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 310027, China*

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

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

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 CNNs, 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-B with 8-bit post-training quantization (PTQ) can achieve 81.57% with only 0.3% 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 an ultra-low 4 bits, with a minimal accuracy loss of 2%, while requiring nearly 8× less memory.

Unedited

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

1 Introduction

Transformer-based architectures(Vaswani A, 2017) have shown great power in Natural Language Processing (NLP) tasks(Devlin J, 2019; Mehta S, 2021). 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 N, 2020; Dosovitskiy A, 2020; Chen ZS, 2021; Graham B, 2021; Liu Z, 2021a; Touvron H, 2021b; Yuan L, 2021; Dong XY, 2022; Yu XD, 2022). transformers consist of a number of blocks containing multi-head self attention (MHSA) and feed-forward networks (FFN), which enable 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 (Chitty-Venkata KT, 2023; Alam N, 2023). Consequently, there is an urgent industry requirement to compress and accelerate these transformer-based models to facilitate broader application.

Much efforts have been invested in facili-

‡ Corresponding author
ORCID: Zheyang Li, https://orcid.org/0000-0002-0229-8707
© Zhejiang University Press 2025

tating the deployment of transformers, including pruning(Han S, 2015; Zheng CY, 2022), distillation(Touvron H, 2021a), quantization(Yao ZW, 2022), and the direct design of more efficient transformers(Choromanski KM, 2022; Yang QM, 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 KT, 2023). Unlike the Quantization-Aware Training method, which necessitates the entire training dataset, Post-Training Quantization only requires unlabeled calibration images, thereby enabling rapid quantization and deployment. Consequently, our focus is on Post-Training Quantization methods to compress transformers. Besides, quantization and other techniques (Han S, 2015; Touvron H, 2021a; Yang QM, 2022) are complementary for model acceleration.

Previous studies such as Ranking-Aware(Liu ZH, 2021) adopts a ranking loss to make the order of the self-attention results after quantization as consistent as possible. FQ-ViT(Lin Y, 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. RepQ-ViT(Li ZK, 2023) uses channel-wise quantization to deal with the imbalance between channels and utilizes 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 log2 quantizer with minimal computations. However, we find that not only the channel variation but also the token variation are both keys to limiting the accuracy of the quantized transformers.

To mitigate the adverse effects caused by these issues, we propose an adaptive outlier correction quantization method(AOCQ) 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 perchannel public factor to deal with the activations before the add operator to balance the large channel discrepancy. Next, based on the brecq(Li YH, 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 makes 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 taht 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 Post-training quantization methods.

The remainder of the paper is organized as follows. Section 2 introduces the fundamental concept of transformers and their main acceleration methods. Section 3 provides the core problem of transformers and the overview of the proposed method, including the algorithm framework and the theoretical proof. Section 4 presents the detailed experimental setup and presents the results and analysis. Section 5 concludes the paper and discusses the directions for future research.

2 Related work

2.1 Vision Transformer

Transformers initially show great success in sequence modeling and neural machine translation(Vaswani A, 2017). Owing to the high flexibility, transformers have been applied to multi-modal and speech tasks. Carion et al.(Carion N, 2020) trained an end-to-end Transformer based detector DETR, achieving results comparable to CNNs. Meanwhile, (Dosovitskiy A, 2020) proposed a pure transformer (ViTs) to treat image patches as the sequences for image classification. Besides, transformer has been

Unedited

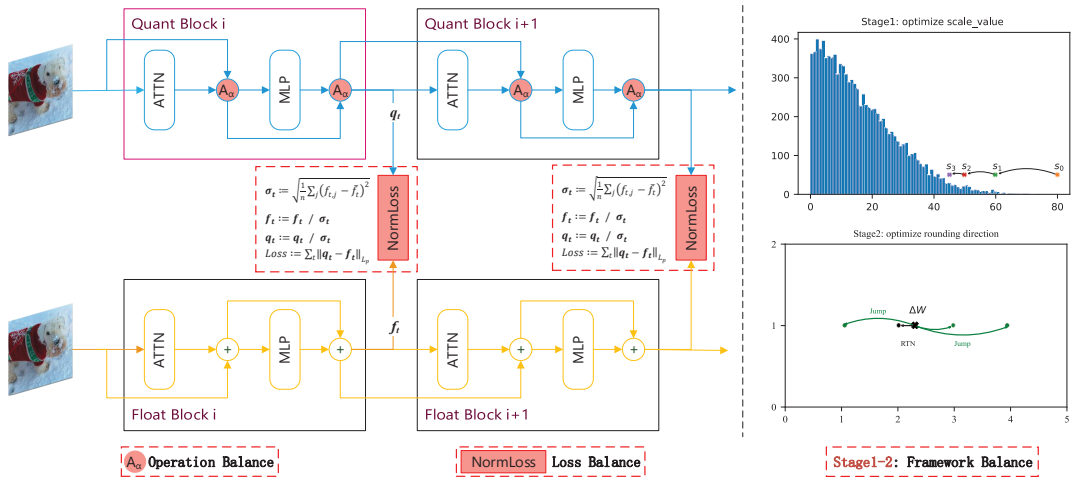


Fig. 1 Framework of Adaptive Outlier Correction Quantization (AOCQ). AOCQ deals with the outliers adaptively in three levels: (a) operation balance. It uses Add_α and LN_α to diminish the imbalance between channels; (b) framework balance. A two-stage optimization method has been used to optimize quantization 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. Ater that, the rounding direction of the weights is optimized in the second stage;(c) loss balance. The token norm function suppresses the extreme tokens in the block-wise reconstruction loss

Unedited

applied to a range of computer vision problems, including object detection (Zhu XZ, 2020; Zhang H, 2023; Li F, 2022; Liu SL, 2022), semantic segmentation(Li F, 2023), image processing(Liang JY, 2021), and video understanding(Liu Z, 2021b). In the following years, Vision Transformer models sprang up like mushrooms in the following years. T2T (Yuan L, 2021) utilizes a token-to-token Transformer to improve the local relations between adjacent tokens before entering the backbone network. A family of shifted window-based transformers, including Swin(Liu Z, 2021a) and Swin V2(Liu Z, 2021c), uses local shifted window-based attention to construct a hierarchical architecture and achieve strong baselines in many vision tasks. (Ding MY, 2022) proposed a new framework, DaViT, which combines mixed blocks of channel attention and spatial attention. Inception Transformer (Si CY, 2022) captures both the high and low frequencies in inputs by applying the Inception mixer of parallel convolution modules and self-attention modules. HiVit(Zhang XS, 2023) came up with a plain and efficient Transformer backbone and showed a clear advantage on several benchmarks. Recently, FasterViT (Hatamizadeh A, 2023) and FLatten Transformer (Han DC, 2023) pay efforts to make the attention more efficient by the hierarchical attention and the depthwise attention. In conclusion, deploying various transformers on resource-limited devices still has been urgent in industry.

2.2 Quantization-Aware Training and Post-Training Quantization

Model compression is an effective strategy for reducing the development costs of deep neural networks (DNNs). These approaches contain pruning (Han S, 2015; Zheng CY, 2022), distillation(Hinton G, 2015; Sanh V, 2019), quantization(Yao ZW, 2022), efficient architecture design(Choromanski KM, 2022; Chen MH), Transformer function and kernel optimization (Hong K, 2023), and hardware acceleration optimization(Qu Z, 2022; Ham TJ, 2020). Quantization is an effective and promising way to accelerate the neural networks. There are two types of quantization methods, Quantization Aware training methods (Zafrir O, 2019; Choi J, 2018) and Post-training quantization methods (Yuan ZH, 2022).

Quantization-aware training (QAT) improves the accuracy of the quantized model by finetuning on the training datasets with straight-through estimator (STE) (Bengio Y, 2013), which is used to approximate the gradient. Similarly, (Gong RH, 2019) introduces a differentiable tanh function to simulate the gradients in the training process. Learned Step Size

Quantization (LSQ) (Esser SK, 2020) pays attention to optimizing the quantization intervals, achieving good performance on several benchmarks.

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

2.3 Post-Training Quantization for ViTs

Several quantization methods (Li YH, 2021; Liu ZH, 2021; Li ZK, 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, 2021) firstly proposed a PTQ method to quantize the vision transformer, where a ranking loss has been proposed to keep the relative order of the self-attention results after quantization. Besides, the models are not fully quantized, where some parts need to retain floating-point units in the hardware, leading to hardware inefficiency. FQ-ViT (Lin Y, 2022) proposes Power-of-Two Factor (PTF) and Log-Int-Softmax (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, 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, 2023) adopts 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 Transformer

A standard transformer consists of a series of blocks which contained multi-head self-attention (MHSA) and feedforward networks (FFN). The attention can be formulated as:

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

where $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ represents query, key and value matrix, respectively. d is the hidden size of $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$. MHSA contains multiple heads, each of which concurrently computes attention operations. All heads are concatenated depthwise and linearly transformed to the output by an FC layer. The MHSA can be formulated as:

$$\begin{aligned}\mathrm{MHSA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) &= [head_1, head_2, ..., head_{\mathrm{h}}] * W^{\mathrm{O}},\\ head_i &= \mathrm{Attention}(QW_i^{\boldsymbol{Q}}, KW_i^{\boldsymbol{K}}, VW_i^{\boldsymbol{V}}),\end{aligned} \quad (2)$$

where $W_{\mathrm{Q}}, W_{\mathrm{K}}, W_{\mathrm{V}}, W_{\mathrm{O}}$ are the corresponding weight parameters. $head_i$ represents the i-th head.

Following MHSA, FFN is constructed by stacking two fully-connected (FC) layers with activation functions, such as ReLU or GeLU. The FFN can be formulated as:

$$\mathrm{FFN}(x) = \phi(xW_1 + b_1)W_2 + b_2, \quad (3)$$

where x is input feature and ϕ stands for the activation function. As a result, the whole block consists of alternative MHSA and FFN. Besides, LayerNorm (Ba JL, 2016) is adopted to normalize activations, which helps avoid extreme outliers.

The whole transformer block can be written as:

$$\begin{aligned}z &= \mathrm{MHSA}(\mathrm{LN}(x)) + x,\\ y &= \mathrm{FFN}(\mathrm{LN}(z)) + z,\end{aligned} \quad (4)$$

where LN is the LayerNorm.

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 also been mentioned in other related works (Shen S, 2020). Lemma 3.1 theoretically supports this phenomenon. The increased activation values and their variances bring great challenge to quantize transformers. The direct

Unedited

Table 1 Fully quantization results for Swin transformers without any tricks. All the operations including FC, softmax, matmul, add, Layernorm and GeLU have been quantized. The outliers in deeper models have worse impact on the performance

Model	Blocks	Top-1 with FP32	Top-1 with W8A8	δ
Swin-Tiny	12	81.35	80.73	-0.62
Swin-Small	24	83.30	72.33	-10.87
Swin-Base	36	83.60	46.13	-37.47

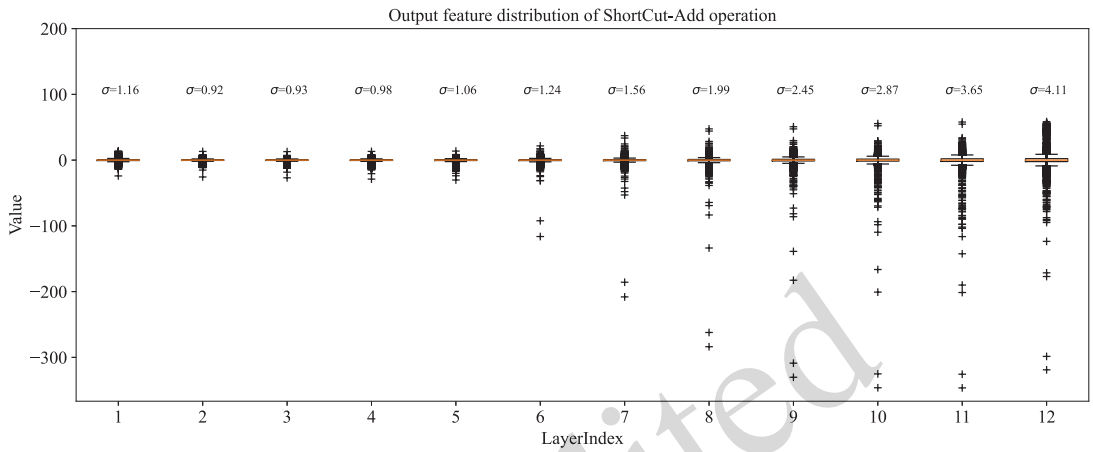


Fig. 2 Boxplot of different layers' activations. As the network deepens, the average and maximum activation values also become larger and larger

Unedited

quantization results of different depths' transformers are shown in Table 1.The more Transformer block model contains, the large the accuracy drop.

Lemma 1 Let l be the depth of the model, d_l stands for the channel dimension, W_{Q} and W_{K} be initialized independently, and $W_{\mathrm{V}} \sim N(0,1)$. Moreover,

For the Post-LN Transformer, we have:

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

$E(||X_{(l,i)}^{post}||_2^2)$ denotes the Expectation of the output features $X_{(l,i)}^{post}$. For the Pre-LN Transformer, we have:

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

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 :

$$\begin{aligned} x_1 &= \mathrm{LayerNorm}(x_0), \\ x_2 &= \mathrm{MHSA}(x_1, bias), \\ x_3 &= x_0 + x_2, \\ x_4 &= \mathrm{LayerNorm}(x_3), \\ x_5 &= \mathrm{FFN}(x_4), \\ x_6 &= x_3 + x_5, \end{aligned} \tag{7}$$

Let $x_{\mathrm{i}} = (t_1, t_2, \ldots, t_n)$, n be the token number and t_i be a token, we have:

$$\begin{aligned} ||x_1||_2^2 &= \sum_{i=1}^{d}(t_{1,i} - \mu_1)^2/(\sigma_1^2), \\ &= \frac{d}{(\sigma_1^2)}\sigma_1^2 = d, \end{aligned} \tag{8}$$

where μ_1 and σ_1^2 are the mean and std of x_1, which is calculated by Layernorm. d means the channel dimension. Similarly,

$$E(||x_4||_2^2) = d. \tag{9}$$

As W_{Q} and W_{K} are initialized independently, the value of softmax input feature is similar and the value

of softmax output is equal to $1/n$. We have

$$\begin{aligned} x_2 &= \mathrm{MHSA}(x_1, bias), \\ &= \frac{1}{n}(1,1,...,1)x_1W^V, \\ &= \frac{1}{n}\sum t_iW^V. \end{aligned} \tag{10}$$

At the same time, when $W \sim N(0,1)$, it can be arrived at:

$$\begin{aligned} E(||tW||_2^2) &= E(\sum_i^d (tW_i)^2), \\ &= E(\sum_i \sum_j tW_iW_j^Tt^T), \\ &= E(\sum_i tW_iW_i^Tt^T, \\ &+ \sum_{i\neq j}\sum_j tW_iW_j^Tt^T), \\ &= \sum_i E(tW_iW_i^Tt^T), \\ &= \sum_i tE(W_iW_i^T)t^T, \\ &= \sum_i^d \frac{1}{d}tt^T, \\ &= ||t||_2^2. \end{aligned} \tag{11}$$

According to Eq 11, it can be calculated as:

$$\begin{aligned} E(||x_2||_2^2) &= E(||(\frac{1}{n}\sum t_i)W^V||_2^2), \\ &= E(||\frac{1}{n}\sum t_i||_2^2), \\ &\leq \frac{1}{n}\sum E(||t_i||_2^2), \\ &= \frac{1}{n}\sum E(||t_i||_2^2) = d. \end{aligned} \tag{12}$$

Because we assume the token is independent and the parameters' expectation is 0. We have

$$\begin{aligned} E(x_2x_0^T) &= \frac{1}{n}E(\sum_{i=1}^n t_iW^Vx_0^T), \\ &= \frac{1}{n}\sum_{i=1}^n E(t_iW^Vx_0^T), \\ &= \frac{1}{n}\sum_{i=1}^n E(E(t_iW^Vx_0^T)|x_0^T), \\ &= 0. \end{aligned} \tag{13}$$

Therefore,

$$\begin{aligned} E(||x_3||_2^2) &= E(||x_0 + x_2||_2^2), \\ &= E(||x_0||_2^2) + E(||x_2||_2^2) + 2E(x_2x_0^T), \\ &\leq E(||x_0||_2^2) + d. \end{aligned} \tag{14}$$

That means,

$$E(||x_0||_2^2) \leq E(||x_3||_2^2) \leq E(||x_0||_2^2) + d. \tag{15}$$

We can get the following equation from the same process:

$$E(||x_6||_2^2) = E(||x_3||_2^2) + \frac{d}{2}. \tag{16}$$

Combining the above formulations, we get:

$$E(||x_0||_2^2) + \frac{d}{2} \leq E(||x_6||_2^2) \leq E(||x_0||_2^2) + \frac{3}{2}d. \tag{17}$$

□

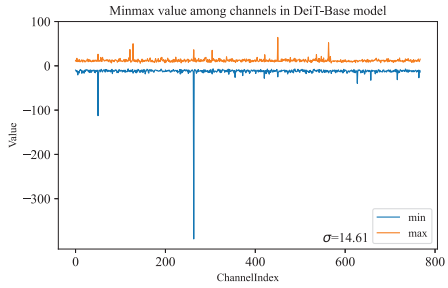
Furthermore, there are serious inter-channel and inter-token variation in vision transformers, which brings unacceptable quantization errors, as shown in Fig. 3. The large range of inter-channel and inter-token variation may make transformer extract more powerful representation. However, preserving the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

3.2 Operation balance

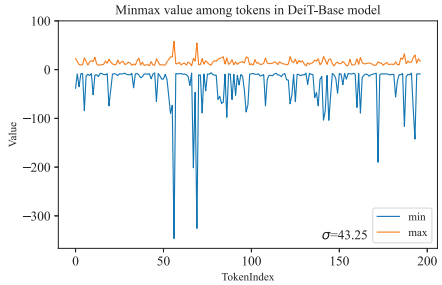
In this part, we introduce two new operators named Add_α and LN_α, where the add operation and LN operation are equipped with a per-channel factor α, respectively. The factors are used to deal with the inter-channel variation directly, which are extracted from the channel-imbalanced features. Applying 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.

In Fig. 5, we plot the density distributions of activations. This visualization clearly shows that the original distribution exhibits dual-peaks 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.

Unedited



(a)



(b)

Fig. 3 The outliers in transformers. These demonstrate that the outliers exist in different channels and tokens: (a)Inter-channel variation in transformers; (b) Inter-token variation in transformers

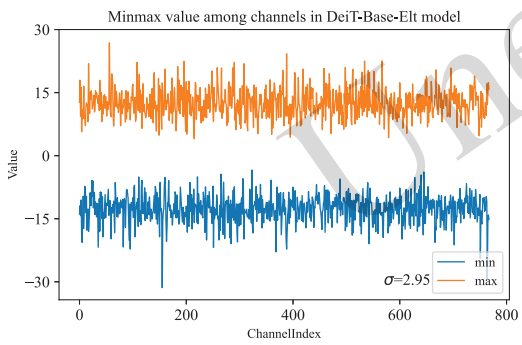


Fig. 4 The activations with Add_α. The comparison of Fig 3(a) demonstrates the outliers in different channels have been diminished largely with Add_α, where std is reduced from 14.61 to 2.95

Concretely, Add_α and LN_α can be written as:

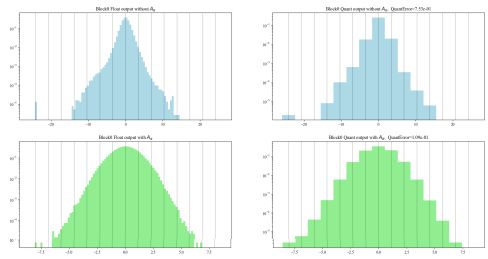
$$\text{Add}_\alpha(x1, x2) = a * x_1 + b * x_2 + c, \tag{18}$$

$$\text{LN}_\alpha(x) = \text{LN}(e * x). \tag{19}$$

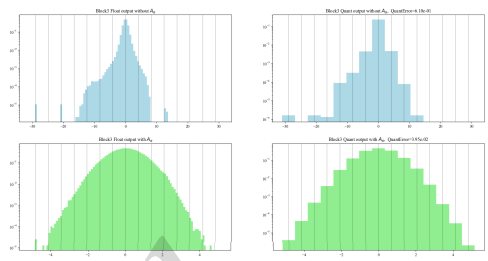
where a, b, c, e are calculated according to the calibration set.

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

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



(a)



(b)

Fig. 5 Comparison of density distribution. The x-axis represents value intervals, while the y-axis represents the count of activations falling within each interval. The left and the right represent the original and quantized distribution. The top and the bottom denote the BasePTQ and our method:(a)Density distribution on block 0;(b)Density distribution on block 0

In conclusion , it is mathematically equivalent to Eq. (4), as shown in Proof 3.2.

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

$$y = x_1 + x_2, \tag{21}$$

x_1 and x_2 stand for the input, while y is the output. Equipped with Add_α, the new calculating process is:

$$x_1^\alpha = \frac{x_1 - c_1}{a_1}, \tag{22}$$

$$x_2^\alpha = \frac{x_2 - c_2}{a_2}, \tag{23}$$

$$\begin{aligned} y &= \text{Add}_\alpha(x_1^\alpha, x_2^\alpha), \\ &= a * x_1^\alpha + b * x_2^\alpha + c, \\ &= a * \frac{x_1 - c_1}{a_1} + b * \frac{x_2 - c_2}{a_2} + c. \end{aligned} \tag{24}$$

Note x_1^α and x_2^α are more smoothly than x_1 and x_2 at the channel level. We quantize x_1^α and x_2^α, instead of x_1 and x_2. As a result, x_1^α and x_2^α leads to lower quantization error. a_1, a_2, c are the channel-level common factors. Let $a = a_1$, $b = a_2$ and $c = c_1 + c_2$,

Unedited

it can be transformed as:

$$\begin{aligned} y &= a_1 * \frac{x_1 - c_1}{a_1} + a_2 * \frac{x_2 - c_2}{a_2} + c_1 + c_2, \\ &= x_1 + x_2. \end{aligned} \tag{25}$$

It proved that the result in Eq(6) is equal to Eq(1). Similarly, the LN in transformer can be written as:

$$y = \frac{x_1 - E(x_1)}{Var(x_1)}. \tag{26}$$

With LN_α, the new calculating process is:

$$x_1^\alpha = \frac{x_1 - c_1}{a_1}, \tag{27}$$

$$\begin{aligned} y &= \mathrm{LN}_\alpha(x_1^\alpha), \\ &= \mathrm{LN}(a * x_1^\alpha), \\ &= \frac{a * x_1^\alpha - E(a * x_1^\alpha)}{Var(a * x_1^\alpha)}, \\ &= \frac{a * \frac{x_1 - c_1}{a_1} - E(a * \frac{x_1 - c_1}{a_1})}{Var(a * \frac{x_1 - c_1}{a_1})}. \end{aligned} \tag{28}$$

Let $a = a_1$, it can be:

$$\begin{aligned} y &= \frac{a_1 * \frac{x_1 - c_1}{a_1} - E(a_1 * \frac{x_1 - c_1}{a_1})}{Var(a_1 * \frac{x_1 - c_1}{a_1})}, \\ &= \frac{x_1 - c - E(x_1 - c)}{Var(x_1 - c)}, \\ &= \frac{x_1 - E(x_1)}{Var(x_1)}. \end{aligned} \tag{29}$$

It shows that Eq(9) is equal to Eq(6). Similarly, we quantize the x_1^α, instead of x_1, to achieve a better result.

□

As shown in Fig. 1, Add_α and LN_α serve as drop-in replacements for the original Add and LN operations, 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 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 the 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 the Fig. 1. Compared with Brecq, the two-stage optimization framework can deal with the adverse effects of outliers.

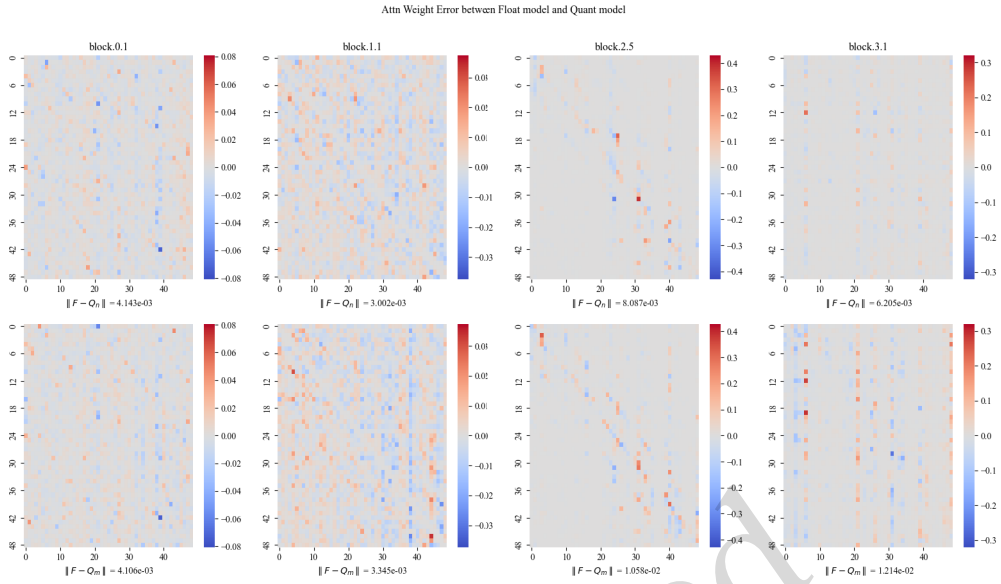
3.4 Loss balance

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 have introduced 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 1.

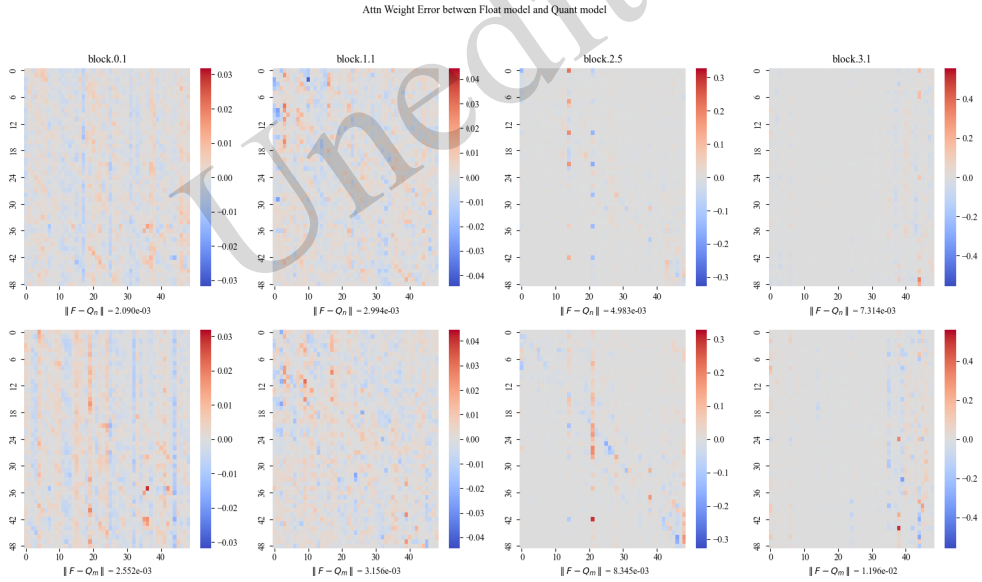
Note that norm-loss is applied only during the quantization reconstruction phase. It is not used during actual inference. Our objective with norm-loss is to consider the importance of all tokens for more balanced quantization parameter learning. Furthermore, we visualized the attention maps derived using 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 Fisher Information Matrix (FIM) to prove the effectiveness of our method theoretically. The condition number represents how easy it is to optimize the corresponding layer(LeCun YA, 2012; Le Cun Y, 1990). However, the global Fisher information matrix is difficult due to the high memory and computational costs. Under mild, the FIM is approximated computed by using the Kronecker product (K-FAC) (Grosse RB, 2016; Ba J, 2017; Huang L, 2020). As a result, the full FIM can be replaced by a block diagonal matrix, $F = diag(F_1, ..., F_k)$. F_k can be computed as:

Unedited



(a)



(b)

Unedited

Fig. 6 Attention map errors. 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, which critically impact 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:(a)Attention map errors on different blocks on ImageNet;(b)Attention map errors on different blocks on CIFAR-10

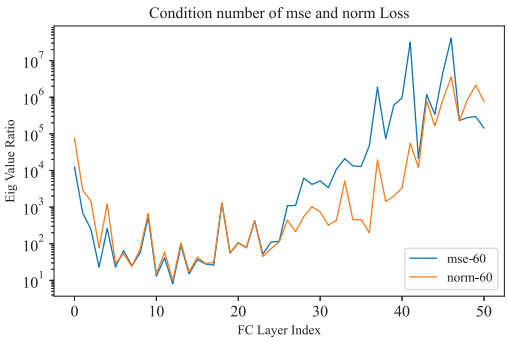


Fig. 7 Condition number of different FC layers in the transformer. The orange and blue represents the condition number w/o norm-loss. In the deep layers of the network (layer 28-40), the condition number of norm-loss significantly decreases compared with regular loss, which means it is more easily optimized

$$\begin{aligned} F_k &= E(vec(W_g)vec(W_g)^T), \\ &= E(vec(gx^T)vec(gx^T)T), \\ &= E((x \times g)(x \times g)^T), \\ &= E((x \times x^T)(g \times g^T)), \\ &\approx E(xx^T) \times E(gg^T). \end{aligned} \tag{30}$$

where x is the $k-th$ input and g means the gradient of x. W_g is the gradient of 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 vision transformers 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.

4.1 Experimental settings

The 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 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. The same as (Lin Y, 2022), softmax and layer normalization layers in vision transformers have been quantized. In block-wise reconstruction process, the 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-RCNN (Ren SQ, 2015).

4.2 Comparison with state-of-the-art methods

The proposed method has been compared with other post-training quantization methods, including PTQ4ViT and BasePTQ(Yuan ZH, 2022), FQ-ViT(Lin Y, 2022).

4.2.1 Image classification

Different vision transformer 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% on ViTs, and with 4-bit quantization, the performance drop is within 2.1%. We also noticed that the larger the model, the smaller the performance drop, indicating that larger models are less sensitive to quantization and their performance is more stable. Comparing Swin and DeiT, we found that DeiT experiences a greater performance drop. The reason may be

Unedited

Table 2 Comparison of the performance of proposed post-training quantization method AOCQ with other SOTA method for image classification on ImageNet. The input resolution is set to 224×224. * means quantization is only used for matrix multiplication, where add and other operations maintain float. BasePTQ is also from (Yuan ZH, 2022)

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

Unedited

Table 3 Comparison of the performance of proposed post-training quantization method AOCQ with other SOTA method for image classification on CIFAR-10

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

Table 4 Comparison of the performance of the proposed post-training quantization method AOCQ with other methods for object detection on COCO. The input resolution is set to 1333×800. * means quantization is only used for matrix multiplication, where add and other operations maintain float

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

Table 5 The effect of different components of AOCQ on ImageNet. ✓ means the method has been adopted

Model	Weight	Activation	operation balance	framework balance	loss balance	Top-1
Swin-Tiny	32	32				81.35
	4	8				66.41
	4	8	✓			78.93
	4	8	✓	✓		80.56
	4	8	✓	✓	✓	80.87

Table 6 The effect of different components of AOCQ on COCO. ✓ means the method has been adopted

Model	Weight	Activation	Operation balance	Framework balance	Loss balance	mAP
FRCNN Swin-T	32	32				45.5
	4	8				33.8
	4	8	✓			43.1
	4	8	✓	✓		44.2
	4	8	✓	✓	✓	44.8

Unedited

that Swin computes the self-attention locally within non-overlapping windows and the activation range is smaller.

Additionally, we conducted experiments on the downstream task using the CIFAR-10 dataset. Experimental results demonstrate that our method achieves a 1.84% gain over the BasePTQ method with W4A8 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% when compressed by 4× (8-bit quantization), and the performance only drops by 0.7% 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 Study

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

4.3.1 The effect of different components

In this section, we have individually verified the effectiveness of the three strategies (operation balance, framework balance, and loss balance) on classfication and detection tasks. From Table 5, it shows that adopting 4-bit quantization directly results in a performance drop of more than 14%. By merely employing the strategy of operation balance, the performance can be significantly improved by 12.48%. Furthermore, by adopting a two-step quantization strategy, the performance is restored, reaching 80.56. Finally, by introducing the strategy of loss balance, the

Table 7 The effect of steps. The results are evaluated on ImageNet. The samples are set to 256

Model	steps	Weight	Activation	Top-1
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

Table 8 The effect of samples. The results are evaluated on ImageNet. The steps are set to 1000

Model	steps	Weight	Activation	Top-1
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

performance achieves 80.87, with the performance drop compared to the float performance being only 0.48%. Experiments on detection in Table 6 also demonstrate the effect of our method.

4.3.2 The 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 W. During the second stage, it can be observed that the more iterations performs, the better the accuracy achieves. From Table 7, it is evident that after the number of steps ≥ 1000, the improvement in performance diminishes. Similarly, the Fig. 8 also 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 samples also make an effect on the results. From the table 8, it shows that 128 samples are enough to achieve the competitive results.

4.3.3 The loss in reconstruction process

In this section, 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 L2 norm ($\alpha = 2$) tends to amplify the imbalance between tokens, highlighting

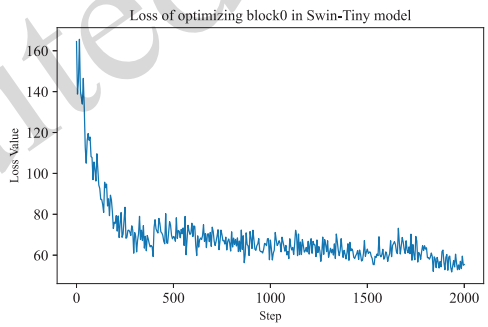


Fig. 8 The loss of different steps. The loss decreases as the steps increases

the significance of outliers. When using the L1 norm ($\alpha = 1$), the impact of outliers is diminished, which consequently improves performance. However, when $\alpha < 1$, the loss does not converge. The result also illustrates the effectiveness of balancing token in loss optimization.

4.4 Visualization

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 clearer illustration of the AOCQ quantization method's impact, we have 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

Unedited

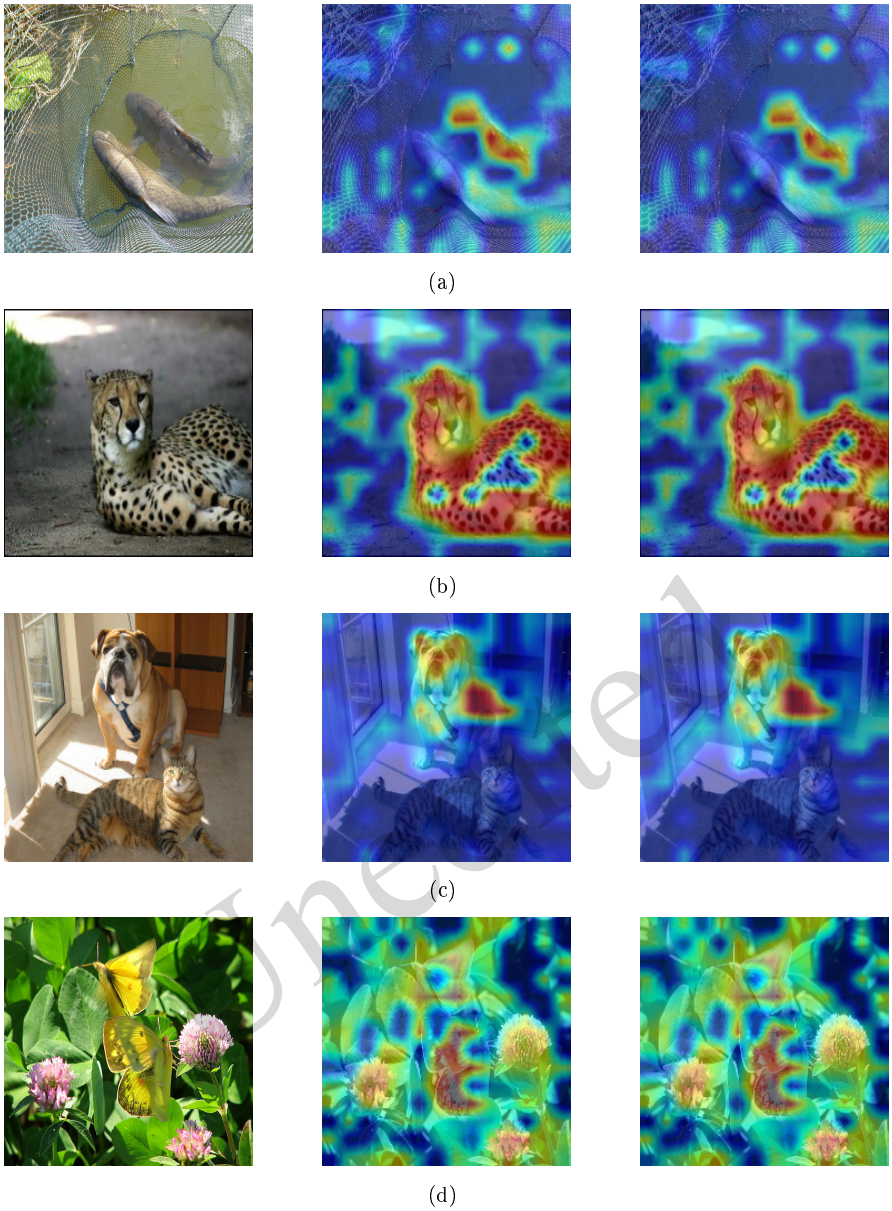


Fig. 9 Gradient-weighted class activation maps. We used Grad-CAM to visualize the attention maps of the ViT-Base model. The left is the original input image. The middle is Grad-CAM on FP32 model and the right is Grad-CAM on W4A8 quantized model with our method. All images were resized to 224x224 resolution, and the output of the last layer was used for visualization:(a)Gradient-weighted Class Activation Map on Fish;(b)Gradient-weighted Class Activation Map on Leopard;(c)Gradient-weighted Class Activation Map on Dog;(d)Gradient-weighted Class Activation Map on Butterfly

may be tiny differences in the background, like the upper left part of the leopard image in Fig. 9(b). This indicates our method using channel and token has a minimal effect on the target.

5 Conclusions

In this paper, we analyze the issue of outliers in transformers transformers and theoretically demonstrate that this phenomenon is caused by the structural design of the Transformer transformer itself. To address this problem, a new method (i.e., AOCQ) has been proposed, which balances the impact of out-

Table 9 Effect of loss balance

Model	loss	W/A	Top-1
Swin	-	32/32	81.35
	L2($\alpha = 2$)	4/8	80.56
	L1($\alpha = 1$)	4/8	80.62
	$\alpha = 0.5$	4/8	loss NaN
	token norm	4/8	80.87

'W/A' means weight/Activation. $||\cdot||^{\alpha}$ is default loss function, where $\alpha = 2$ is known as L2. In contrast, token norm is adopted in our approach

liers 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 only a 0.7% decrease in 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 LLM are greater compared to ViTs, which means 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 of the study, and participated in its design and coordination ,and helped to draft the paper.

Conflict of interest

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

References

Alam N Kolawole S SSea, 2023. Vision transformers for mobile applications: A short survey, . https://arxiv.org/abs/2305.19365

Ba J Grosse R MJ, 2017. Distributed second-order optimization using kronecker-factored approximations. *Proc 5th Int Conf on Learning Representations, p1-17*, .

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

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

Carion N Massa F SGea, 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 FJea. Autoformer: Searching transformers for visual recognition. *Proc IEEE/CVF Int Conf on Computer Vision*, . https://doi.org/10.1109/ICCV48922.2021.01205

Chen ZS Xie LX NJea, 2021. Visformer: The vision-friendly transformer. Proc IEEE/CVF Int Conf on Computer Vision, https://doi.org/10.1109/ICCV48922.2021.00063

Chitty-Venkata KT Mittal S EMea, 2023. A survey of techniques for optimizing transformer inference. *aJ Syst Archit*, . https://doi.org/10.1016/j.sysarc.2023.102990

Choi J Wang Z VSea, 2018. Pact: Parameterized clipping activation for quantized neural networks, https://arxiv.org/abs/1805.06085

Choromanski KM Likhosherstov V DDea, 2022. Rethinking attention with performers. Proc 9th Int Conf on Learning Representations.

Devlin J Chang MW LKea, 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc Conf North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, . https://doi.org/10.18653/v1/N19-1423

Ding MY Xiao B CNea, 2022. Davit: Dual attention vision transformers. *Proc 17th European Conf on Computer Vision*, . https://doi.org/10.1007/978-3-031-20053-3_5

Dong XY Bao JM CDea, 2022. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition*, . https://doi.org/10.1109/CVPR52688.2022.01181

Dosovitskiy A Beyer L KAea, 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc 9th Int Conf on Learning Representations*, .

Esser SK McKinstry JL BDea, 2020. Learned step size quantization.

Gong RH Liu XL JSea, 2019. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. Proc IEEE/CVF Int Conf on Computer Vision, https://doi.org/10.1109/ICCV.2019.00495

Graham B El-Nouby A THea, 2021. Levit: a vision transformer in convnet's clothing for faster inference. Proc IEEE/CVF Int Conf on Computer Vision, p.p.12259-12269, https://doi.org/10.1109/ICCV48922.2021.01204

Unedited

Grosse RB MJ, 2016. A kronecker-factored approximate fisher matrix for convolution layers. *Proc 33rd Int Conf on Machine Learning*, .

Ham TJ Jung SJ KSea, 2020. A^3: Accelerating attention mechanisms in neural networks with approximation. *Proc IEEE Int Symp on High Performance Computer Architecture*, .

Han DC Pan XR HYea, 2023. Flatten transformer: Vision transformer using focused linear attention. Proc IEEE/CVF Int Conf on Computer Vision, https://doi.org/10.1109/ICCV51070.2023.00548

Han S Pool J TJea, 2015. Learning both weights and connections for efficient neural network. Proc 29th Int Conf on Neural Information Processing Systems.

Hatamizadeh A Heinrich G YHea, 2023. Fastervit: Fast vision transformers with hierarchical attention. *Proc 12th Int Conf on Learning Representations*, .

Hinton G Vinyals O DJ, 2015. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38-39.

Hong K Dai GH XJea, 2023. Flashdecoding++: Faster large language model inference on gpus, https://arxiv.org/abs/2311.01282

Huang L Qin J LLea, 2020. Layer-wise conditioning analysis in exploring the learning dynamics of dnns. *Proc 16th European Conf on Computer Vision*, . https://doi.org/10.1007/978-3-030-58536-5_23

Le Cun Y Kanter I SS, 1990. Second order properties of error surfaces: learning time and generalization. *Proc 4th Int Conf on Neural Information Processing Systems*, .

LeCun YA Bottou L OGea, 2012. Efficient backprop. *Montavon G, Orr GB, Müller KB (Eds), Neural Networks: Tricks of the Trade*, :p.9-48. https://doi.org/10.1007/978-3-642-35289-8_3

Li F Zhang H LSea, 2022. Dn-detr: Accelerate detr training by introducing query denoising. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.p.13619-13627, https://doi.org/10.1109/CVPR52688.2022.01325

Li F Zhang H SPea, 2023. Semantic-sam: Segment and recognize anything at any granularity, . https://arxiv.org/abs/2307.04767

Li YH Gong RH TXea, 2021. Brecq: Pushing the limit of post-training quantization by block reconstruction. *Proc 9th Int Conf on Learning Representations*, .

Li ZK Ma LP CMea, 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 Chen MJ XJea, 2024. Psaq-vit v2: Towards accurate and general data-free quantization for vision transformers, https://doi.org/10.1109/TNNLS.2023.3301007

Li ZK Xiao JR YLea, 2023. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. Proc IEEE/CVF Int Conf on Computer Vision, p.p.17227-17236, https://doi.org/10.1109/ICCV51070.2023.01580

Liang JY Cao JZ SGea, 2021. Swinir: Image restoration using swin transformer. *Proc IEEE/CVF Int Conf on Computer Vision Workshops*, . https://doi.org/10.1109/ICCVW54120.2021.00210

Lin Y Zhang TY SPea, 2022. Fq-vit: Post-training quantization for fully quantized vision transformer. Proc 31st Int Joint Conf on Artificial Intelligence, p.1173-1179.

Liu SL Li F ZHea, 2022. DAB-DETR: Dynamic anchor boxes are better queries for DETR. Proc 10th Int Conf on Learning Representations.

Liu Z Lin YT CYea, 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. Proc IEEE/CVF Int Conf on Computer Vision, p.p.10012-10022, https://doi.org/10.1109/ICCV48922.2021.00986

Liu Z Ning J CYea, 2021b. Video swin transformer. *Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition*, . https://doi.org/10.1109/CVPR52688.2022.00320

Liu Z Hu H LYea, 2021c. Swin transformer v2: Scaling up capacity and resolution. *Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition*, . https://doi.org/10.1109/CVPR52688.2022.01170

Liu ZH Wang YH HKea, 2021. Post-training quantization for vision transformer. Proc 35th Int Conf on Neural Information Processing Systems.

Mehta S Ghazvininejad M ISea, 2021. Delight: Deep and light-weight transformer. *Proc 9th Int Conf on Learning Representations*, .

Nagel M Amjad RA vBMea, 2020. Up or down? adaptive rounding for post-training quantization.

Qu Z Liu L TFea, 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.

Ren SQ He KM GRea, 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Proc 29th Int Conf on Neural Information Processing Systems*, 28.

Sanh V Debut L CJea, 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, . https://arxiv.org/abs/1910.01108

Shen S Yao ZW GAea, 2020. Powernorm: Rethinking batch normalization in transformers. Proc 37th Int Conf on Machine Learning, p.8741-8751.

Si CY Yu WH ZPea, 2022. Inception transformer. *roc 36th Int Conf on Neural Information Processing Systems*, .

Touvron H Cord M DMea, 2021a. Training data-efficient image transformers & distillation through attention. Proc 38th Int Conf on Machine Learning.

Touvron H Cord M SAea, 2021b. Going deeper with image transformers. *Proc IEEE/CVF Int Conf on Computer Vision*, . https://doi.org/10.1109/ICCV48922.2021.00010

Vaswani A Shazeer N PNea, 2017. Attention is all you need. *Proc 31st Int Conf on Neural Information Processing Systems*, 30.

Yang QM Zhang K LCea, 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 ZMea, 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. Proc 36th Int Conf on Neural Information Processing Systems.

Yu XD Shi DH WXea, 2022. Soit: Segmenting objects with instance-aware transformers. *Proc 36th AAAI Conf on Artificial Intelligence*, .

Unedited

Yuan L Chen YP WTea, 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. Proc IEEE/CVF Int Conf on Computer Vision, p.558-567, https://doi.org/10.1109/ICCV48922.2021.00060

Yuan ZH Xue CH CYea, 2022. Ptq4vit: Post-training quantization framework for vision transformers, . https://arxiv.org/abs/2111.12293

Zafrir O Boudoukh G IPea, 2019. Q8bert: Quantized 8bit bert. Proc 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition.

Zhang H Li F LSea, 2023. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *Proc 11th Int Conf on Learning Representations*, .

Zhang XS Tian YJ XLea, 2023. Hivit: A simpler and more efficient design of hierarchical vision transformer. Proc 11th Int Conf on Learning Representations.

Zheng CY Li ZY ZKea, 2022. Savit: Structure-aware vision transformer pruning via collaborative optimization. Proc 36th Annual Conf on Neural Information Processing Systems.

Zhu XZ Su WJ LLea, 2020. Deformable detr: Deformable transformers for end-to-end object detection. *Proc 9th Int Conf on Learning Representations*, .

Unedited