

IPTQ-ViT: Post-Training Quantization of Non-linear Functions for Integer-only Vision Transformers

Gihwan Kim¹ Jemin Lee² Hyungshin Kim¹

¹Chungnam National University, Daejeon, Republic of Korea

²Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

gihwan.kim98@o.cnu.ac.kr leejaymin@etri.re.kr hyungshin@cnu.ac.kr

Abstract

*Previous Quantization-Aware Training (QAT) methods for vision transformers rely on expensive retraining to recover accuracy loss in non-linear layer quantization, limiting their use in resource-constrained environments. In contrast, existing Post-Training Quantization (PTQ) methods either partially quantize non-linear functions or adjust activation distributions to maintain accuracy but fail to achieve fully integer-only inference. In this paper, we introduce IPTQ-ViT, a novel PTQ framework for fully integer-only vision transformers without retraining. We present approximation functions: a polynomial-based GELU optimized for vision data and a bit-shifting-based Softmax designed to improve approximation accuracy in PTQ. In addition, we propose a unified metric integrating quantization sensitivity, perturbation, and computational cost to select the optimal approximation function per activation layer. IPTQ-ViT outperforms previous PTQ methods, achieving up to 6.44%*p* (avg. 1.78%*p*) top-1 accuracy improvement for image classification, 1.0 mAP for object detection. IPTQ-ViT outperforms partial floating-point PTQ methods under W8A8 and W4A8, and achieves accuracy and latency comparable to integer-only QAT methods. We plan to release our code¹.*

1. Introduction

Vision Transformers (ViTs) [7] have achieved state-of-the-art performance in various computer vision tasks [2, 3, 21, 25, 31, 33, 34]. However, their large model size and computational complexity pose considerable challenges for deployment on resource-constrained devices and mobile environments [8]. To address these challenges, model compression and inference acceleration techniques have been actively studied. Among these techniques, quantization methods that lower parameter precision effectively reduce both

model size and computational cost [8, 13, 22].

Integer-only quantization improves computational efficiency by replacing floating-point operations with integer arithmetic, thereby reducing data transfer, eliminating dequantization overhead, and fully utilizing hardware-efficient integer units [12, 14, 27, 28]. While this approach has shown strong success in Convolutional Neural Networks (CNNs) that primarily consist of linear operations [10, 28], applying it to vision transformers remains challenging. Vision transformers rely on non-linear functions such as Softmax, LayerNorm, and GELU, which are not naturally compatible with integer arithmetic and often require approximation or fake quantization for deployment. In addition, activation layers in vision transformers typically exhibit long-tailed and imbalanced distributions, making them highly sensitive to quantization and leading to performance degradation [17, 19, 20, 30].

To mitigate these issues, QAT methods [11, 12, 14] approximate all non-linear functions with integer-friendly operations and reduce quantization errors through retraining. However, QAT methods rely on high-performance GPUs and incur significant retraining overhead which limit their applicability in real-world deployments. In contrast, PTQ methods [11, 15–17, 19, 20, 24, 30] are deployment friendly as they enable the quantization of pretrained models without retraining. PTQ methods, however, struggle to mitigate the accuracy degradation caused by quantizing non-linear operations without retraining. Furthermore, existing PTQ approaches typically rely on partial or fake quantization for activation layers, failing to achieve fully integer-only inference. One notable attempt is FQ-ViT [19], a state-of-the-art PTQ method that applies quantization to LayerNorm and Softmax to enable integer-only inference. However, since GELU remains in floating point, it does not achieve a fully integer-only vision transformer.

In this paper, we present IPTQ-ViT, a novel PTQ framework for fully integer-only vision transformers that effectively mitigates accuracy degradation in PTQ caused by

¹<https://github.com/gihwan-kim/IPTQ-ViT.git>

quantized non-linear operations without retraining. QAT-based approximation functions, such as I-ViT [14] and I-BERT [12], can be directly applied to PTQ for vision transformers to address non-linear quantization challenges. However, we observe that heuristically applying these functions in PTQ leads to severe accuracy degradation, with accuracy dropping to as low as 0.08% in extreme cases (see Tab. 1). We analyze the reasons behind this accuracy drop and propose new approximation functions tailored to PTQ. Kim et al. [11] select approximation functions based on quantization sensitivity in QAT. We extend this idea to PTQ without retraining, thereby reducing analysis overhead. Our method also expands the function-assignment search space to enhance flexibility and accuracy. To support this, we propose a unified metric that integrates quantization sensitivity, perturbation, and computational cost, enabling efficient layer-wise function assignment.

Our main contributions are as follows:

1. We introduce a novel PTQ framework for integer-only vision transformers, fully quantizing both linear and non-linear operations without retraining.
2. We propose new approximation functions tailored for PTQ: *Data-aware Poly-GELU* optimized for vision data and *Efficient Bit-Softmax* which improves approximation accuracy.
3. We design *Unified Metric* that integrates quantization sensitivity, perturbation, and computational complexity to assign optimal approximation functions per activation layer.
4. We conduct extensive experiments on image classification, object detection, and latency evaluation. IPTQ-ViT outperforms existing PTQ methods and achieves accuracy and latency comparable to QAT methods.

2. Related Work

PTQ for Vision Transformers. PTQ methods have been proposed to efficiently deploy Vision Transformers on resource-constrained devices [8]. Previous studies focus on correcting activation distribution distortions that negatively impact quantization performance [17, 19, 30, 32]. RepQ-ViT [17] and FQ-ViT [19] tackle channel-wise variance in LayerNorm and imbalance in attention maps. PTQ4ViT [30] separates GELU and Softmax using Twin-Uniform quantization, while ERQ [32] corrects quantization errors via ridge regression. Recent works, data-free PTQ [15, 16, 24], explore synthetic calibration data to eliminate reliance on real data. However, these PTQ methods rely on partial floating-point operations or fail to fully quantize activation layers, limiting fully integer-only inference.

Non-linear Operation Quantization. Integer-only quantization replaces floating-point operations with integer arithmetic, offering significant efficiency gains [27], and has been successfully applied to CNNs [10, 27]. However, ap-

plying integer-only quantization to ViTs and Large Language Models (LLMs) presents challenges due to the non-linearity of activation layers such as Softmax, GELU, and LayerNorm. I-BERT [12] proposes polynomial approximations for GELU and Softmax in language models. FQ-ViT [19] proposes log2 quantization and i-exp [12] to approximate Softmax, utilizing power-of-two scaling for LayerNorm in PTQ. ShiftAddLLM [29] streamlines these approximations to utilize only shift-add operations. I-ViT [14] approximates Softmax and GELU using bit-shifting and corrects quantization error with retraining, while Kim et al. [11] select optimal functions per layer using SQNR-based metrics in QAT. I-LLM [9] extends bit-shifting approximation to LLMs.

Limitations of Non-linear Function Quantization. The methods [9, 12, 23, 29] tailored for LLMs often fail to generalize to vision tasks, as evidenced by the performance degradation of I-BERT [12] when applied to QAT-based ViTs [12]. QAT-based ViT methods [11, 12, 14] require significant retraining costs, powerful GPU resources, and the accessibility of the complete training dataset, which limits their applicability in deployment. PTQ-based methods [19] offer greater ease of deployment; however, they do not involve retraining, resulting in increased sensitivity to approximation error.

3. Motivation

To achieve integer-only vision transformers, approximation functions for all non-linear operations such as Softmax, GELU, and LayerNorm are essential. Existing QAT methods [11, 12, 14] reduce approximation errors through retraining; however, these are dependent on training resources and hard to generalize to diverse model architectures. They also require full training datasets and involve complex hyperparameter tuning, which limits their applicability in real-world deployments. Conversely, PTQ methods enable quantization without retraining but fail to fully quantize non-linear operations to integer operations, resulting in accuracy degradation. An alternative approach is to directly use QAT-based approximation functions [12, 14] in PTQ setting as a heuristic solution to the limitations of prior quantization methods. However, as shown in Tab. 1, these heuristic applications of QAT-based functions (denoted as I-ViT* and I-BERT*) result in a severe drop in accuracy to 0.08% under both W8A8 and W4A8 settings. This performance loss is due to two key issues: (1) existing approximation functions are tailored for language data distributions and do not generalize well to vision tasks, and (2) PTQ performs quantization without retraining, making it difficult to compensate for non-linear function approximation errors.

Fig. 1 visualizes the distorted activation distributions of the GELU layer in ViT-B when employing I-BERT’s approximation function. Compared to Fig. 1 (a), Fig. 1 (b)

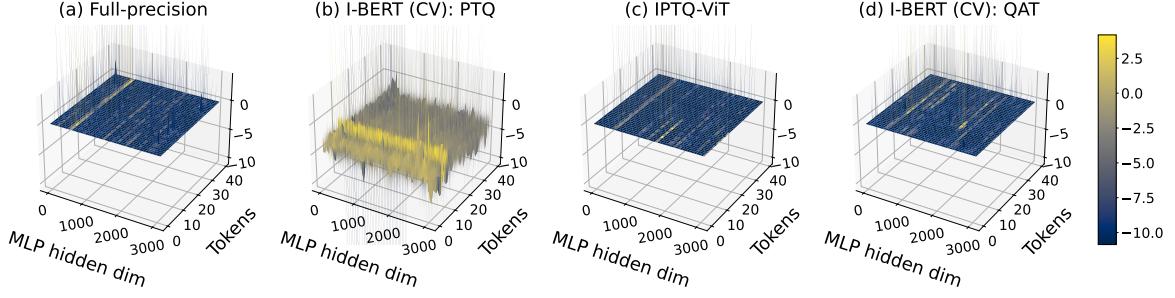


Figure 1. Activation distributions of GELU in the 11-th block of ViT-B, which shows the highest quantization sensitivity in Tab. 2 (I-BERT*). Visualized for (a) full-precision, (b) PTQ-quantized I-BERT, (c) our method, and (d) QAT-quantized I-BERT, with token subsampling applied. Both (b) and (d) use i-GELU [12] of the QAT-based approximation. (b) shows massive imbalance, highlighting the limitation of applying QAT-designed methods to PTQ settings in vision tasks. More results are presented in Appendix Fig 4.

demonstrates that an approximation function designed for language models fails to sustain a stable distribution under PTQ, leading to increased quantization error and significant performance degradation. In contrast, Fig. 1 (d) shows that retraining can stabilize the activation distribution even with language-model-based approximations. Additionally, Tab. 2 shows an analysis of quantization sensitivity in Signal-to-Quantization Noise Ratio (SQNR) for the GELU and Softmax layers of ViT-B. A lower SQNR indicates higher quantization errors, meaning the layer is more sensitive to quantization and leads to accuracy loss. The sensitivity dramatically increases in deeper layers for heuristic application of QAT-based functions (I-BERT* and I-ViT*). Notably, I-ViT* reports a $3.2\times$ increase in quantization sensitivity at the 5-th block compared to the 4-th block. Such increased sensitivities observed in both methods indicate that their simple approximation functions fail to reduce quantization errors in PTQ, leading to accuracy degradation.

To address these limitations, we propose approximation methods and a quantization pipeline that enables integer-only ViTs without retraining. Our IPTQ-ViT outperforms FQ-ViT, a state-of-the-art PTQ approach for non-linear operations, achieving higher accuracy under both W8A8 and W4A8 settings, as shown in Tab. 1.

4. Method

4.1. Background

A quantizer is defined to map a real-valued input $X \in \mathbb{R}$ to an integer value q within the range of b -bit integers, $q = \text{clip}(\lfloor X/s \rfloor + z, 0, 2^b - 1)$, where the scale $s = (\alpha - \beta)/(2^b - 1)$ and zero-point $z = \text{clip}(\lceil -\beta/s \rceil, 0, 2^b - 1)$. The quantization function is applied across the entire model. All linear (*Conv*, *Linear*, *MatMul*) and non-linear (*Softmax*, *GELU*, *LayerNorm*) operations are thus executed with integer arithmetic, enabling integer-only vision transformers.

Method	W/A	DeiT-T	DeiT-S	DeiT-B	ViT-B	Swin-T	Swin-S
Baseline	FP	72.21	79.85	81.85	84.53	81.35	83.20
I-ViT*	8/8	61.66	49.65	0.10	0.10	59.39	0.10
	4/8	58.98	56.44	0.33	0.34	64.93	0.39
I-BERT*	8/8	68.37	77.31	80.88	81.71	35	82.49
	4/8	0.08	0.10	0.10	0.09	0.10	0.10
FQ-ViT [19]	8/8	71.61	79.17	81.20	83.31	81.29	82.13
	4/8	66.91	76.93	79.99	78.73	80.73	81.67
IPTQ-ViT	8/8	72.10	79.76	81.84	84.19	81.09	83.08
	4/8	66.90	77.28	80.98	82.03	79.13	82.53

Table 1. Top-1 accuracy (%) of QAT-based non-linear methods under PTQ on vision transformers (ImageNet-1k). * indicates direct PTQ application of QAT methods (I-ViT [14], I-BERT [12]) using official code. "W/A" denotes weight/activation bit-width.

Method	Model	blk1	blk2	blk3	blk4	blk5	blk6
I-ViT*	ViT-B	0.50	-1.10	-4.15	-7.50	-24.13	-32.66
I-BERT*	ViT-B	-11.25	-16.00	-11.44	-12.13	-12.40	-12.47
Method	Model	blk6	blk7	blk8	blk9	blk10	blk11
I-ViT*	ViT-B	-34.65	-27.42	-24.61	-21.04	-17.22	-10.72
I-BERT*	ViT-B	-11.35	-8.98	-13.43	-22.62	-25.08	-17.81

Table 2. Layer-wise quantization sensitivity (SQNR in dB, ↑ better) of QAT-based approximation functions on ViT-B under PTQ. I-ViT* applies Shiftmax [14] to Softmax, and I-BERT* uses i-GELU [12] for GELU.

4.2. Overall Pipeline

Fig. 2 illustrates the overall quantization pipeline of IPTQ-ViT. The framework consists of three stages: (1) conducting layer-wise *Unified Metric* analysis, (2) assigning approximation functions to each non-linear layer based on the metric, (3) calibrating the mixed quantized model.

The search space of approximation functions is defined using four types: bit-shifting [14], polynomial [12], logarithm [19] and our proposed methods. The supported search space for approximation functions varies by non-linear layer type: (i) Softmax — logarithm, polynomial, bit-

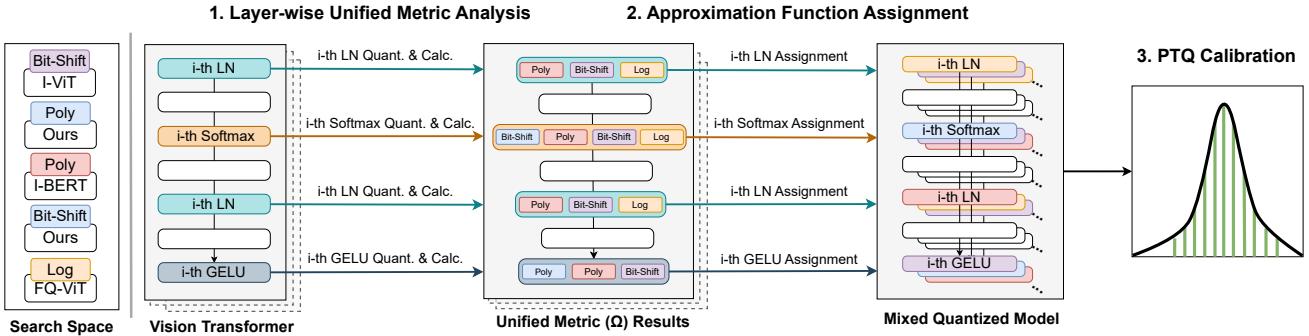


Figure 2. Overview of IPTQ-ViT pipeline. In stage 1, each non-linear layer is quantized with all candidate approximation functions and the Unified Metric is computed for each case. Stage 2 assigns an approximation function that has a maximum metric value per activation layer. Stage 3 calibrates the mixed quantized model.

shifting, and our *Efficient Bit-Softmax*; (ii) GELU — polynomial, bit-shifting, and our *Data-aware Poly-GELU*; (iii) LayerNorm — logarithm, polynomial, and bit-shifting. For each non-linear layer, we compute a metric for all candidate approximation functions and assign the one with the highest score. This layer-wise selection forms a mixed-quantized model. For instance, ViT-B contains 49 activation layers: 2 LayerNorms, 1 Softmax, and 1 GELU per Transformer block (12 blocks), plus one additional LayerNorm before the first block. Thus, the full model can be constructed with 159 computations. Finally, PTQ calibration is applied to this model to finalize quantization parameters.

4.3. Data-aware Poly-GELU for Integer-only GELU

The proposed *Data-aware Poly-GELU* is a polynomial-based GELU approximation optimized for vision tasks in PTQ to support integer-only inference. It addresses the limitations of i-GELU [12], originally designed for language models in I-BERT [12], as discussed in Section 3.

$$\text{GELU}(x) = \frac{1}{2} \cdot x \cdot \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right) \right) \quad (1)$$

I-BERT [12] approximates the error function (erf) with a second-order polynomial, denoted as L_{libert} , to implement integer-only GELU inference. In Eq. (2), i-GELU [12] directly solves an optimization problem targeting the GELU function defined in Eq. (1). However, this approach introduces complexity due to multiplicative terms: x , $1/2$, and $1 + \text{erf}(\frac{x}{\sqrt{2}})$, making the optimization sensitive to input distribution. To solve the problem, we simplify the optimization problem of I-BERT [12] by formulating the approximation as an optimization over its core component, the error function (erf), as shown in Eq. (4). This simplification allows for a more stable and accurate approximation of erf.

$$\begin{aligned} \min_{a,b,c} \frac{1}{2} & \left\| \text{GELU}(x) - x \cdot \frac{1}{2} \left[1 + L_{\text{libert}}\left(\frac{x}{\sqrt{2}}\right) \right] \right\|_2^2 \\ \text{s.t. } & L_{\text{libert}}(x) = a(x+b)^2 + c \end{aligned} \quad (2)$$

In contrast to the fixed approximation range based on language data used by the previous method [12], our approach determines the approximation range by calculating the minimum and maximum values from the vision data and recalculates the polynomial coefficients accordingly. Further details on the approximation range and its effect can be found in Appendix G.1. Additionally, we extend a quartic polynomial to enhance accuracy for erf approximation, defined in Eq. (3), with coefficients a is -0.019913 and b is -2.698088 . These coefficients are pre-quantized as a constant before inference.

$$L_{\text{ours}}(x) = \text{sign}(x) \cdot [a \cdot (\text{clip}(|x|, \text{max} = -b) + b)]^4 + 1 \quad (3)$$

$$\arg \min_{a,b} \sum_{i=1}^N \|\text{erf}(x_i) - L_{\text{ours}}(x_i; a, b)\|_2^2 \quad (4)$$

for $x \in [\min(T), \max(T)]$

The resulting polynomial GELU function is defined in Eq. (5). *Data-aware Poly-GELU* establishes a trade-off between computational cost and accuracy depending on the polynomial degree. We empirically determine the optimal polynomial degree by evaluating model accuracy across different degrees within our quantization pipeline.

$$\text{Data-aware-Poly-GELU}(x) = \frac{1}{2} \cdot x \cdot \left[1 + L_{\text{ours}}\left(\frac{x}{\sqrt{2}}\right) \right] \quad (5)$$

We evaluated our polynomial GELU approximation (Eq. (5)) with varying polynomial degrees on ImageNet-1k using our quantization pipeline. As the degree increases, accuracy consistently improves: 79.24%, 79.30%, and 79.76% for degrees 2, 3, and 4 on DeiT-S, and 81.22%, 81.30%, and 81.84% on DeiT-B, respectively. Although a quartic polynomial incurs higher computational cost, we reduce this overhead by reusing integer and squared terms. Given its accuracy gain, we select the quartic polynomial. Fig. 3 illustrates our proposed methods for erf and GELU, showing their close similarity to the original functions. As

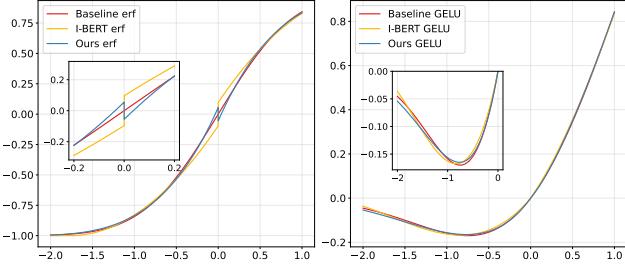


Figure 3. Left: Comparison of erf approximations by baseline, I-BERT [12], and ours. Right: Comparison of GELU approximations by baseline, i-GELU [12], and ours.

shown in Tab. 3, both the L^2 and L^∞ approximation errors of our proposed method are reduced compared to previous methods [12].

Method	L^2 error	L^∞ error
erf I-BERT [12]	0.0264	0.0962
erf Eq. (3) (Ours)	0.0098	0.0550
i-GELU [12]	0.0094	0.0182
Data-aware Poly-GELU (Ours)	0.0051	0.0093

Table 3. Comparison of our method and I-BERT [12] for approximating erf and GELU. L^2 and L^∞ errors are computed over the range $(-3, 3)$.

4.4. Efficient Bit-exp for Integer-only Softmax

Softmax converts input values into a probability distribution, representing the relative importance of each element. To prevent overflow of Softmax approximation, previous methods [12, 14, 19] generally subtract the maximum value from the input of the exponential function. We apply the same stabilization in Eq. (6).

$$\text{Softmax}(x_i) = \frac{e^{S_{x_i} \cdot (Q_{x_i} - Q_{\max})}}{\sum_{j=1}^d e^{S_{x_j} \cdot (Q_{x_j} - Q_{\max})}} = \frac{e^{S_{\Delta_i} \cdot Q_{\Delta_i}}}{\sum_{j=1}^d e^{S_{\Delta_j} \cdot Q_{\Delta_j}}} \quad (6)$$

In Section 3, we identified significant accuracy degradation when directly applying I-ViT [14] to PTQ. Here, we highlight the overly simple bit-shift approximation (Shiftmax) used in I-ViT [14] as a primary cause. Firstly, we summarize the Shiftmax [14] approximation process. To approximate Softmax, one must approximate the exponential function, whose input is expressed as a product of scaling factor S_Δ and quantized tensor Q_Δ . The exponential function is reformulated into base-2 exponential form, as shown in Eq. (7). In Eq. (8), $S_\Delta \cdot Q_\Delta$ is decomposed into an integer component $-q$ and a fractional component $S_\Delta \cdot (-r)$. An integer component $-q$ is represented using bit-shift operations, as defined in Eq. (9). I-ViT [14] further approximates the left-hand side of Eq. (10) by a linear function in the form

of $1 + x/2$.

$$e^{S_\Delta \cdot Q_\Delta} = 2^{S_\Delta \cdot Q_\Delta \cdot \log_2 e} \approx 2^{S_\Delta \cdot (Q_\Delta + (Q_\Delta \gg 1) - (Q_\Delta \gg 4))} = 2^{S_\Delta \cdot Q_p} \quad (7)$$

$$2^{S_\Delta \cdot Q_p} = 2^{-q + S_\Delta \cdot (-r)} \quad (8)$$

$$= 2^{S_\Delta \cdot (-r)} \gg q \quad (9)$$

$$2^{S_\Delta \cdot (-r)} \approx [S_\Delta \cdot (-r)]/2 + 1 \quad (10)$$

While retraining in QAT can compensate for quantization errors introduced by bit-shift approximations, it is not feasible in PTQ. To address the limitation, we propose an approximation function, *Efficient Bit-exp*. We reformulate the left-hand side of Eq. (10) as a base-2 exponential function, replacing the overly simple linear form. It is then approximated with a Taylor series, as defined in Eq. (11). High-order polynomials increase computational cost. Given the limited range of fractional inputs $S_\Delta \cdot (-r)$, we adopt a first-degree polynomial (Eq. (12)) to minimize computational overhead. Furthermore, we approximate the constant $\ln 2$ as the binary value $(0.1011)_b$, allowing us to implement the exponential function solely with bit-shift and addition operations, eliminating complex multiplications. Our improved exponential approximation is defined in Eq. (14).

$$2^X = e^{\ln 2 \cdot X} \approx \sum_{d=0}^D \frac{(\ln 2 \cdot X)^d}{d!} \quad (11)$$

$$\approx 1 + \ln 2 \cdot X \quad \text{for } D = 1 \quad (12)$$

$$2^{S_\Delta \cdot (-r)} \approx [\Phi(S_\Delta \cdot (-r))] + 1 \quad (13)$$

$$\text{s.t. } \Phi(x) = x \gg 1 + x \gg 3 + x \gg 4 \quad (14)$$

$$\text{Efficient-Bit-exp}(x) = S_\Delta \cdot [\Phi(-r) + \lfloor 1/S_\Delta \rfloor] \quad (14)$$

Efficient Bit-exp approximates the numerator of Eq. (6), while the denominator is represented by the sum of these approximations. To generate the probability distribution, we adopt integer division (IntDiv) introduced by I-ViT [14]. Based on *Efficient Bit-exp* and IntDiv, we define our integer-only Softmax function, *Efficient Bit-Softmax*, as shown in Eq. (15), where M is a large integer to prevent overflow and b denotes the bit-width. This precise exponential approximation enables integer-only Softmax computation under PTQ.

$$\begin{aligned} \text{Efficient-Bit-Softmax}(x) &= \frac{Q_{\exp_i}}{\sum_j^d Q_{\exp_j}} \\ &= \text{IntDiv}\left(Q_{\exp_i}, \sum_j^d Q_{\exp_j}, b\right) \\ &= \left(\left\lfloor \frac{2^M}{\sum_j^d Q_{\exp_j}} \right\rfloor \cdot Q_{\exp_i}\right) \\ &\gg (M - (b - 1)) \quad (15) \end{aligned}$$

$$\text{s.t. } Q_{\exp_i} = \text{Efficient-Bit-exp}(Q_i)$$

As shown in Tab. 4, the first-degree approximation (“Ours-D1”) of Eq. (11) achieves higher accuracy than the

second-degree (“Ours-D2”) across all evaluated models and bit-widths. Therefore, we choose a first-degree polynomial approximation to balance accuracy and efficiency. A detailed analysis of the lower accuracy of the second-order approximation is provided in Appendix H. Tab. 5 presents a comparison of the L^2 and L^∞ approximation errors between our method and I-ViT [14] in the fractional range $(-1, 1)$. Our method achieves higher approximation accuracy than I-ViT [14], facilitating an efficient integer-only implementation appropriate for PTQ.

Method	W/A	DeiT-T	DeiT-S	DeiT-B	ViT-B	Swin-T	Swin-S
Ours-D1	8/8	72.10	79.76	81.84	84.19	81.19	83.08
Ours-D2	8/8	71.93	79.37	81.62	84.03	80.90	82.95
Ours-D1	4/8	66.90	77.28	80.98	82.03	79.13	82.53
Ours-D2	4/8	65.93	76.16	80.01	80.87	78.59	81.62

Table 4. Top-1 accuracy (%) for different polynomial degrees of Eq. (15) on the ImageNet-1K. “D” denotes degree. Degree 1 shows better performance on W8A8 and W4A8 than degree 2.

Method	L^2 error	L^∞ error
base-2 exp (I-ViT [14])	0.1717	0.5
base-2 exp (Ours)	0.1126	0.3069

Table 5. Comparison of base-2 exponential approximation functions. L^2 and L^∞ errors are evaluated over the range $(-1, 1)$.

4.5. Unified Metric for Approximation Function Assignment

In order to assign an optimal non-linear approximation function to each activation layer in PTQ, we propose a *Unified Metric* that jointly considers three factors: quantization sensitivity (\mathcal{Q}), quantization perturbation (\mathcal{P}), and computational cost (\mathcal{C}). The approximation function with the highest *Unified Metric* score is selected for each layer during stage 2 of the pipeline (Fig. 2).

Although SQNR is widely used to assess quantization sensitivity, it measures relative error on a logarithmic scale and may obscure meaningful differences in absolute error. For example, in Eq. 18, assuming an input power of $\mathbb{E}[(X)^2] = 10^4$, perturbations of 100 and 60 correspond to SQNR values of 20 dB and 22.21 dB, respectively. While the SQNR difference is only 2.21 dB, the absolute error difference is 40, which may significantly affect accuracy in deeper layers. This suggests that SQNR alone may not adequately capture error accumulation. To address this limitation, we additionally incorporate \mathcal{P} , as defined in Eq. (19), jointly considering both SQNR and perturbation provides richer guidance for assigning appropriate approximation functions.

To estimate the efficiency of each approximation function, we include the number of arithmetic and bit operations

in *Unified Metric* (Ω), counting as in FLOPs. Similar to prior quantization works, we use this operation count as an indirect metric for computational cost. Since the factors of Ω differ in scales and signs, we apply the Softplus function as defined in Eq. (17) to normalize components of the metric. The transformed $N(\mathcal{P})$ and $N(\mathcal{C})$ are converted to reciprocal forms, as lower values are preferable. As shown in Eq. (16), *Unified Metric* is defined as the harmonic mean of the three factors to balance their contributions, which prevents disproportionate influence from extreme values.

$$\Omega = \sum_i^L \frac{3}{N(\mathcal{Q}_i)^{-1} + N(\mathcal{P}_i) + N(\mathcal{C}_i)} \quad (16)$$

$$N(x) = \log(1 + \exp(x)) \quad (17)$$

$$\mathcal{Q} = 20 \log \frac{\mathbb{E}[X^2]}{\mathbb{E}[(X - Q)^2]} \quad (18)$$

$$\mathcal{P} = \|X - Q\|_2^2 \quad (19)$$

$$\mathcal{C} = \text{Integer operation count of the layer} \quad (20)$$

5. Experiments

5.1. Experimental Setup

We evaluate IPTQ-ViT on both image classification and object detection with W8A8 and W4A8 settings, where “W” and “A” denote weight and activation bit-width. We adopt symmetric, channel-wise quantization for weights and asymmetric, layer-wise quantization for activations. For comparison, we include PTQ methods [15–17, 20, 24, 30] and integer-only QAT baselines (I-BERT [12], I-ViT [14]), with their non-linear approximations re-implemented under PTQ (denoted *). All experiments are conducted on a single NVIDIA RTX 3090 GPU.

Image Classification. We benchmark ViT [7], DeiT [26], and Swin [21] on ImageNet-1K [6], reporting top-1 accuracy against PTQ and integer-only QAT methods.

Object Detection. We evaluate Swin-T and Swin-S backbones within Cascade Mask R-CNN [1] on COCO [18] using MMDetection [4], comparing box and mask AP with prior PTQ baselines.

Latency. We measure end-to-end latency of batch size 8 by deploying IPTQ-ViT with TVM [5] in W8A8. I-ViT [14] is the only prior integer-only ViTs reporting latency, but its evaluation recipe (e.g., auto-tuning settings, warm-up iterations, and repetition counts) was not disclosed. To ensure a fair comparison, we evaluated I-ViT [14], the FP32 baseline, and our method under an identical evaluation setting. Full details are provided in Appendix J.

Calibration. We use randomly sampled training images. Results under varying calibration set sizes are provided in Appendix C.3 and D.1.

Method	Type	Opt	W/A	DeiT-T	Diff	DeiT-S	Diff	DeiT-B	Diff	ViT-B	Diff	Swin-T	Diff	Swin-S	Diff	
Baseline		FP	FP	FP	72.21	-	79.85	-	81.85	-	84.53	-	81.35	-	83.20	-
I-BERT [12]	QAT	IO	8/8	71.33	+0.77	79.11	+0.65	80.79	+1.05	83.70	+0.49	80.15	+0.94	81.86	+1.22	
I-ViT [14]		IO		72.24	-0.14	80.12	-0.36	81.74	+0.10	84.76	-0.57	81.50	-0.41	83.01	+0.07	
I-ViT*		IO		61.66	+10.44	49.65	+30.11	0.10	+81.74	59.39	+21.70	0.10	+82.98			
I-BERT*		IO		68.37	+3.73	77.31	+2.44	80.88	+0.96	81.71	+2.48	35.00	+46.09	82.49	+0.59	
FQ-ViT [19]		PF		71.61	+0.49	79.17	+0.59	81.20	+0.64	83.31	+0.88	81.29	-0.20	82.13	+0.95	
PTQ4ViT [30]		PF		71.72	+0.38	79.47	+0.29	81.48	+0.36	84.25	-0.06	81.24	-0.15	83.10	-0.02	
RepQ-ViT [17]	PTQ	PF	8/8	72.05	+0.05	79.55	+0.21	81.45	+0.39	81.45	+2.74	81.28	-0.19	82.34	+0.74	
PSAQ-ViT V1 [15]		PF		71.56	+0.54	76.92	+2.84	79.10	+2.74	37.36	+46.83	75.35	+5.74	76.64	+6.44	
PSAQ-ViT V2 [16]		PF		72.17	-0.07	79.56	+0.20	81.52	+0.32	N/A	N/A	80.20	+0.89	82.13	+0.95	
NoisyQuant-Linear [20]		PF		N/A	N/A	79.11	+0.65	81.30	+0.54	84.10	+0.09	81.05	+0.04	83.07	+0.01	
NoisyQuant-PTQ4ViT [20]		PF		N/A	N/A	79.51	+0.25	81.45	+0.39	84.22	-0.03	81.25	-0.16	83.13	-0.05	
CLAMP-ViT [24]		PF		72.17	-0.07	79.55	+0.21	81.77	+0.07	84.19	+0.00	81.17	-0.08	82.50	+0.58	
IPTQ-ViT	PTQ	IO	8/8		72.10		79.76		81.84		84.19		81.09		83.08	
FQ-ViT [19]		PF		66.91	-0.01	76.93	+0.35	79.99	+0.99	78.73	+3.30	80.73	-1.60	81.67	+0.86	
I-ViT*		IO		58.98	+7.92	56.44	+20.84	0.33	+80.65	0.34	+81.69	64.93	+14.20	0.39	+82.14	
I-BERT*		IO		0.08	+66.82	0.10	+77.18	0.10	+80.88	0.09	+81.94	0.10	+79.03	0.10	+82.43	
PTQ4ViT [30]		PF		66.57	+0.33	76.96	+0.32	79.47	+1.51	67.99	+14.04	N/A	N/A	79.62	+2.91	
RepQ-ViT [17]	PTQ	PF	4/8	68.75	-1.85	76.75	+0.53	80.12	+0.86	76.29	+5.74	80.51	-1.38	82.14	+0.39	
PSAQ-ViT V1 [15]		PF		65.57	+1.33	73.23	+0.45	77.05	+3.93	25.34	+56.69	71.79	+7.34	75.14	+7.39	
PSAQ-ViT V2 [16]		PF		68.61	-1.71	76.36	+0.92	79.49	+1.49	N/A	N/A	76.28	+2.85	78.86	+3.67	
CLAMP-ViT [24]		PF		69.93	-3.03	77.03	+0.25	80.93	+0.05	78.73	+3.30	80.28	-1.15	82.51	+0.02	
IPTQ-ViT	PTQ	IO	4/8		66.90		77.28		80.98		82.03		79.13		82.53	

Table 6. Top-1 accuracy (%) of quantized ViTs on image classification with the ImageNet-1k. * denotes results reproduced with the official code. "Diff" denotes the accuracy gap between IPTQ-ViT and compared methods. **Green** and **bold** indicate IPTQ-ViT improvements. "IO": all ops computed in integer arithmetic. "PF": at least one op executed in floating point.

SearchSpace / Metric	GELU	Softmax	SQNR	Unified	DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S
<i>Baseline FQ-ViT [19]</i>									
Legacy / SQNR			✓		71.61	79.17	81.20	81.29	82.13
Legacy / Unified				✓	71.48	78.83	80.99	78.96	82.27
Extended (Softmax) / SQNR		✓	✓		71.16	79.03	81.20	80.01	82.3
Extended (GELU) / SQNR	✓		✓		72.08	79.52	81.67	80.09	82.37
Extended / SQNR	✓	✓	✓		71.93	79.50	81.28	80.17	82.43
Extended / Unified	✓	✓		✓	71.97	79.66	81.73	80.42	82.57

Table 7. Ablation study on ImageNet-1K (top-1 accuracy, %) under W8A8. ✓ indicates the use of a component: *Data-aware Poly-GELU* ("GELU"), *Efficient Bit-Softmax* ("Softmax"), and *Unified Metric* ("Unified"). "Legacy" denotes the search space with prior approximations (I-ViT [14], I-BERT [12], FQ-ViT [19]), and "Extended" augments it with our proposed methods.

5.2. Evaluation on Image Classification

As shown in Tab. 6, IPTQ-ViT consistently matches or has better accuracy at W8A8 and W4A8. In W8A8, IPTQ-ViT achieves average accuracy gains of 0.66%p on DeiT-S, 0.68%p on DeiT-B, and 7.21%p on Swin-S compared to other PTQ methods, except NoisyQuant-PTQ4ViT [20] on Swin-S. IPTQ-ViT surpasses FQ-ViT [19], the SOTA PTQ method for fully quantized ViTs, on all models except Swin-T, while maintaining integer-only inference. Against integer-only QAT methods [12, 14], IPTQ-ViT consistently outperforms I-BERT [12] and exceeds I-ViT [14] on DeiT-B and Swin-S, without any retraining. These results demonstrate the effectiveness of our methods. A detailed ablation study is presented in Section 6. Moreover, directly applying QAT-based functions in PTQ (I-BERT* and I-ViT*) leads to considerable accuracy degradation in both W8A8 and W4A8, highlighting the difficulty of approximating non-

linear operations without retraining.

5.3. Evaluation on Object Detection

In Tab. 8, IPTQ-ViT delivers competitive performance across Swin-T and Swin-S backbones under various quantization settings. Notably, with Swin-S at W8A8, we surpass FQ-ViT [19] by +1.0 box AP and +0.7 mask AP. Compared to RepQ-ViT [17] and CLAMP-ViT [24], which do not quantize non-linear layers, IPTQ-ViT achieves comparable accuracy. For instance, on Swin-T (W6A6), IPTQ-ViT is only -0.4 box AP and -0.3 mask AP lower than RepQ-ViT [17]. As discussed in Section 3, we observed severe accuracy degradation also on object detection when directly applying QAT-based approximation functions to PTQ. I-ViT* and I-BERT* show average drops of 42.55 box AP and 36.63 mask AP under W4A8 compared to W8A8.

Model	Method	W8A8		W6A6		W4A8	
		AP _{box}	AP _{mask}	AP _{box}	AP _{mask}	AP _{box}	AP _{mask}
Swin-S	Baseline (FP)	51.8	44.7	51.8	44.7	51.8	44.7
	FQ-ViT [19]	50.8	44.1	N/A	N/A	48.2	41.3
	I-ViT*	49.1	42.4	1.2	1.0	0.3	0.3
	I-BERT*	50.4	43.5	41.7	35	14.9	13.6
	PSAQ-ViT V2 [16]	50.9	44.1	N/A	N/A	47.9	41.4
	PTQ4-ViT [17]	20.8	18.7	12.5	10.8	38.5	33.8
	RepQ-ViT [17]	51.6	44.6	51.4	44.6	49.2	42.8
Swin-T	CLAMP-ViT [24]	51.4	44.6	N/A	N/A	48.5	42.2
	IPTQ-ViT	51.8	44.8	51.4	44.5	48.2	41.8
	Baseline (FP)	50.4	43.7	50.4	43.7	50.4	43.7
	FQ-ViT [19]	49.7	43.3	N/A	N/A	N/A	N/A
RepQ-ViT [17]	I-ViT*	48.3	42.7	2.9	2.6	1.3	1.2
	I-BERT*	49.7	43.1	40.9	35.5	10.8	10.1
	PTQ4-ViT [17]	40.3	35.6	14.7	13.6	25.3	22.7
	RepQ-ViT [17]	NA	NA	50.0	43.5	NA	NA
	IPTQ-ViT	50.4	43.7	49.6	43.2	43.1	37.7

Table 8. Object detection on COCO. * indicates results reproduced using official code. **Bold** indicates the top performance methods.

5.4. Quantization Runtime

In Fig. 4, we report the quantization runtime and ImageNet-1K top-1 accuracy of IPTQ-ViT compared with prior PTQ methods on DeiT-S (W8A8). The runtime measures only the quantization process, excluding inference. IPTQ-ViT completes quantization in 2.37 minutes and achieves higher accuracy than all baselines. In contrast, PSAQ-ViT V1 [15] and CLAMP-ViT [24] require extra procedures such as synthetic data generation, analysis and PTQ calibration, which increase overhead. FQ-ViT [19] records the shortest runtime, its accuracy is noticeably lower than IPTQ-ViT. The results of additional models are shown in Appendix E.

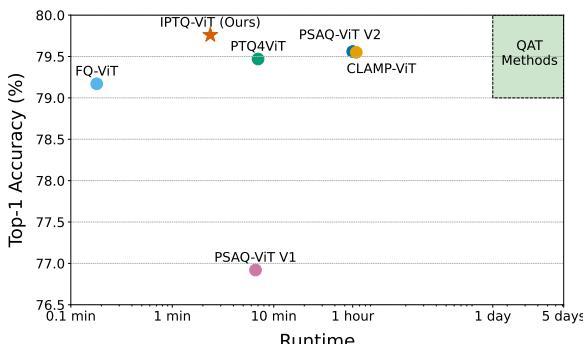


Figure 4. Quantization runtime on *DeiT-S* under W8A8, measured on a single NVIDIA RTX 3090 GPU. Times exclude inference.

5.5. Latency Evaluation

Tab. 9 reports the latency of DeiT-T, DeiT-S, and DeiT-B under W8A8. For comparison, we also include the original I-ViT [14] results on an RTX 2080Ti GPU and our re-implementation I-ViT^{\$} on the same RTX 3090 GPU. IPTQ-ViT achieves a comparable latency to I-ViT^{\$} on a

3090 GPU, while maintaining +0.10%p higher top-1 accuracy on DeiT-B without retraining. It consistently delivers a 1.9–2.6× speedup over the FP32 baseline, with DeiT-B reduced from 18.7 ms to 7.11 ms (2.63× faster). These results demonstrate that the *Unified Metric*-based assignment in the IPTQ-ViT pipeline not only reduces theoretical latency but also translates into real efficiency gains on actual hardware.

Method	GPU	DeiT-T (ms)	DeiT-S (ms)	DeiT-B (ms)
Baseline (FP)	3090	3.98	6.58	18.7
I-ViT [14]	2080Ti	1.61	2.97	7.93
I-ViT ^{\$}	3090	1.72 ($\times 2.31$)	2.97 ($\times 2.21$)	6.99 ($\times 2.68$)
Ours	3090	1.73 ($\times 2.30$)	3.46 ($\times 1.90$)	7.11 ($\times 2.63$)

Table 9. End-to-end latency (ms) with batch size 8. All results are on an RTX 3090 GPU except the original I-ViT [14] (2080 Ti).

6. Ablation Study

We evaluate the individual and combined effects of our three proposed methods: (1) *Data-aware Poly-GELU*, (2) *Efficient Bit-Softmax*, and (3) *Unified Metric*. All experiments follow the IPTQ-ViT pipeline (Figure 2) and analyze accuracy changes across different ViTs. Results are summarized in Table 7. First, replacing SQNR with *Unified Metric* (“Legacy/Unified”) consistently improves accuracy over “Legacy/SQNR”, confirming its effectiveness in guiding approximation selection. “Legacy” denotes only using existing approximation functions. Next, expanding the search space with our proposed functions (“Extended {Softmax or GELU}/SQNR”) further boosts accuracy beyond the “Legacy” setting and even outperforms FQ-ViT [19] on most models. Notably, adding only *Efficient Bit-Softmax* yields a +0.47%p gain on DeiT-B over FQ-ViT, showing its standalone benefit. When both Poly-GELU and Bit-Softmax are included (“Extended/SQNR”), performance improves over using either alone, demonstrating their complementary nature. Finally, combining the full search space with *Unified Metric* (“Extended/Unified”) delivers the best results across all models.

7. Acknowledgement

This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA (Defense Acquisition Program Administration)) (No.KRIT-CT-22-040, Heterogeneous Satellite constellation based ISR Research Center, 2025).

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-

2023-00277060, Development of open edge AI SoC hardware and software platform)

8. Conclusion

We presented IPTQ-ViT, the first PTQ framework that enables fully integer-only vision transformers, a goal previously infeasible with conventional PTQ. By introducing tailored approximations for non-linear layers and a unified metric for effective function assignment, IPTQ-ViT delivers accurate and efficient integer-only inference without re-training. Experiments demonstrate that it outperforms prior PTQ baselines, achieves accuracy on par with integer-only QAT methods, and offers strong practicality for deployment with demonstrated latency.

References

- [1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. [6](#)
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. [1](#)
- [3] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. [1](#)
- [4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Y Cao, Y Xiong, X Li, S Sun, W Feng, Z Liu, J Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. arxiv 2019. *arXiv preprint arXiv:1906.07155*, 1906. [6](#)
- [5] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594, 2018. [6](#)
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [6](#)
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [1, 6](#)
- [8] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC, 2022. [1, 2](#)
- [9] Xing Hu, Yuan Cheng, Dawei Yang, Zhihang Yuan, Jiangyong Yu, Chen Xu, and Sifan Zhou. I-lm: Efficient integer-only inference for fully-quantized low-bit large language models. *arXiv preprint arXiv:2405.17849*, 2024. [2](#)
- [10] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. [1, 2](#)
- [11] Gihwan Kim, Jemin Lee, Sihyeong Park, Yongin Kwon, and Hyungshin Kim. Mixed non-linear quantization for vision transformers. In *European Conference on Computer Vision*, pages 97–112. Springer, 2024. [1, 2](#)
- [12] Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pages 5506–5518. PMLR, 2021. [1, 2, 3, 4, 5, 6, 7](#)
- [13] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. [1](#)
- [14] Zhikai Li and Qingyi Gu. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17065–17075, 2023. [1, 2, 3, 5, 6, 7, 8](#)
- [15] Zhikai Li, Liping Ma, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Patch similarity aware data-free quantization for vision transformers. In *European conference on computer vision*, pages 154–170. Springer, 2022. [1, 2, 6, 7, 8](#)
- [16] Zhikai Li, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Psaq-vit v2: Toward accurate and general data-free quantization for vision transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. [2, 7, 8](#)
- [17] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17227–17236, 2023. [1, 2, 6, 7, 8](#)
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. [6](#)
- [19] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021. [1, 2, 3, 5, 7, 8](#)
- [20] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2023. [1, 6, 7](#)
- [21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

- Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 6
- [22] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 105:107281, 2020. 1
- [23] Jiajun Qin, Tianhua Xia, Cheng Tan, Jeff Zhang, and Sai Qian Zhang. Picachu: Plug-in cgra handling upcoming nonlinear operations in llms. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 845–861, 2025. 2
- [24] Akshat Ramachandran, Souvik Kundu, and Tushar Krishna. Clamp-vit: Contrastive data-free learning for adaptive post-training quantization of vits. In *European Conference on Computer Vision*, pages 307–325. Springer, 2024. 1, 2, 6, 7, 8
- [25] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 1
- [26] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 6
- [27] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020. 1, 2
- [28] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pages 11875–11886. PMLR, 2021. 1
- [29] Haoran You, Yipin Guo, Yichao Fu, Wei Zhou, Huihong Shi, Xiaofan Zhang, Souvik Kundu, Amir Yazdanbakhsh, and Yingyan Celine Lin. Shiftaddllm: Accelerating pre-trained llms via post-training multiplication-less reparameterization. *Advances in Neural Information Processing Systems*, 37:24822–24848, 2024. 2
- [30] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, pages 191–207. Springer, 2022. 1, 2, 6, 7
- [31] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo: Transformer-based yolo for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2799–2808, 2021. 1
- [32] Yunshan Zhong, Jiawei Hu, You Huang, Yuxin Zhang, and Rongrong Ji. Erq: Error reduction for post-training quantization of vision transformers. In *Forty-first International Conference on Machine Learning*, 2024. 2
- [33] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. Octr: Octree-based transformer for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5166–5175, 2023. 1
- [34] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1