Quantformer: Learning Extremely Low-precision Vision Transformers

Ziwei Wang, *Student Member, IEEE,* Changyuan Wang, Xiuwei Xu, *Student Member, IEEE,* Jie Zhou, *Senior Member, IEEE,* and Jiwen Lu, *Senior Member, IEEE*

Abstract—In this paper, we propose extremely low-precision vision transformers called Quantformer for efficient inference. Conventional network quantization methods directly quantize weights and activations of fully-connected layers without considering properties of transformer architectures. Quantization sizably deviates the self-attention compared with full-precision counterparts, and the shared quantization strategy for diversely distributed patch features causes severe quantization errors. To address these issues, we enforce the self-attention rank in quantized transformers to mimic that in full-precision counterparts with capacity-aware distribution for information retention, and quantize patch features with group-wise discretization strategy for quantization error minimization. Specifically, we efficiently preserve the self-attention rank consistency by minimizing the distance between the self-attention in quantized and real-valued transformers with adaptive concentration degree, where the optimal concentration degree is selected according to the self-attention entropy for model capacity adaptation. Moreover, we partition patch features in different dimensions with differentiable group assignment, so that features in different groups leverage various discretization strategies with minimal rounding and clipping errors. Experimental results show that our Quantformer outperforms the state-of-the-art network quantization methods by a sizable margin across various vision transformer architectures in image classification and object detection. We also integrate our Quantformer with mixed-precision quantization to further enhance the performance of the vanilla models.

Index Terms—Vision transformers, network quantization, self-attention rank consistency, group-wise discretization, differentiable search

1 Introduction

Transformer [47], [1], [14] has revolutionized natural language processing (NLP) because of the flexibility in modeling long-range dependencies. Inspired by the great success, vision transformers were widely studied to achieve promising performance in image classification [15], [46], [35], object detection [4], [12], semantic segmentation [63], [38] and many others. However, the heavy storage and computational cost obstructs the vision transformers from being deployed in realistic applications with limited resources such as mobile phones and robots. Hence, it is desirable to design vision transformers with fewer parameters and more lightweight architectures.

Recently, several compression techniques for convolutional neural networks have been proposed including pruning [45], [23], [21], quantization [24], [9], [61], low-rank decomposition [26], [40], [25] and efficient architecture design [28], [55], [43]. Among these methods, quantization leads to extremely high compression ratio for memory saving and computation acceleration due to the sizable bitwidth reduction and low-precision multiply-accumulate operations (MACs). However, directly quantizing the weights and activations in vision transformers ignores the properties of transformer architectures. First, low-precision quantization deviates the self-attention in quantized transformers from that in the full-precision counterparts, which leads to inaccurate patch dependencies for subsequent feature extraction. Second, conventional network quantization methods utilize the discretization function with the same thresholds and rounding points for patch features from different dimensions. Since the real-valued patch features distribute diversely across various dimensions, the shared quantization policies result in sizable rounding and clipping errors respectively for narrowly and widely distributed features. Therefore, the deviated self-attention and the considerable quantization loss both significantly degrade the performance of low-precision vision transformers compared with their full-precision counterparts.

In this paper, we present a Quantformer approach to learn extremely low-precision vision transformers for efficient inference. Unlike existing methods which directly quantize fully-connected layers without considering the properties of transformer architectures, we enforce the self-attention rank in quantized transformers to mimic that in full-precision counterparts with capacity-aware distribution for information retention. We also discretize the patch features in different dimensions with group-wise quantization, where patch features in similar distribution are discretized with the shared quantization strategy. Therefore, the self-attention consistency between quantized and real-valued transformers is preserved, and the quantization errors resulted from diversely distributed patch features are alleviated with negligible computation overhead. Figure 1 demonstrates the difference between Quantformer and the conventional network quantization methods. More specifically, we minimize the rank difference between the self-attention in quantized and full-precision transformers for

- *Ziwei Wang, Changyuan Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: wang-zw18@mails.tsinghua.edu.cn, 201811210202@mail.bnu.edu.cn, xxw21@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn, lujiwen@tsinghua.edu.cn.*
- *Corresponding author: Jiwen Lu*
- *Code: https://github.com/ZiweiWangTHU/Quantformer.git.*

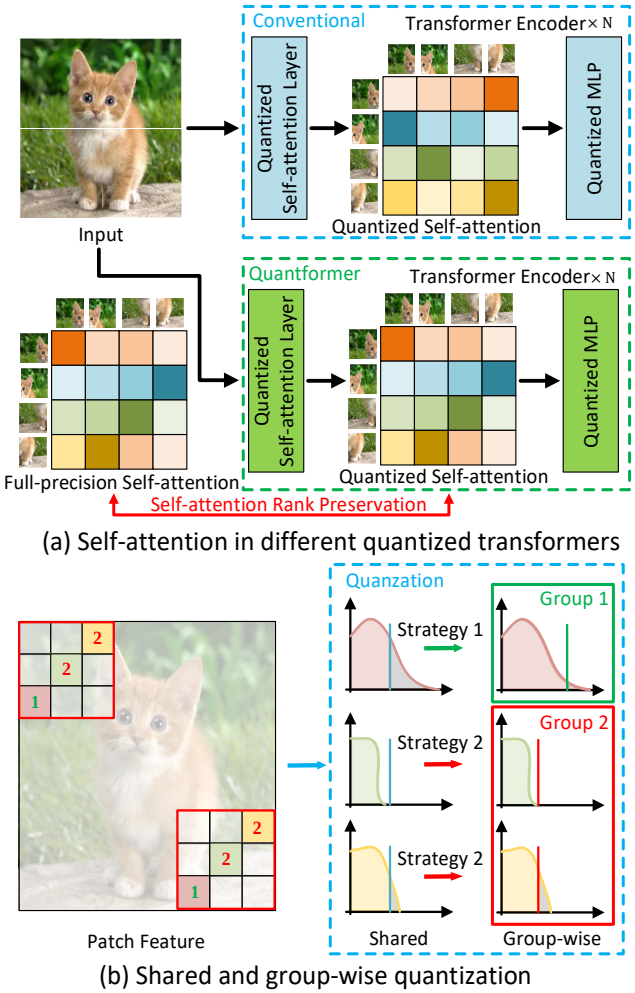


Figure 1. The difference between Quantformer and the conventional network quantization methods. (a) The self-attention in the conventional quantized vision transformers and our Quantformer, where the full-precision self-attention is also demonstrated for reference. The color darkness in the same row depicts the self-attention ranks across patches, where dark colors stand for ranks in top positions. Conventional quantization methods deviate the self-attention rank from that in full-precision transformers, and leads to inaccurate patch dependencies for feature extraction. On the contrary, our Quantformer preserves the self-attention rank consistency via the presented capacity-aware self-attention imitation. (b) The difference between the shared and group-wise quantization strategies, where the former causes more significant quantization loss including rounding and clipping errors for the diversely distributed patch features. Our Quantformer partitions the patch features in different dimensions according to the feature distribution, where quantization strategies with various thresholds and rounding points are employed for features in different groups with alleviated quantization errors. Different colors in the patch feature demonstrate features in different dimensions, and those belonging to different groups are denoted by various indexes.

consistency preservation. To solve the NP-hard problem of rank consistency regularization, we alternatively minimize the distance between the self-attention in quantized transformers and the full-precision counterparts with adaptive concentration degree. The concentration degree is optimally selected based on the self-attention entropy, so that the self-attention rank consistency is preserved without the mismatch between the self-attention distribution and the limited capacity of the quantized vision transformers. We efficiently partition patch feature dimensions by differentiable group assignment, where features in different groups utilize various quantization strategies. Patch features represent the linear layer output for each patch, whose dimension is the number of elements. Features in different dimensions represent various elements within one patch as shown by different colors of patch features in Figure 1(b). Those belonging to different groups are denoted by various indexes in patch features of Figure 1(b). As a result, the optimal thresholds and quantization points are selected for full-precision patch features with alleviated clipping and rounding errors. Compared with the state-of-the-art network quantization methods, experiments on ImageNet [13] for image classification and COCO [31] for object detection show that our Quantformer achieves higher performance with negligible computational overhead across various vision transformer architectures including DeiT [46] and Swin Transformer [35]. Moreover, the techniques presented in Quantformer can be integrated with mixed-precision quantization to further enhance the vanilla models.

The contributions of this work are summarized as follows:

- To the best of our knowledge, we propose the first low-precision quantized vision transformers whose weights and activations are represented in less than four bits.
- We propose the capacity-aware self-attention imitation to efficiently preserve the consistency between the self-attention rank in quantized and full-precision transformers, so that the long-range dependency is accurately mined without capacity insufficiency.
- We present the group-wise quantization to discrete patch features in different dimensions with the optimal thresholds and quantization points, where the clipping and rounding errors in quantized transformers are alleviated with negligible computation overhead.
- We conduct extensive experiments on image classification and object detection, and the results consistently show that our Quantformer outperforms the state-of-the-art network quantization methods by a sizable margin. We also integrate the proposed techniques to mixed-precision quantization in order to further enhance the vanilla models.

2 Related Work

In this section, we briefly review three related topics including 1) network quantization, 2) efficient vision transformers and 3) differentiable search.

2.1 Network Quantization

In order to reduce the storage and computational complexity, network quantization has aroused broad interest in computer vision to enable the deployment in mobile devices. Existing network quantization methods can be categorized into two types: binary networks and models in multiple bits. For the former, weights and activations are represented by binary numbers [52], [51], where the multiply-accumulate (MAC) operations in full-precision networks are substituted by xnor-bitcount. Hubara *et al.* [24] first presented binary neural networks which were optimized by straight-through

estimators (STE), and Rastegari *et al.* [42] scaled the binarized weights and activations for quantization error minimization. Liu *et al.* [36] added shortcut connections between adjacent layers to enhance the representation ability caused by binarization. Wang *et al.* [52] mined the channel-wise interactions in convolutional neural networks to eliminate the inconsistent signs between the quantized and full-precision feature maps. Because the huge performance gap between the binary networks and their full-precision counterparts limits the model practicability, networks in multiple bits are proposed to achieve better performance. Choi *et al.* [9] learned the activation clipping thresholds for optimal quantization scale selection, and Zhang *et al.* [61] further optimized the quantizer basis and encoding to minimize the expected quantization errors. Lee *et al.* [28] scaled each gradient element computed by STE to match the direction of objective and the gradient without discretization errors. Aiming to achieve better accuracy-complexity trade-offs, mixed-precision quantization [48], [3], [53] selects the optimal bitwidth for each network component to adaptively remove the network redundancy. To alleviate quantization errors, Zhao *et al.* [62] duplicated and halved outlier channels to move the outliers towards distribution center with functional identity, and Liu *et al.* [33] approximated the full-precision weight tensors by quantized counterparts with the optimal basis number. Nevertheless, directly quantizing weights and activations of vision transformers with conventional network quantization methods fails to consider the properties of transformer architectures. The self-attention in quantized transformers is deviated from that in the full-precision counterparts, and the quantization policy with shared thresholds and discretization points for patch features that distribute diversely across channels causes sizable clipping and rounding errors. Therefore, the low-precision vision transformers underperform the full-precision counterparts significantly.

2.2 Efficient Vision Transformers

Inspired by the great success of transformers [47] in the NLP domain [14], [1], [34], vision transformers [15], [46], [35] were proposed to mine global dependencies with self-attention mechanism. Because of the extremely heavy storage and computational cost, efficient vision transformers are desirable for applying the large pretrained models in realistic applications. To address this, efficient architecture design [17], [5], [58], architecture search [6], [8], sparse attention [7], [11], [57], [50], [41] and network quantization are presented to reduce the storage and computational cost. To construct lightweight architectures of vision transformers, Fan *et al.* [17] utilized the multiscale feature pyramid fusion to reduce the spatial resolution of top layers. Chen *et al.* [5] observed that self-attention in consecutive layers was similar, so that they employed the shared self-attention in neighboring layers to decrease the computational cost. For architecture search, Chen *et al.* [6] acquired the optimal global and local dependency learning modules hierarchically by evolutionary algorithm to achieve better accuracy-complexity trade-offs. Sparse attention aims to enhance the efficiency of dense self-attention by removing redundant patches. Chu *et al.* [11] proposed the spatially separable self-attention that alternatively mined the global and local dependencies with significant computational complexity decrease. Moreover, Rao *et al.* [41] pruned unimportant tokens according to the input to achieve fine-grained resource assignment for samples in various hardness. Since quantization can achieve extremely high compression ratio, quantized transformers [60], [44], [27] were presented in NLP domain. However, these methods are incompatible with vision tasks and result in obvious performance degradation, because the model capacity changes significantly with different bitwidths and feature distribution across channels varies obviously in quantized vision transformers. To quantize vision transformers, Liu *et al.* [37] and Yuan *et al.* [59] designed post-training quantization framework by rescaling the latent weights of quantized layers with a small calibration set. Although [37] presented ranking-aware quantization for self-attention in vision transformer discretization, the applied hinge loss fails to consider the model capacity variance and leads to capacity insufficiency for quantized vision transformers. Moreover, the high complexity of hinge loss computation causes extremely high training cost. Moreover, sizable performance degradation is observed in the post-training quantization framework for extremely low-precision vision transformers, which prevents them to be deployed in mobile devices with limited resources.

2.3 Differentiable Search

Aiming at efficiently finding the optimal solution in large search space, differentiable search has been widely adopted in many search problems including network architecture search [32], [10], [30], mixed-precision quantization [3], [53], [49] and feature learning [20], [54]. Differentiable search methods usually regard each choice in the search space as a branch of the superstructure, and update the component importance via gradient descent during the search phase. When the search stage completes, the superstructure is discretized by preserving the components with the largest branch importance weight to obtain the optimal solution. For network architecture search, Liu *et al.* [32] relaxed the architecture selection to be continuous where the branch importance and supernet weights were trained jointly, and they also leveraged the difference approximation method to effectively solve the bi-level optimization. Moreover, Chu *et al.* [10] presented the competitive mechanism to eliminate the unfair advantages of skipping connections in differentiable architecture search, and also added the zero-one loss in the overall objective to minimize the discretization errors of architecture discretization. For mixed-precision quantization, Cai *et al.* [3] assigned different bitwidths for weights and activations in various branches of the supernet, and imposed the complexity constraint that increased the importance of low-precision branches for model compression. Wang *et al.* [53] observed that locating the attribution correctly was a general ability for accurate visual analysis despite of the model bitwidths, and they preserved the attribution rank consistency between the quantized and full-precision networks during the differentiable bitwidth search. For feature aggregation, Guan *et al.* [20] presented the bridge loss for feature weight optimization, and enhanced the knowledge distillation via the bi-directional

path between teacher and student models. To acquire the optimal partition of patch features in different channels, we generalized the differentiable search strategies to partition assignment of group-wise quantization.

3 Approach

In this section, we first briefly review network quantization that leads to extremely high compression ratio for memory saving and computation acceleration. Then we introduce the self-attention rank preservation for quantized transformers, where the capacity-aware self-attention imitation is presented for efficient implementation. Finally, we demonstrate the group-wise quantization strategy for patch feature discretization, and the differentiable search methods are employed for efficient partition assignment with minimal quantization errors.

3.1 Network Quantization

Network quantization decreases the bitwidths of weights and activations to save storage cost and accelerate inference. Let us denote $\boldsymbol{X}^r$ as the real-valued matrix in neural networks including weights and activations, and the k-bit rounding function Q_k discretizes elements in $\boldsymbol{X}^r$ to the nearest quantization point to form the quantized matrix $\boldsymbol{X}^q$:

$$x_i^q = Q_k(x_i^r) \in \{q_0, q_1, ..., q_{2^k-1}\} \tag{1}$$

where x_i^r and x_i^q represent the i_{th} element in the real-valued and quantized matrix respectively, and q_j means the j_{th} quantization point. The uniform quantization scheme [48], [29] is considered due to the highly efficient implementation on the hardware. The distance between any adjacent quantization points is equal in uniform quantization, which is denoted as Δ_k for k-bit quantization. Given the upper and lower bounds of the quantization range u and l, the quantization interval Δ_k is defined as $\frac{u-l}{2^k-1}$. The quantized value is determined for each real-valued element by assigning the first and last rounding point to the minimum and maximum of the quantization range respectively. The real-valued elements larger than u or smaller than l are set to u and l respectively so that all elements are clipped into the quantization range. The quantized elements in uniform rounding can be depicted as follows:

$$x_i^q = Q_k(x_i^r) = \Delta_k \cdot (x_i^n - b) \tag{2}$$

where $x_i^n = [\frac{\hat{x_i^r} - l}{\Delta_k}]$ stands for the rounding index of the real-valued element x_i^r, and b demonstrates the quantization bias index. In the definition of rounding index, $\hat{x_i^r}$ represents the clipped elements of x_i^r, and $[x]$ illustrates the nearest integer around x. We replace the multiply-accumulate (MAC) operations in full-precision neural networks with efficient integer arithmetic in quantized models:

$$\boldsymbol{W}_q \boldsymbol{A}_q = \mathcal{Q}_k(\boldsymbol{W}_r)\mathcal{Q}_k(\boldsymbol{A}_r) = \Delta_k^{\boldsymbol{W}} \Delta_k^{\boldsymbol{A}} (\boldsymbol{W}_N \boldsymbol{A}_N) \tag{3}$$

where $\boldsymbol{W}_q$ and $\boldsymbol{A}_q$ are the quantized weights and activations, and $\boldsymbol{W}_r$ and $\boldsymbol{A}_r$ mean their full-precision counterparts. The quantization bias index term is omitted for clarity. The quantization function $\mathcal{Q}_k(\boldsymbol{x})$ indicates that we leverage the quantization strategy shown in (2) for each element in matrix $\boldsymbol{x}$. The distance between adjacent quantization points in k-bit quantization for weights and activations are represented by $\Delta_k^{\boldsymbol{W}}$ and $\Delta_k^{\boldsymbol{A}}$ respectively, and $\boldsymbol{W}_N$ and $\boldsymbol{A}_N$ demonstrate the rounding index matrix for weights and activations. We only quantize the linear layers of vision transformers in low-precision, and assign the bitwidth of query, key, value tensors to eight in order to avoid severe performance drop. Following [37], [56], [59], we apply float layernorm and softmax layers in vision transformers as they are very sensitive to quantization.

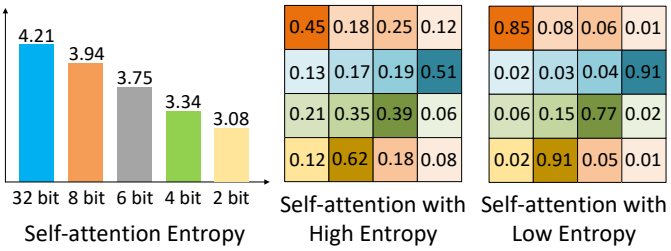


Figure 2. The left figure shows the Shannon Entropy of the self-attention in DeiT-T with different bitwidths on ImageNet, where the converged models are leveraged to evaluate the average entropy across various rows, layers and samples. The middle and right figures demonstrate the example of self-attention with different entropy, while the self-attention rank keeps the same. The bitwidth reduction causes the decreased self-attention entropy, because the degraded model capacity enforces the quantized transformers to focus on the most important patches without self-attention divergence. In order to accurately mine the long-range dependencies among patches without capacity insufficiency, we preserve the self-attention rank consistency instead of minimizing the self-attention distance directly (best viewed in color).

However, directly quantizing weights and activations in vision transformers without considering the architectures causes two issues. First, low-precision quantization deviates the self-attention in quantized transformers from that in the full-precision counterparts, which leads to inaccurate patch dependencies for feature extraction. Second, leveraging the discretization strategy with the same thresholds and quantization points for diverse patch features causes significant rounding and clipping errors. As a result, the inconsistent self-attention and discretization errors both result in sizable performance degradation of quantized vision transformers.

3.2 Preserving Self-attention Rank Consistency with Capacity-aware Distribution

Low-precision quantization deviates the self-attention in quantized vision transformers from that in the full-precision counterparts, which results in inaccurate focus for patch dependency mining during feature extraction. Therefore, the information retention is harmed in quantized vision transformers with significant performance degradation. In order to maintain the accurate focus for long-range dependency learning, we preserve the self-attention rank consistency between quantized and full-precision vision transformers. In this section, we first introduce the capacity-aware self-attention rank preservation for quantized vision transformers, and then detail the efficient implementations via capacity-aware self-attention imitation.

3.2.1 *Capacity-aware Self-attention Rank Preservation*

We first revisit the attention mechanism in vision transformers. The flattened feature maps in the l_{th} layer are denoted

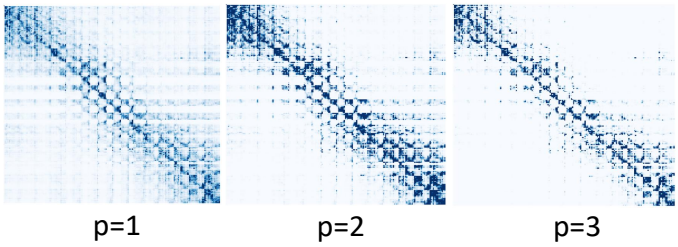


Figure 3. An example showing the l_p norm of self-attention with different p, where the rank remains unchanged and the concentration degree varies. Larger p leads to more concentrated self-attention and vice versa (best viewed in color).

as $\boldsymbol{X}^l$, and the corresponding self-attention $\boldsymbol{A}_l$ for the l_{th} layer can be obtained via the query $\boldsymbol{Q}_l$ and key $\boldsymbol{K}_l$:

$$\boldsymbol{A}_l = \boldsymbol{Q}_l \boldsymbol{K}_l = \boldsymbol{X}_l \boldsymbol{W}_l^Q \boldsymbol{W}_l^{K^T} \boldsymbol{X}_l^T \tag{4}$$

where $\boldsymbol{W}_l^Q$ and $\boldsymbol{W}_l^K$ are the query and key computation matrix in the l_{th} layer. The output of multi-head self-attention (MHSA) module can be written in the following:

$$MHSA(\boldsymbol{X}^l) = \sigma(\frac{1}{\sqrt{d_l}} \boldsymbol{A}_l) \boldsymbol{X}^l \boldsymbol{W}_l^V \boldsymbol{W}_l^O \tag{5}$$

where $\sigma(\cdot)$ means the softmax function and d_l is the dimension of the patch features. $\boldsymbol{W}_l^V$ and $\boldsymbol{W}_l^O$ respectively represent the value and output computation matrix. With the layernorm layers and identity shortcut, the multi-head self-attention and multi-layer perceptron modules are stacked iteratively for feature extraction.

Self-attention is crucial in vision transformers as it measures the patch importance in the forward pass of MHSA modules. Quantizing weights and activations of fully-connected layers destroys the informative self-attention learned in full-precision vision transformers, and the patch importance order is significantly changed in the forward pass of feature extraction. Therefore, we aim to preserve the self-attention consistency between the quantized vision transformers and their full-precision counterparts in order to maintain the feature informativeness. Figure 2 shows the average Shannon Entropy of the self-attention in the DeiT-T architecture with different bitwidths. The vision transformer in lower precision obtains self-attention in smaller entropy due to the limited carried information, which indicates more concentrated self-attention distribution. As the network capacity between the quantized and full-precision models is huge, directly minimizing the distance between the self-attention in quantized transformers and that in the full-precision counterparts fails to remove the redundant information in the compressed model, while causes capacity insufficiency with degraded performance. Therefore, we instead preserve the self-attention rank consistency between the quantized and full-precision transformers with capacity-aware distribution, which enables the quantized transformers to focus on important patches while adaptively adjust the self-attention distribution without capacity insufficiency. Figure 2 also depicts intuitive examples of self-attention with the same rank and different entropy, where self-attention in low entropy with concentrated distribution is acquired for models in inferior capacity. We define the self-attention rank consistency loss J_{src} for model training in the following form:

$$\begin{aligned} J_{src} &= \sum_l \sum_m \sum_n ||r(A_{q,mn}^l) - r(A_{r,mn}^l)||^2 \\ s.t. \quad & \sum_n A_{q,mn}^l \log A_{q,mn}^l = C_l \sum_n A_{r,mn}^l \log A_{r,mn}^l \end{aligned} \tag{6}$$

where $A_{q,mn}^l$ represents the element in the m_{th} row and n_{th} column of the self-attention matrix in the l_{th} layer of quantized vision transformers, and $A_{r,mn}^l$ demonstrates the corresponding element in the full-precision models. $r(x)$ depicts the rank of the element x in self-attention, which equals to k if the element x is the k_{th} largest among all others in the same row. The layer-wise hyperparameter C_l reveals the capacity difference between the quantized and full-precision vision transformers for the self-attention in the l_{th} layer. By enforcing the self-attention rank in the quantized models to mimic that in the full-precision counterparts, the important patches are correctly focused without capacity insufficiency so that the long-range dependencies among patches are accurately mined for informative feature extraction.

3.2.2 Capacity-aware Self-attention Imitation

Since directly minimizing the inconsistency between self-attention ranks in quantized and full-precision transformers is NP-hard, we present capacity-aware distribution self-attention imitation to efficiently preserve the self-attention rank consistency. The range of self-attention is set to $[0, 1]$ where the element summation over each row equals to one. Therefore, the l_p norm of self-attention can adjust the self-attention divergence without changing the rank by modifying the value of p. Figure 3 shows an example of the l_p norm of self-attention with different p, where the rank remains the same and the concentration degree for self-attention varies. In order to efficiently preserve the self-attention rank consistency between the quantized and full-precision transformers, we minimize the difference between the self-attention in quantized transformers and the l_p norm of that in full-precision models. The self-attention rank consistency loss can be rewritten as follows for efficient implementation during the optimization process:

$$J_{src} = \sum_l \sum_m \sum_n ||A_{q,mn}^l - \frac{(A_{r,mn}^l)^{p_l}}{\sum_n (A_{r,mn}^l)^{p_l}}||^2 \tag{7}$$

where p_l means the hyperparameter p of the l_p norm in the l_{th} layer, and large p_l leads to concentrated self-attention and vice versa. As the representational capacity of each layer in the vision transformer varies, the hyperparameter p_l in the self-attention rank consistency loss should be assigned with larger value for quantized layers with lower capacity. The Shannon Entropy of the self-attention depicts the amount of carried information that also reveals the network capacity. Denoting the layer-wise Shannon Entropy of the self-attention in the l_{th} quantized and full-precision transformer layers as E_q^l and E_r^l respectively, the hyperparameter p_l can be decided as follows:

$$p_l = \frac{E_r^l}{E_q^l} = \frac{\sum_m \sum_n A_{r,mn}^l \log A_{r,mn}^l}{\sum_m \sum_n A_{q,mn}^l \log A_{q,mn}^l} \tag{8}$$

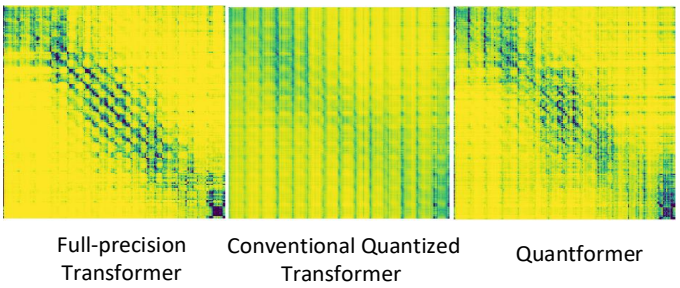


Figure 4. The self-attention visualization in the top layer of the full-precision transformer, the conventional quantized transformer and our Quantformer, where the DeiT-T architecture with 4-bit weights and activations is applied. The darker color represents larger value and vice versa. Conventional quantization methods deviate the self-attention rank from that in full-precision transformers, while our Quantformer preserves the self-attention rank consistency with capacity-aware distribution.

The hyperparameter p_l becomes large when the self-attention entropy in the l_{th} layer of the quantized models is much smaller than that in the full-precision transformers. The strong correlation between (6) and (7) is verified in Appendix C via theoretical proof, model statistics and ablation studies. Specifically, p_l is assigned to one for equal E_q^l and E_r^l, which indicates the comparable model capacity for the l_{th} layer in quantized and full-precision transformers. In this case, directly minimizing the distance between the self-attention in the l_{th} layer of quantized and full-precision models results in accurate long-range dependency mining. We feed forward the input samples to pre-trained full-precision vision transformers during the training process to obtain $A_{r,mn}^l$ across all layers. By optimizing the self-attention rank consistency loss with capacity-aware distribution, the quantized transformers correctly focus on important patch relation without capacity insufficiency. Figure 4 visualizes the self-attention in full-precision vision transformer, conventional quantized vision transformer and our Quantformer, where our Quantformer keeps the similar self-attention rank with that in the full-precision counterparts. Meanwhile, the concentration degree of the self-attention in our Quantformer is adaptively adjusted according to the network capacity in order to prevent capacity insufficiency.

3.3 Group-wise Quantization for Patch Features

Existing network quantization methods directly discretize the weights and activations in the same strategy, which leverages shared clipping thresholds and quantization points for all elements in a layer. Although the shared quantization strategy is efficient on hardware, it significantly degrades the performance of vision transformers due to the large clipping and rounding errors for diversely distributed patch features. In order to alleviate the quantization loss in vision transformers, we present the group-wise quantization where patch features in the same partition utilize the quantization strategy with shared thresholds and quantization points. We first describe the details of group-wise quantization, and then demonstrate the differentiable search framework for optimal group assignment of patch features in different dimensions.

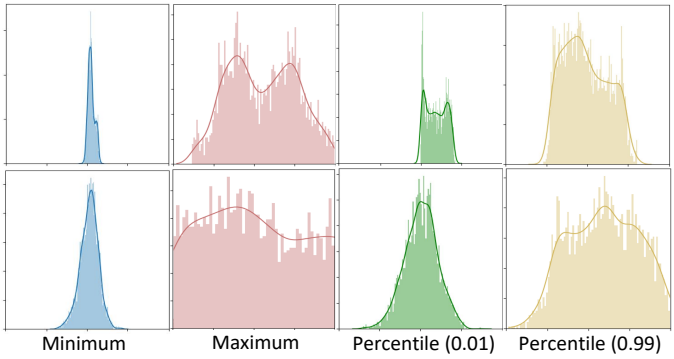


Figure 5. The minimum, maximum, percentile at 0.01 and percentile at 0.99 for patch features in different dimensions in top layers (top row) and bottom layers (bottom row) of vision transformers, where the architecture of 4-bit DeiT-T is applied. The horizontal axis represents the value for patch features in different dimensions and the vertical axis means the corresponding frequencies. Since the range of patch features in different dimensions varies significantly regardless of layers, leveraging the shared discretization strategy for them yields sizable quantization errors.

Table 1
The number of parameters and bit-operations (BOPs) of vision transformers for image classification on ImageNet with layer-wise quantization and channel-wise quantization respectively. Param. depicts the number of network parameters (M) and BOPs is the bit-operations (G).

Quantization	DeiT-T		DeiT-S		DeiT-B	
	Param.	BOPs	Param.	BOPs	Param.	BOPs
Layer-wise	0.69	34.9	2.89	103.2	11.04	340.3
Channel-wise	0.84	109.8	3.04	404.4	11.19	1545.0
Increase	21.7%	214.6%	5.2%	291.9%	0.8%	354.0%

3.3.1 *Group-wise Quantization*

Conventional quantization approaches [48], [9] utilize sharable parameters for each layer as the discretization range thresholds. However, the diversely distributed patch features across different dimensions disable the vision transformer to learn compatible thresholds with acceptable quantization errors. Figure 5 illustrates the minimum, maximum, percentile at 0.01 and percentile at 0.99 of real-valued elements in patch features across all dimensions, which distribute very diversely regardless of the layer indexes. Therefore, the shared quantization strategy across feature dimensions results in high quantization errors, because small quantization range causes sizable clipping loss and the large one leads to significant rounding errors.

Inspired by channel-wise quantization [39], we adopt different discretization strategies for features in various dimensions to alleviate quantization errors. However, employing an unique quantization strategy for features in each dimension leads to heavy storage and computation overhead because of the significantly increased rounding functions. Table 1 demonstrates the number of parameters and bit-operations (BOPs) of vision transformers for image classification on ImageNet with layer-wise quantization and channel-wise quantization respectively, where channel-wise quantization increases the storage and computational complexity significantly. Hence, directly leveraging channel-wise quantization for vision transformers to reduce the quantization errors is not feasible. Meanwhile, quantizing

features whose optimal quantization range is similar with the same discretization strategy can efficiently achieve better trade-offs between the quantization errors and model complexity. We present group-wise quantization for patch features in different dimensions, so that the quantization errors for the diversely distributed elements are alleviated without sizably additional storage and computational complexity. We also demonstrate the accuracy-efficiency trade-offs of layer-wise, channel-wise and group-wise quantization in Appendix E.

Let us assume features from different dimensions are divided into C groups, and the activation quantization strategy can be written as follows by modifying (2) into a group-wise manner:

$$z_i^q = Q_k^{c(i)}(z_i^r) = \Delta_k^{c(i)}([\frac{\hat{z_i^r} - l_{c(i)}}{\Delta_k^{c(i)}}] - b_{c(i)}) \tag{9}$$

where z_i^q and $c(i)$ respectively represent the quantized feature elements and the group assignments for the i_{th} dimension, and $Q_k^{c(i)}$ demonstrates the k_{th}-bit rounding function of the $c(i)_{th}$ group. Meanwhile, $\hat{z_r^i}$ is the clipped full-precision feature element with the original real-valued element z_r^i in the i_{th} dimension, where the original element is clamped into the activation range with the upper and lower bounds $u_{c(i)}$ and $l_{c(i)}$. Moreover, $\Delta_k^{c(i)}$ and $b_{c(i)}$ respectively depict the distance between adjacent k-bit quantization points and the quantization bias index for the elements assigned to the $c(i)_{th}$ group. Figure 6 demonstrates an example of shared and group-wise quantization policy on feature distribution for different dimensions, where the latter selects the optimal quantization range for features in different dimensions to alleviate quantization errors.

As the initialization of quantization range thresholds influences the model performance, we adopt the percentile of feature distribution as the initial quantization range thresholds. We randomly partition the patch features in different dimensions into C groups for quantization range initialization, where each group contains equal numbers of elements. The lower bound l_c and the upper bound u_c for the c_{th} group is defined as follows:

$$\begin{aligned} l_c &= \{z_i^r \in Z_c | r_c(z_i^r) = Np_0\} \\ u_c &= \{z_i^r \in Z_c | r_c(z_i^r) = N(1-p_0)\} \end{aligned} \tag{10}$$

where Z_c means all feature elements assigned to the c_{th} group and $r_c(\cdot)$ is the ranking function in Z_c. Meanwhile, N stands for the number of feature elements in each group and $p_0 \in [0.5, 1]$ is the hyperparameter that indicates the percentile. We select the element that ranks among the p_0 and $1-p_0$ percentile of feature elements in the c_{th} group as the initialized quantization range, so that the outliers with extreme values that result in large quantization errors can be clipped.

3.3.2 Differentiable Search for Group Assignment

Properly partitioning features in different dimensions is critical for group-wise quantization, as discretizing elements in similar distribution with the same quantization strategy can reduce quantization errors without heavy storage and computational overhead. Nevertheless, the group assignment for patch features across different dimensions faces two challenges with significantly increased search cost. First, enumerating all assignment permutation to acquire the optimal solution is NP-hard. Second, the feature distribution changes during the training process, which requires the group assignment to be updated dynamically. In order to efficiently investigate the distribution similarity among features in various dimensions for optimal group assignment, we employ the differentiable search framework [32], [3] to partition the features in different dimensions. We discretize each feature element by C quantization functions with different thresholds and rounding points in parallel, where C represents the number of partitions in group-wise quantization. The quantized counterparts from different quantization functions are summed with various importance weights to form the intermediate activations. Figure 7 depicts the pipeline of the differentiable search framework for optimal group assignment acquisition. The feed-forward propagation for each layer is written as follows:

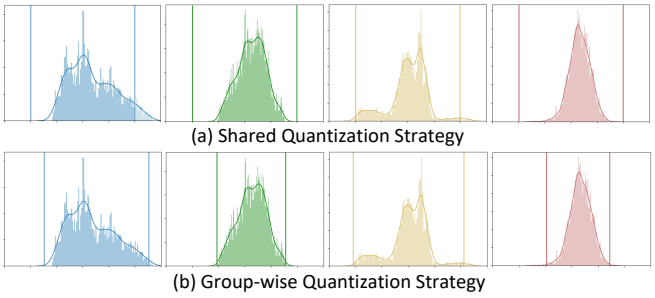


Figure 6. The feature distribution and the quantization thresholds of (a) the shared quantization strategy and (b) the group-wise quantization strategy for four different dimensions in a layer, where each column demonstrates the feature element statistics of one dimension. The 4-bit DeiT-T model is leveraged for the visualization. The horizontal axis stands for the value of the feature elements, and the vertical axis means the corresponding frequencies. The vertical lines demonstrate the upper and lower bound of the quantization range. The shared quantization strategy leads to large rounding errors for features in concentrated distribution and causes high clipping errors for those in divergent distribution. On the contrary, the group-wise quantization selects the optimal quantization range for features in each dimension according to the feature distribution with negligible computational and storage cost, where the rounding and clipping errors are both alleviated.

$$z_i^q = \sum_{c=1}^{C} \pi_c Q_k(z_i^r) = \sum_{c=1}^{C} \pi_c \Delta_k^{c(i)} \cdot [\frac{\hat{z_i^r} - l_{c(i)}}{\Delta_k^{c(i)}}] \tag{11}$$

where π_c represents the importance weight of the c_{th} quantization function. Since discretizing the group assignment from the continuous space usually causes non-negligible discrepancy, we aim to minimize the entropy of the branch importance weights in group assignment with the discrepancy minimization loss:

$$J_{dm} = -\frac{1}{C} \sum_{c=1}^{C} \pi_c \log \pi_c \tag{12}$$

The discrepancy minimization loss enforces the branch importance weights to approach zero or one, so that the feature discrepancy caused by the discrete group assignment after search can be alleviated.

We jointly optimize the network parameters, the quantization thresholds and the importance weights during the

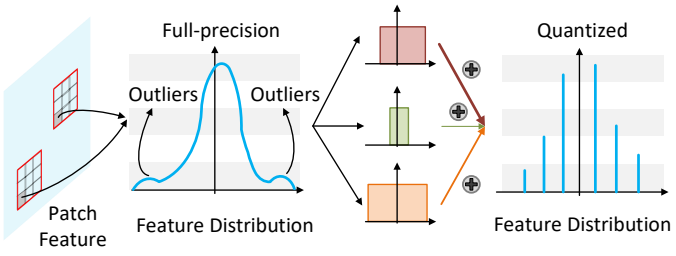
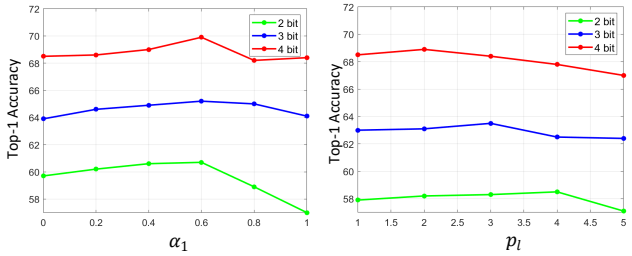


Figure 7. The differentiable search framework for group assignment of patch features in various dimensions. Patch features in each dimension are discretized by several quantization strategies with different thresholds and rounding points in parallel, and the quantized features are summed with various importance weights to form the output. By jointly optimizing the network parameters, the importance weights and the quantization thresholds, the quantization strategy with the largest importance weight after optimization is selected.



(a) (b)

Figure 8. (a) The top-1 classification accuracy with varied hyperparameter α_1 in the overall objective for quantized DeiT-T in different bitwidths. (b) The performance of Quantformer with fixed p_l assignment in the self-attention consistency loss, where p_l was set to 1, 2, 3, 4 and 5 respectively.

differentiable search process until convergence or achieving the maximum iteration steps, where the overall learning objective can be written as follows:

$$J = J_{task} + \alpha_1 J_{src} + \alpha_2 J_{dm} \tag{13}$$

where J_{task} means the task loss, and α_1 and α_2 stand for the hyperparameters that control the importance of the self-attention rank consistency loss and discrepancy minimization loss. When the optimization completes, we discretize elements in different dimensions with the quantization function that is evaluated with the highest importance weights. Finally, we finetune the quantized transformers with the optimal feature partitions in group-wise quantization via the objective only composed of the task loss and the self-attention consistency loss.

4 Experiments

In this section, we conducted extensive experiments to evaluate our methods on ImageNet for image classification and on COCO for object detection. We first briefly introduce the datasets and the implementation details, and then verify the effectiveness of the presented self-attention rank preservation for information retention and group-wise quantization for discretization error alleviation via ablation study. Finally, we compare our Quantformer with the state-of-the-art network quantization methods on vision transformers to show the superiority.

4.1 Datasets and Implementation Details

We introduce the datasets that we applied and the data preprocessing techniques in the following:

ImageNet: ImageNet (ILSVRC2012) contains approximately 1.2 million and 50k images from 1000 classes for training and validation. During the training stage, we cropped a 224×224 random region from the resized image whose shorter side was 256 in the forward pass. Meanwhile, we applied the 224×224 center crop in inference. Moreover, we scaled and biased all pixels into the range $[0, 1]$. We used the top-1 and top-5 classification accuracies as the evaluation metric, and leveraged the number of parameters and BOPs as the storage and computational cost respectively.

Table 2
The performance w.r.t. different value assignment strategies for dynamic p_l in the self-attention rank consistency loss. p_l is ensured to be one when the entropy of self-attention in quantized and full-precision transformers is equal.

SRC Loss	W/A-bit	Top-1	Top-5
E_r/E_q	2/2	60.7	84.0
	3/3	65.2	87.0
	4/4	69.9	89.7
$\sqrt{E_r/E_q}$	2/2	59.5	83.1
	3/3	64.0	86.6
	4/4	69.1	89.5
$\exp(E_r/E_q)$	2/2	58.7	82.8
	3/3	64.3	86.6
	4/4	69.0	89.2

COCO: The images in the COCO dataset were collected from 80 different categories, and our experiments were conducted on COCO 2017. We trained our model with 118k images from the training set and tested our Quantformer on the test-dev set [35] including 20k images. Following the standard COCO evaluation metric [31], we utilize the mean average precision (mAP) for IoU $\in [0.5 : 0.05 : 0.95]$ as the evaluation metric. We also report average precision with the IOU threshold 50% and 75% represented as AP_{50} and AP_{75} respectively. We followed the experimental settings in [35] including multi-scale training, learning rate schedule and soft NMS.

We evaluated Quantformer with the architectures of DeiT [46] and Swin Transformer [35], where the models with different sizes were utilized. For object detection, we adopted the Mask R-CNN framework [22] and Cascade R-CNN pipeline [2]. The bitwidths for quantized linear layers could be set to 2, 3, and 4 respectively to achieve various accuracy-complexity trade-offs. The officially released weights of full-precision networks for DeiT and Swin Transformer were adopted as the pre-trained models for quantized vision transformer learning. We leverage uniform quantization with learnable upper and lower bounds for weights and activations. The number of partitions in group-wise quantization was set to eight in most experiments. We randomly partitioned the patch features equally in different dimensions into eight groups for quantization range initialization, and the quantization range for each group was initialized according to the percentile parameter p_0 which was set to 0.99. During the differentiable search for group assignment of patch features in different dimensions,

Table 3
The accuracy of 2-bit, 3-bit and 4-bit Quantformer, where the number of partitions in the group-wise quantization is varied. Param. depicts the number of network parameters and BOPs is the bit-operations.

Bitwidth	#Groups.	Params.	BOPs	Top-1	Top-5
2-bit	1	0.39M	22.3G	57.9	82.1
	2	0.39M	22.4G	58.9	82.7
	4	0.39M	22.7G	60.1	83.4
	8	0.39M	23.2G	60.7	84.0
	16	0.39M	24.3G	60.8	84.1
3-bit	1	0.53M	27.6G	62.0	84.8
	2	0.53M	27.7G	63.7	86.0
	4	0.54M	28.0G	64.3	86.4
	8	0.54M	28.5G	65.2	87.0
	16	0.54M	29.6G	65.3	87.2
4-bit	1	0.69M	34.9G	68.0	88.5
	2	0.69M	35.0G	69.2	89.2
	4	0.69M	35.3G	69.6	89.5
	8	0.69M	35.8G	69.9	89.7
	16	0.70M	36.9G	69.9	89.8

we jointly updated the network parameters, quantization thresholds and importance weight of different quantization strategies. For the overall objective optimization in differentiable search, the hyperparameter α_1 and α_2 that control the importance of the self-attention rank consistency loss and discrepancy minimization loss were set to 0.6 and 0.05 respectively. In the finetuning stage for discretized models with the optimal group-wise quantization, we removed the discrepancy minimization loss in the optimization.

For the parameter optimization in differentiable search on the ImageNet dataset, the number of the training epochs is 100 for all architectures. The learning rate started from $2e$-5, $5e$-5, $8e$-5 for quantized transformers in 2, 3, 4 bits respectively and all ended with $1e$-6 via the cosine annealing decay strategy [46]. The quantized models with the optimal feature element partition in group-wise quantization were finetuned with 20 epochs via the same learning rate schedule. For object detection, the backbone was initialized by the full-precision weight released by [35]. The bitwidth of the patch embedding layer and the prediction layers in classification and detection is set to eight, and inference in other convolutional layers of the detection head applies low-precision numbers. During the differentiable search with the COCO dataset, the learning rate was initially set as $1e$-5, $3e$-5, $5e$-5 for quantized transformers in 2, 3, 4 bits respectively, which was also decayed to $1e$-6 with the cosine annealing decay strategy for the total 20 epochs. Similarly, the acquired quantized vision transformer with the optimal group assignment in group-wise quantization was sequentially trained by 5 epochs with the same learning rate settings as in differentiable search. The batchsize was set as 512 and 8 for experiments on ImageNet and COCO respectively, and the AdamW optimizer [19] was leveraged to update the network parameters.

4.2 Ablation Study

Since quantization sizably deviates the self-attention in quantized transformers from that in the full-precision counterparts, we enforce the self-attention in quantized models to mimic that in full-precision transformers without capacity insufficiency. In order to investigate the effectiveness of the self-attention rank preservation with capacity-aware distribution, we varied the hyperparameter α_1 that controlled the importance of self-attention rank consistency loss in the overall objective, and assigned the value of p_l in (7) with different strategies. Meanwhile, we adopted quantization strategies with different thresholds and quantization points for diversely distributed patch features in various dimensions, which alleviate the rounding and clipping errors caused by the shared quantization strategy. Aiming at verifying the quantization loss reduction brought by group-wise quantization, we trained our group-wise quantization with different numbers of partitions. We also changed the initialization strategy for the learnable quantization range to observe the influences. To demonstrate the impact of the gap between continuous space and discrete group assignment in the differentiable search, we adjusted the hyperparamter α_2 that balanced the importance of discrepancy minimization loss in the overall objective. Finally, we visualized the branch importance evolution during the differentiable search. We conducted the ablation study with the DeiT-T architecture on ImageNet.

Table 4
The performance w.r.t. different initialization strategies for the quantization range. KLD demonstrates the calibration method that minimize the KL divergence between the quantized and full-precision features. Max means that the quantization range is symmetrically set to the maximum absolute value of the real-valued feature elements in the group. Percentile stands for the quantization range initialization method adopted in Quantformer.

Method	W/A-bit	Top-1	Top-5
KLD	2/2	58.3	82.0
	3/3	63.0	85.4
	4/4	68.8	89.3
Max	2/2	58.4	82.4
	3/3	64.1	86.3
	4/4	69.3	89.6
Percentile	2/2	60.7	84.0
	3/3	65.2	87.0
	4/4	69.9	89.7

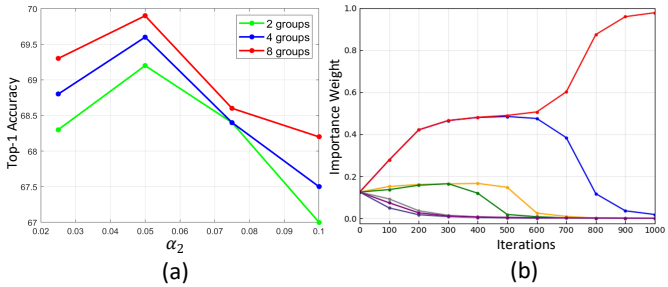


Figure 9. (a) The top-1 classification accuracy with varied hyperparameter α_2 in the overall objecive for 4-bit DeiT-T, where different numbers of partitions for group-wise quantization are leveraged. (b) The evolution of branch importance weights during the differentiable search.

4.2.1 *Effects of Self-attention Rank Preservation with Capacity-aware Distribution*

Performance w.r.t. the hyperparameter α_1 in the overall objective: The hyperparameter α_1 in the overall training objective controls the importance of the self-attention rank consistency loss. We varied α_1 from 0 to 1 for quantized

Table 5
The storage cost, computation complexity and the accuracy on ImageNet of different network quantization methods across various vision transformer architectures and bitwidth settings. Param. depicts the number of network parameters and BOPs is the bit-operations, which evaluate the storage and computational cost respectively.

Model	Method	2-bit				3-bit				4-bit			
		Param.	BOPs	Top-1	Top-5	Param.	BOPs	Top-1	Top-5	Param.	BOPs	Top-1	Top-5
DeiT-T	Full-precision	5.11M	1.3T	72.2	91.1	5.11M	1.3T	72.2	91.1	5.11M	1.3T	72.2	91.1
	PACT	0.39M	22.3G	57.5	81.6	0.53M	27.6G	60.6	83.9	0.69M	34.9G	65.6	87.2
	DSQ	0.39M	22.3G	57.7	81.8	0.53M	27.6G	61.0	84.2	0.69M	34.9G	66.3	87.8
	LSQ	0.39M	22.3G	58.5	82.5	0.53M	27.6G	64.0	85.2	0.69M	34.9G	67.8	88.6
	Quantformer	0.39M	23.2G	60.7	84.0	0.54M	28.5G	65.2	87.0	0.69M	35.8G	69.9	89.7
DeiT-S	Full-precision	22.32M	4.7T	79.9	95.0	22.32M	4.7T	79.9	95.0	22.32M	4.7T	79.9	95.0
	PACT	1.55M	53.0G	61.6	84.5	2.22M	73.9G	72.7	91.4	2.89M	103.2G	76.0	93.1
	DSQ	1.55M	53.0G	61.6	84.5	2.22M	73.9G	73.0	90.9	2.89M	103.2G	76.0	92.9
	LSQ	1.55M	53.0G	61.9	84.9	2.22M	73.9G	74.1	91.9	2.89M	103.2G	76.7	93.3
	Quantformer	1.55M	54.8G	65.2	87.1	2.23M	75.7G	75.4	92.8	2.90M	105.0G	78.2	94.2
DeiT-B	Full-precision	86.64M	18.0T	81.8	95.6	86.64M	18.0T	81.8	95.6	86.64M	18.0T	81.8	95.6
	PACT	5.71M	139.5G	70.7	90.2	8.38M	223.2G	76.9	93.3	11.04M	340.3G	78.3	93.8
	DSQ	5.71M	139.5G	70.9	90.0	8.38M	223.2G	77.0	93.5	11.04M	340.3G	78.4	93.9
	LSQ	5.71M	139.5G	71.2	90.5	8.38M	223.2G	77.7	93.6	11.04M	340.3G	78.5	94.0
	Quantformer	5.71M	143.2G	73.8	92.0	8.38M	226.9G	78.3	93.9	11.05M	344.0G	79.7	94.3
Swin-T	Full-precision	28.00M	4.6T	81.2	95.5	28.00M	4.6T	81.2	95.5	28.00M	4.6T	81.2	95.5
	PACT	2.22M	78.7G	70.6	89.8	3.03M	99.5G	74.8	92.5	3.85M	128.6G	77.0	93.3
	DSQ	2.22M	78.7G	70.6	90.2	3.03M	99.5G	76.0	92.9	3.85M	128.6G	77.6	93.5
	LSQ	2.22M	78.7G	71.5	90.5	3.03M	99.5G	75.9	92.7	3.85M	128.6G	77.8	93.8
	Quantformer	2.23M	81.6G	74.2	92.1	3.04M	102.4G	77.4	93.7	3.85M	131.5G	78.3	94.2
Swin-S	Full-precision	49.68M	8.9T	83.2	96.2	49.68M	8.9T	83.2	96.2	49.68M	8.9T	83.2	96.2
	PACT	3.55M	135.2G	73.0	91.0	5.03M	176.8G	76.8	93.0	6.51M	235.1G	78.9	93.7
	DSQ	3.55M	135.2G	73.7	92.0	5.03M	176.8G	77.4	93.7	6.51M	235.1G	79.5	94.6
	LSQ	3.55M	135.2G	74.9	92.6	5.03M	176.8G	78.0	93.9	6.51M	235.1G	79.9	94.6
	Quantformer	3.56M	139.9G	76.7	93.3	5.04M	181.5G	79.2	94.6	6.51M	239.8G	81.0	95.1

transformers in 2, 3, and 4 bits respectively, where the top-1 accuracy is reported in Figure 8(a). Medium α_1 achieves the optimal performance, because large α_1 harms the semantic information learned in the quantized transformers while the small one fails to enforce the quantized transformers to focus on important tokens due to the quantization errors.

Performance w.r.t. different value assignment strategies for p_l in (7): The l_p norm of the self-attention in full-precision transformers preserves the rank while adjusts the concentration degree for quantized models with different capacities. To investigate the influence of the value assignment strategies for p_l, we optimized the self-attention rank consistency loss with fixed p_l that was manually assigned and dynamic p_l that was decided based on capacity. For fixed p_l, the value was assigned with 1, 2, 3, 4 and 5 respectively representing various concentration degree in the self-attention rank preservation for quantized transformers, where Figure 8(b) demonstrates the top-1 accuracy. The performance of quantized transformers utilizing different dynamic strategies for p_l is demonstrated in Table 2, where the relationship between the self-attention entropy of quantized and full-precision models for p_l definition is varied.

The dynamic strategy outperforms the fixed one in all bitwidth settings, which indicates the effectiveness of the adaptive self-attention distribution based on network capacity. For fixed strategies, medium p_l shows superior performance compared with others. Small p_l enforces the self-attention in quantized transformers to focus on excess patches with capacity insufficiency, and large p fails to fully utilize the capacity of the quantized model with information loss. For the dynamic strategy, utilizing the division between the self-attention entropy of quantized and full-precision transformers as p_l achieves the best results. The comparison with the hinge loss of ranking-aware quantization [37] is demonstrated in Appendix B.

4.2.2 *Effects of Group-wise Quantization*

Performance w.r.t. the hyperparameter α_2 in the overall objective: The hyperparameter α_2 in the overall loss demonstrates the importance of discrepancy minimization between the continuous search space and the discrete assignment for feature element partition. Similar to the ablation study for the hyperparameter of self-attention consistency loss, α_2 ranged in $[0, 0.1]$ was evaluated on 4-bit DeiT-T and different numbers of partitions were also utilized in the group-wise quantization. Figure 9(a) illustrates the top-1 accuracy, where medium α_2 outperforms other settings. High α_2 disables the model to learn the task-relevant information and low α_2 fails to alleviate the significant discrepancy between the continuous search space and the discrete group assignment after the differentiable search.

Performance w.r.t. the partition numbers in group-wise quantization: Discretizing features in different dimensions with various quantization strategies alleviates quantization errors while increases the storage and computational complexity. We implemented our Quantformer with different partition numbers in group-wise quantization to investigate the influence on accuracy and efficiency. Table 3 illustrates the results, where the performance enhancement for group-wise quantization with more than 8 partitions is slight with

significantly increased complexity overhead. To efficiently quantize vision transformers with sizable accuracy increase, we assign the number of partitions with 8 in other experiments.

Performance w.r.t. the initialization strategy for the learnable quantization range: As empirically proven in [48] that quantization range initialization contributes significantly to the final accuracy, we learned the optimal quantization range initialization with a small calibration set which consisted of 128 samples from the training images. The initialized quantization range was decided in the following three ways: with the goal of minimizing the KL divergence between the quantized and real-valued tensors [48], symmetrically assigned with the maximal absolute value of real-valued elements, or set to the percentile of the full-precision feature distribution [29]. The top-1 accuracy of our Quantformer with different calibration strategies is shown in Table 4, where assigning the initialized quantization range with the percentile in feature distribution acquires the highest accuracy. The outliers in patch features influence the initialized quantization range obtained via the maximal real-valued elements or that acquired by minimizing the KL divergence between the quantized and real-valued features, while initializing the quantization range with the percentile in feature distribution is robust to outliers which contribute significantly to quantization errors.

Visualization of branch importance in differentiable search: Figure 9(b) depicts the evolution of branch importance weights during the differentiable search for group assignment. At the early stage of the differentiable search, the difference among the branch importance weights is not obvious because the task risk dominates instead of the discrepancy minimization loss. Meanwhile, the network weights are not well-trained and the contribution of various quantization strategies is similar, so that the update of the branch importance weights is not significant. When the network gradually converges, the task loss only makes slight contribution in the gradient descent. The principal impacts on the performance are resulted from the quantization functions with different rounding and clipping errors. The discrepancy minimization loss dominates and the branch importance weights are updated significantly. Therefore, the difference among branch importance weights increases with alleviated discretization discrepancy.

4.3 Comparison with the State-of-the-art Network Quantization Methods

In this section, we compare our Quantformer with the state-of-the-art network quantization methods including PACT [9], DSQ [18] and LSQ [16] on ImageNet for image classification and on COCO for object detection. The architectures of DeiT-T/S/B [46] and Swin-T/S [35] were employed for the evaluation on image classification, and we leveraged Swin-T/S [35] as the backbone for the comparison on object detection. We also provide the performance of full-precision vision transformers for reference. The accuracy of state-of-the-art methods were obtained by re-running the officially released code or re-implementing the approaches to quantize the vision transformers.

Table 6
Comparison of storage cost, computational complexity, top-1 and top-5 classification accuracies on ImageNet with state-of-the-art mixed-precision quantization in DeiT-T/S/B. HAQ+Quant. and EdMIPS+Quant. respectively represent the combination of HAQ and Quantformer and the integration of EdMIPS and Quantformer .

Methods	Params.	BOPs	Top-1	Top-5
DeiT-T				
Baseline	5.11M	1.3T	72.2	91.1
HAQ	0.50M	20.9G	62.6	85.3
HAQ+Quant.	0.50M	21.8G	65.3	86.4
EdMIPS	0.45M	23.3G	62.1	85.0
EdMIPS+Quant.	0.45M	24.2G	64.9	86.2
HAQ	0.63M	31.7G	67.5	87.8
HAQ+Quant.	0.63M	32.6G	69.6	89.9
EdMIPS	0.62M	32.3G	67.7	88.0
EdMIPS+Quant.	0.62M	33.2G	69.4	89.8
DeiT-S				
Baseline	22.10M	4.7T	79.9	95.0
HAQ	2.07M	58.6G	72.5	91.1
HAQ+Quant.	2.07M	60.4G	75.1	92.6
EdMIPS	1.86M	64.7G	72.7	91.0
EdMIPS+Quant.	1.86M	66.5G	75.0	92.4
HAQ	2.69M	93.2G	75.8	93.0
HAQ+Quant.	2.69M	95.0G	76.9	93.3
EdMIPS	2.64M	93.3G	75.7	92.9
EdMIPS+Quant.	2.64M	95.1G	77.0	93.6
DeiT-B				
Baseline	87.80M	15.8T	83.5	96.5
HAQ	7.82M	184.4G	77.6	93.1
HAQ+Quant.	7.82M	188.1G	78.6	94.2
EdMIPS	7.01M	201.9G	77.4	93.3
EdMIPS+Quant.	7.01M	205.6G	78.2	93.9
HAQ	10.26M	305.4G	78.3	93.9
HAQ+Quant.	10.27M	309.1G	79.2	94.3
EdMIPS	10.08M	301.4G	78.4	94.0
EdMIPS+Quant.	10.08M	305.1G	79.7	94.5

4.3.1 *Evaluation on image classification*

Results on ImageNet: Table 5 shows the comparison of storage cost, computational complexity, top-1 and top-5 accuracies on ImageNet across different vision transformer architectures and network quantization methods, where the bitwidths of weights and activations for the fully-connected layers were set as 2, 3, and 4 respectively. Our Quantformer significantly accelerates the computation and save the storage cost by $7.84\times$ (86.64M vs. 11.05M) and $52.33\times$ (18.0T vs. 344.0G) for DeiT-B and $7.63\times$ (49.68M vs. 6.51M) and $37.11\times$ (8.9T vs. 239.8G) for Swin-S in 4-bit settings. The efficiency enhancement is less notable for Swin Transformer because of the high-precision downsampling layers for patch concatenation without significant performance degradation.

PACT enables the quantization thresholds to be learnable, and DSQ utilizes the differentiable soft quantization function in the forward and backward passes during training to reduce the optimization difficulty. LSQ accurately approximates the gradient to the quantizer step size for more fine-grained optimization and scales the step size to improve convergence. However, they all face the challenges of deviated self-attention rank and significant discretization errors in quantized vision transformers. On the contrary, our Quantformer employs capacity-aware self-

Table 7
The storage and computation complexity, mean average precision for IOU from 0.5 to 0.95 (mAP) and average precision with the IOU threshold 50% (AP_{50}) and 75% (AP_{75}) of different network quantization methods across various vision transformer architectures and detection frameworks, where the bitwidth was set to 2, 3, 4 respectively. Param. depicts the number of network parameters and BOPs is the bit-operations, which evaluate the storage and computational cost respectively.

Backbone	BitWidth	Method	Mask R-CNN					Cascade R-CNN				
			Param.	BOPs	mAP	AP_{50}	AP_{75}	Param.	BOPs	mAP	AP_{50}	AP_{75}
Swin-T	32bit	–	48.00M	273.41T	46.0	68.1	50.3	86.00M	762.88T	50.4	69.2	54.7
	2bit	PACT	6.64M	2.28T	38.5	60.4	42.5	16.07M	4.19T	42.0	60.7	45.9
		DSQ	6.64M	2.28T	39.5	61.6	43.5	16.07M	4.19T	43.2	61.3	47.0
		LSQ	6.64M	2.28T	40.0	62.2	44.3	16.07M	4.19T	43.3	63.6	47.3
		Quantformer	6.64M	2.34T	41.8	63.4	44.9	16.07M	4.25T	44.2	63.9	48.8
	3bit	PACT	7.54M	3.49T	40.8	62.9	45.4	16.97M	7.79T	46.0	64.4	50.2
		DSQ	7.54M	3.49T	41.2	63.2	45.1	16.97M	7.79T	46.3	64.8	50.2
		LSQ	7.54M	3.49T	41.7	64.0	45.6	16.97M	7.79T	46.2	64.7	50.1
		Quantformer	7.54M	3.55T	42.8	64.4	47.0	16.97M	7.85T	47.1	65.2	51.6
	4bit	PACT	8.45M	5.17T	42.7	64.9	47.1	17.88M	12.82T	46.8	65.3	50.9
		DSQ	8.45M	5.17T	43.4	65.8	47.8	17.88M	12.82T	47.7	66.3	51.7
		LSQ	8.45M	5.17T	43.7	65.8	48.1	17.88M	12.82T	48.2	66.7	52.2
		Quantformer	8.46M	5.23T	44.9	66.9	49.4	17.89M	12.88T	49.1	67.2	53.3
Swin-S	32bit	–	69.00M	367.62T	48.5	70.2	53.5	107.00M	858.11T	51.9	70.7	56.3
	2bit	PACT	7.96M	3.48T	40.4	62.0	44.8	17.39M	5.39T	42.8	61.4	46.8
		DSQ	7.96M	3.48T	41.2	62.9	45.2	17.39M	5.39T	43.8	62.1	47.7
		LSQ	7.96M	3.48T	41.9	63.4	46.5	17.39M	5.39T	43.3	61.9	47.7
		Quantformer	7.96M	3.58T	43.8	65.5	47.5	17.39M	5.49T	44.7	63.4	48.9
	3bit	PACT	9.54M	5.13T	43.4	65.3	47.8	18.97M	9.43T	45.0	63.7	49.4
		DSQ	9.54M	5.13T	44.2	66.0	48.9	18.97M	9.43T	45.8	64.5	50.1
		LSQ	9.54M	5.13T	44.3	65.8	48.7	18.97M	9.43T	45.3	64.2	49.6
		Quantformer	9.55M	5.23T	45.5	66.7	49.3	18.98M	9.53T	46.8	65.2	51.2
	4bit	PACT	11.11M	7.43T	44.8	66.4	49.6	20.54M	15.08T	49.8	68.4	54.1
		DSQ	11.11M	7.43T	45.2	67.1	49.7	20.54M	15.08T	50.9	69.3	55.3
		LSQ	11.11M	7.43T	45.1	66.9	49.6	20.54M	15.08T	50.8	69.4	55.2
		Quantformer	11.12M	7.53T	47.2	69.0	52.0	20.55M	15.18T	51.5	70.1	56.1

attention imitation to preserve the self-attention rank consistency between quantized and full-precision transformers, and leverages group-wise quantization with various thresholds and rounding points for patch features across different dimensions. Therefore, our Quantformer improves the top-1 accuracy by a sizable margin in vision transformers with different bitwidths and various architectures compared with the state-of-the-art baseline methods. Comparison with more baseline methods are illustrated in Appendix D.

The presented techniques including self-attention rank preservation and group-wise quantization can be integrated with mixed-precision quantization methods for further performance boosting under the given storage and computational cost constraint. Mixed precision quantization assigns optimal bitwidths to different layers according to their informativeness to enhance the accuracy-complexity trade-off. As the vision transformer with mixed-precision quantization also suffers from deviated self-attention rank and significant quantization errors, we employed our Quantformer as an plug-and-play module for DeiT architectures where the layer-wise bitwidth assignment was decided by HAQ [48] and EdMIPS [3]. The implementation details are demonstrated in Appendix A. In order to show the performance with different accuracy-complexity trade-offs, we applied two BOPs constraints for each architecture. Table 6 illustrates the results, where the integration of Quantformer can strengthen the performance of vanilla mixed-precision quantization methods across different complexity budgets.

4.3.2 Evaluation on Object Detection

Results on COCO: Despite of demonstrating the number of parameters, BOPs, mean average precision (mAP), we also report the average precision at different IOU thresholds with different bitwidth settings in object detection. Meanwhile, we leveraged both the Mask R-CNN [22] and Cascade R-CNN [2] to evaluate the generalization ability across different detection frameworks for our Quantformer. Table 7 shows the storage cost, computational complexity and the accuracy, where our Quantformer outperforms the baseline methods across various architectures and detection frameworks. With similar parameter numbers and BOPs to the state-of-the-art methods LSQ, the presented Quantformer improves the mAP of 4-bit Swin-T by 1.2% (44.9% vs. 43.7%) and 0.9% (49.1% vs. 48.2%) with Mask R-CNN and Cascade R-CNN, and 2.1% (47.2% vs. 45.1%) and 0.7% (51.5% vs. 50.8%) of 4-bit Swin-S with the above detection frameworks. Meanwhile, the performance enhancement in the Mask R-CNN detection framework is more obvious than Cascade R-CNN, as the hierarchical structures in Cascade R-CNN help alleviate information loss in quantized vision transformers. Therefore, the self-attention rank preservation and the group-wise quantization strengthen information retention more significantly for the Mask R-CNN.

5 Conclusion

In this paper, we have proposed extremely low-precision vision transformers called Quantformer for efficient inference. The presented Quantformer preserves the self-attention

rank consistency between quantized transformers and full-precision counterparts with capacity-ware distribution, so that the long-range dependencies are correctly mined without capacity inefficiency. We partition features in different dimensions for group-wise quantization to minimize the discretization errors with negligible complexity overhead. Extensive experiments on image classification and object detection have demonstrated the superiority of Quantformer.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603, Grant 61822603 and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, *et al.* Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *TPAMI*, 43(5):1483–1498, 2019.

[3] Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. In *CVPR*, pages 2349–2358, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[5] Boyu Chen, Peixia Li, Baopu Li, Chuming Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Psvit: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021.

[6] Boyu Chen, Peixia Li, Chuming Li, Baopu Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Glit: Neural architecture search for global and local image transformer. In *ICCV*, pages 12–21, 2021.

[7] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021.

[8] Minghao Chen, Kan Wu, Bolin Ni, Houwen Peng, Bei Liu, Jianlong Fu, Hongyang Chao, and Haibin Ling. Searching the search space of vision transformer. *NIPS*, 2021.

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

[10] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair darts: Eliminating unfair advantages in differentiable architecture search. In *ECCV*, pages 465–480, 2020.

[11] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 1(2):3, 2021.

[12] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] 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.

[16] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[17] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

[18] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, pages 4852–4861, 2019.

[19] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[20] Yushuo Guan, Pengyu Zhao, Bingxuan Wang, Yuanxing Zhang, Cong Yao, Kaigui Bian, and Jian Tang. Differentiable feature aggregation search for knowledge distillation. In *ECCV*, pages 469–484, 2020.

[21] Yi Guo, Huan Yuan, Jianchao Tan, Zhangyang Wang, Sen Yang, and Ji Liu. Gdp: Stabilized neural network pruning via gates with differentiable polarization. In *ICCV*, pages 5239–5250, 2021.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[23] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *ICCV*, pages 1389–1397, 2017.

[24] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NIPS*, pages 4114–4122, 2016.

[25] Yerlan Idelbayev and Miguel A Carreira-Perpinán. Low-rank compression of neural nets: Learning the rank of each layer. In *CVPR*, pages 8049–8059, 2020.

[26] Hyeji Kim, Muhammad Umar Karim Khan, and Chong-Min Kyung. Efficient neural network compression. In *CVPR*, pages 12569–12577, 2019.

[27] Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. I-bert: Integer-only bert quantization. *arXiv preprint arXiv:2101.01321*, 2021.

[28] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *CVPR*, pages 6448–6457, 2021.

[29] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, pages 2810–2819, 2019.

[30] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*, 2019.

[31] 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 *ECCV*, pages 740–755, 2014.

[32] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[33] Xingchao Liu, Mao Ye, Dengyong Zhou, and Qiang Liu. Post-training quantization with multiple points: Mixed precision without mixed precision. In *AAAI*, volume 35, pages 8697–8705, 2021.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[35] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[36] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, pages 722–737, 2018.

[37] Zhenhua Liu, Yunhe Wang, Kai Han, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *arXiv preprint arXiv:2106.14156*, 2021.

[38] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *ICCV*, pages 8741–8750, 2021.

[39] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and

bias correction. In *ICCV*, pages 1325–1334, 2019.
[40] Bo Peng, Wenming Tan, Zheyang Li, Shun Zhang, Di Xie, and Shiliang Pu. Extreme network compression via filter group approximation. In *ECCV*, pages 300–316, 2018.
[41] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NIPS*, 34:13937–13949, 2021.
[42] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542, 2016.
[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.
[44] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI*, volume 34, pages 8815–8821, 2020.
[45] Yehui Tang, Yunhe Wang, Yixing Xu, Yiping Deng, Chao Xu, Dacheng Tao, and Chang Xu. Manifold regularized dynamic network pruning. In *CVPR*, pages 5018–5028, 2021.
[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021.
[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.
[48] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *CVPR*, pages 8612–8620, 2019.
[49] Ying Wang, Yadong Lu, and Tijmen Blankevoort. Differentiable joint pruning and quantization for hardware efficiency. In *ECCV*, pages 259–277, 2020.
[50] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *NIPS*, 34:11960–11973, 2021.
[51] Ziwei Wang, Jiwen Lu, Ziyi Wu, and Jie Zhou. Learning efficient binarized object detectors with information compression. *TPAMI*, 44(6):3082–3095, 2021.
[52] Ziwei Wang, Jiwen Lu, and Jie Zhou. Learning channel-wise interactions for binary convolutional neural networks. *TPAMI*, 43(10):3432–3445, 2021.
[53] Ziwei Wang, Han Xiao, Jiwen Lu, and Jie Zhou. Generalizable mixed-precision quantization via attribution rank preservation. In *ICCV*, pages 5291–5300, 2021.
[54] Ziwei Wang, Han Xiao, Yueqi Duan, Jie Zhou, and Jiwen Lu. Learning deep binary descriptors via bitwise interaction mining. *TPAMI*, 2022.
[55] Han Xiao, Ziwei Wang, Zheng Zhu, Jie Zhou, and Jiwen Lu. Shapley-nas: Discovering operation contribution for neural architecture search. In *CVPR*, pages 11892–11901, 2022.
[56] Sheng Xu, Yanjing Li, Teli Ma, Bohan Zeng, Baochang Zhang, Peng Gao, and Jinhu Lu. Tervit: An efficient ternary vision transformer. *arXiv preprint arXiv:2201.08050*, 2022.
[57] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. *arXiv preprint arXiv:2108.01390*, 2021.
[58] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.
[59] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization framework for vision transformers. *arXiv preprint arXiv:2111.12293*, 2021.
[60] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*, 2019.
[61] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *ECCV*, pages 365–382, 2018.
[62] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *ICML*, pages 7543–7552, 2019.
[63] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, *et al.* Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021.

Ziwei Wang received the B.S. degree from the Department of Physics, Tsinghua University, China, in 2018. He is currently working toward the PhD degree in the Department of Automation, Tsinghua University, China. His research interests include network compression and binary representation. He has published over 10 scientific papers in TPAMI, CVPR, ICCV and ECCV. He serves as a regular reviewer member for TIP, TCSVT, CVPR, ICCV, ECCV, NeurIPS, ICML, IJCAI, ICLR, WACV and ICME.

Changyuan Wang is currently an undergraduate student in the School of Artificial intelligence, Beijing Normal University. His research interests in computer vision, efficient inference and unmanned systems.

Xiuwei Xu received the B.Eng degree from Tsinghua University in 2021. He is currently a PhD candidate in the Department of Automation at Tsinghua University. His research interests include data/computation-efficient learning and 3D vision.

Jie Zhou (M'01-SM'04) received the BS and MS degrees both from the Department of Mathematics, Nankai University, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), China, in 1995. From then to 1997, he served as a postdoctoral fellow in the Department of Automation, Tsinghua University, China. Since 2003, he has been a full professor in the Department of Automation, Tsinghua University. His research interests include computer vision and pattern recognition. In recent years, he has authored more than 100 papers have been published in TPAMI, TIP and CVPR. He is an associate editor for TPAMI and two other journals. He received the National Outstanding Youth Foundation of China Award. He is a senior member of the IEEE and Fellow of the IAPR.

Jiwen Lu (M'11-SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, and a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He serves as the Co-Editor-of-Chief for PRL, an Associate Editor for the TIP, TCSVT, the TBIOM, and Pattern Recognition. He also serves as the Program Co-Chair of IEEE FG'2023, VCIP'2022, AVSS'2021 and ICME'2020. He received the National Outstanding Youth Foundation of China Award. He is an IAPR Fellow.