2022.01181

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16×16 words: Transformers for image recognition at scale. Proc 9th Int Conf on Learning Representations, p.1-21. https://doi.org/10.48550/arXiv.2010.11929

Esser SK, McKinstry JL, Bablani D, et al., 2020. Learned step size quantization. Proc 8th Int Conf on Learning Representations, p.1-12.

Gong RH, Liu XL, Jiang SH, et al., 2019. Differentiable soft quantization: bridging full-precision and low-bit neural networks. Proc IEEE/CVF Int Conf on Computer Vision, p.4851-4860. https://doi.org/10.1109/ICCV.2019.00495

Graham B, El-Nouby A, Touvron H, et al., 2021. LeViT: a vision Transformer in ConvNet's clothing for faster inference. Proc IEEE/CVF Int Conf on Computer Vision, p.12239-12249. https://doi.org/10.1109/ICCV48922.2021.01204

Grosse RB, Martens J, 2016. A Kronecker-factored approximate Fisher matrix for convolution layers. Proc 33rd Int Conf on Machine Learning, p.573-582.

Ham TJ, Jung SJ, Kim S, et al., 2020. A^3: accelerating attention mechanisms in neural networks with approximation. Proc IEEE Int Symp on High Performance Computer Architecture, p.328-341. https://doi.org/10.1109/HPCA47549.2020.00035

Han DC, Pan XR, Han YZ, et al., 2023. FLatten Transformer: vision Transformer using focused linear attention. Proc IEEE/CVF Int Conf on Computer Vision, p.5938-5948. https://doi.org/10.1109/ICCV51070.2023.00548

Han S, Pool J, Tran J, et al., 2015. Learning both weights and connections for efficient neural networks. Proc 29th Int Conf on Neural Information Processing Systems, p.1135-1143.

Hatamizadeh A, Heinrich G, Yin HX, et al., 2024. FasterViT: fast vision Transformers with hierarchical attention. Proc 12th Int Conf on Learning Representations, p.1-24.

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. *Comput Sci*, 14(7):38-39.

Hong K, Dai GH, Xu JM, et al., 2023. FlashDecoding++: faster large language model inference on GPUs. https://arxiv.org/abs/2311.01282

Huang L, Qin J, Liu L, et al., 2020. Layer-wise conditioning analysis in exploring the learning dynamics of DNNs. Proc 16th European Conf on Computer Vision, p.384-401. https://doi.org/10.1007/978-3-030-58536-5_23

LeCun Y, Kanter I, Sona SA, 1990. Second order properties of error surfaces: learning time and generalization. Proc 4th Int Conf on Neural Information Processing Systems, p.918-924.

LeCun Y, Bottou L, Orr GB, et al., 2012. Efficient BackProp. In: Montavon G, Orr GB, Miller KB (Eds.), Neural Networks: Tricks of the Trade. Springer, Berlin, Heidelberg, p.9-48. https://doi.org/10.1007/978-3-642-35289-8_3

Li F, Zhang H, Liu SL, et al., 2022. DN-DETR: accelerate DETR training by introducing query denoising. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.13609-13617. https://doi.org/10.1109/CVPR52688.2022.01325

Li F, Zhang H, Sun P, et al., 2025. Segment and recognize anything at any granularity. Proc 18th European Conf on Computer Vision, p.467-484. https://doi.org/10.1007/978-3-031-73195-2_27

Li YH, Gong RH, Tan X, et al., 2021. BRECQ: pushing the limit of post-training quantization by block reconstruction. Proc 9th Int Conf on Learning Representations, p.1-16.

Li ZK, Ma LP, Chen MJ, et al., 2022. Patch similarity aware data-free quantization for vision Transformers. Proc 17th European Conf on Computer Vision, p.154-170. https://doi.org/10.1007/978-3-031-20083-0_10

Li ZK, Xiao JR, Yang LW, et al., 2023. RepQ-ViT: scale reparameterization for post-training quantization of vision transformers. Proc IEEE/CVF Int Conf on Computer Vision, p.17181-17190. https://doi.org/10.1109/ICCV51070.2023.01580

Li ZK, Chen MJ, Xiao JR, et al., 2024. PSAQ-ViT V2: toward accurate and general data-free quantization for vision transformers. *IEEE Trans Neur Netw Learn Syst*, 35(12):17227-17238. https://doi.org/10.1109/TNNLS.2023.3301007

Liang JY, Cao JZ, Sun GL, et al., 2021. SwinIR: image restoration using Swin Transformer. Proc IEEE/CVF Int Conf on Computer Vision Workshops, p.1833-1844. https://doi.org/10.1109/ICCVW54120.2021.00210

Lin Y, Zhang TY, Sun PQ, et al., 2022. FQ-ViT: post-training quantization for fully quantized vision Transformer. Proc 31st Int Joint Conf on Artificial Intelligence, p.1173-1179. https://doi.org/10.24963/ijcai.2022/164

Liu SL, Li F, Zhang H, et al., 2022. DAB-DETR: dynamic anchor boxes are better queries for DETR. Proc 10th Int Conf on Learning Representations, p.1-20.

Liu Z, Lin YT, Cao Y, et al., 2021. Swin Transformer: hierarchical vision Transformer using shifted windows. Proc IEEE/CVF Int Conf on Computer Vision, p.9992-10002. https://doi.org/10.1109/ICCV48922.2021.00986

Liu Z, Ning J, Cao Y, et al., 2022a. Video Swin Transformer. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3192-3201. https://doi.org/10.1109/CVPR52688.2022.00320

Liu Z, Hu H, Lin YT, et al., 2022b. Swin Transformer V2: scaling up capacity and resolution. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11999-12009. https://doi.org/10.1109/CVPR52688.2022.01170

Liu ZH, Wang YH, Han K, et al., 2021. Post-training quantization for vision Transformer. Proc 35th Int Conf on Neural Information Processing Systems, Article 2152.

Mehta S, Ghazvininejad M, Iyer S, et al., 2021. DeLighT: deep and light-weight Transformer. Proc 9th Int Conf on Learning Representations, p.1-19.

Nagel M, Amjad RA, van Baalen M, et al., 2020. Up or down? Adaptive rounding for post-training quantization. Proc 37th Int Conf on Machine Learning, Article 667.

Qu Z, Liu L, Tu FB, et al., 2022. DOTA: detect and omit weak attentions for scalable Transformer acceleration. Proc 27th ACM Int Conf on Architectural Support for Programming Languages and Operating Systems, p.14-26. https://doi.org/10.1145/3503222.3507738

Ren SQ, He KM, Girshick R, et al., 2015. Faster R-CNN: towards real-time object detection with region proposal networks. Proc 29th Int Conf on Neural Information Processing Systems, p.91-99.

Sanh V, Debut L, Chaumond J, et al., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. https://arxiv.org/abs/1910.01108

Shen S, Yao ZW, Gholami A, et al., 2020. PowerNorm: rethinking batch normalization in Transformers. Proc 37th Int Conf on Machine Learning, Article 811.

Si CY, Yu WH, Zhou P, et al., 2022. Inception Transformer. Proc 36th Int Conf on Neural Information Processing Systems, Article 1707.

Touvron H, Cord M, Sablayrolles A, et al., 2021a. Going deeper with image Transformers. Proc IEEE/CVF Int Conf on Computer Vision, p.32-42. https://doi.org/10.1109/ICCV48922.2021.00010

Touvron H, Cord M, Douze M, et al., 2021b. Training data-efficient image Transformers & distillation through attention. Proc 38th Int Conf on Machine Learning, p.10347-10357.

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Yang QM, Zhang K, Lan CX, et al., 2022. Unified normalization for accelerating and stabilizing Transformers. Proc 30th ACM Int Conf on Multimedia, p.4445-4455. https://doi.org/10.1145/3503161.3547860

Yao ZW, Aminabadi RY, Zhang MJ, et al., 2022. ZeroQuant: efficient and affordable post-training quantization for large-scale Transformers. Proc 36th Int Conf on Neural Information Processing Systems, Article 1970.

Yu XD, Shi DH, Wei X, et al., 2022. SOIT: segmenting objects with instance-aware Transformers. Proc 36th AAAI Conf on Artificial Intelligence, p.3188-3196. https://doi.org/10.1609/aaai.v36i3.20227

Yuan L, Chen YP, Wang T, et al., 2021. Tokens-to-Token ViT: training vision Transformers from scratch on ImageNet. Proc IEEE/CVF Int Conf on Computer Vision, p.538-547. https://doi.org/10.1109/ICCV48922.2021.00060

Yuan ZH, Xue CH, Chen YQ, et al., 2022. PTQ4ViT: post-training quantization framework for vision Transformers. https://arxiv.org/abs/2111.12293

Zafrir O, Boudoukh G, Izsak P, et al., 2019. Q8BERT: quantized 8bit BERT. Proc 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition, p.36-39. https://doi.org/10.1109/EMC2-NIPS53020.2019.00016

Zhang H, Li F, Liu SL, et al., 2023. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. Proc 11th Int Conf on Learning Representations, p.1-19.

Zhang XS, Tian YJ, Xie LX, et al., 2023. HiViT: a simpler and more efficient design of hierarchical vision Transformer. Proc 11th Int Conf on Learning Representations, p.1-15.

Zheng CY, Li ZY, Zhang K, et al., 2022. SAViT: structure-aware vision Transformer pruning via collaborative optimization. Proc 36th Annual Conf on Neural Information Processing Systems, Article 655.

Zhu XZ, Su WJ, Lu LW, et al., 2021. Deformable DETR: deformable Transformers for end-to-end object detection. Proc 9th Int Conf on Learning Representations, p.1-16.