Instance-Aware Group Quantization for Vision Transformers

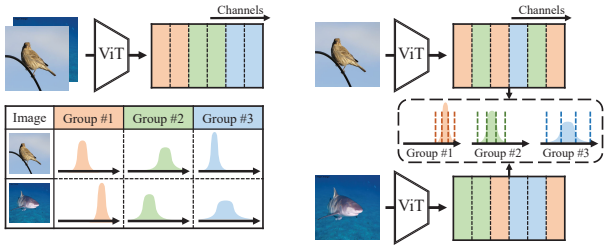
Jaehyeon Moon[1,2] Dohyung Kim[1] Junyong Cheon[1] Bumsub Ham[1*]
[1]Yonsei University [2]Articron
https://cvlab.yonsei.ac.kr/projects/IGQ-ViT/

Abstract

Post-training quantization (PTQ) is an efficient model compression technique that quantizes a pretrained full-precision model using only a small calibration set of unlabeled samples without retraining. PTQ methods for convolutional neural networks (CNNs) provide quantization results comparable to full-precision counterparts. Directly applying them to vision transformers (ViTs), however, incurs severe performance degradation, mainly due to the differences in architectures between CNNs and ViTs. In particular, the distribution of activations for each channel vary drastically according to input instances, making PTQ methods for CNNs inappropriate for ViTs. To address this, we introduce instance-aware group quantization for ViTs (IGQ-ViT). To this end, we propose to split the channels of activation maps into multiple groups dynamically for each input instance, such that activations within each group share similar statistical properties. We also extend our scheme to quantize softmax attentions across tokens. In addition, the number of groups for each layer is adjusted to minimize the discrepancies between predictions from quantized and full-precision models, under a bit-operation (BOP) constraint. We show extensive experimental results on image classification, object detection, and instance segmentation, with various transformer architectures, demonstrating the effectiveness of our approach.

1. Introduction

Transformers [34] can capture long-range dependencies across sequential inputs, which is of central importance in natural language processing, aggregating contextual information and providing discriminative feature representations. Recently, vision transformers (ViTs) [10] has demonstrated the effectiveness of transformers for images, providing state-of-the-art results on various visual recognition tasks, including image classification [24, 33], object detection [24, 40], and semantic segmentation [24, 32, 38]. However, a series of fully-connected (FC) and self-attention layers in ViTs requires a substantial amount of memory and computational cost, making it challenging to deploy them on devices with limited resources (*e.g.*, drones and mobile phones). The growing demand for ViTs to operate on the resource-constrained devices has led to increased interest in developing network quantization techniques for ViTs.



(a) Group quantization (b) IGQ-ViT

Figure 1. Visual comparison of group quantization and IGQ-ViT. (a) Conventional group quantization techniques [7, 31] divide consecutive channels uniformly into a number of groups without considering their dynamic ranges. The distribution of activations in each group varies significantly for individual input instances. (b) To alleviate this problem, IGQ-ViT proposes an instance-aware grouping technique that splits the channels of activation maps and softmax attentions across tokens dynamically for each input instance at runtime.

Network quantization generally reduces bit-widths of weights and activations of a model for an efficient inference process, which can be categorized into two groups: Quantization-aware training (QAT) and post-training quantization (PTQ). QAT methods [11, 41, 42] train full-precision models, while simulating the quantization process by inserting discretizers into networks to quantize, such that the discrepancy between the full-precision and quantized models is minimized in terms of accuracy. This suggests that QAT methods require entire training samples, and they are computationally expensive, making them impractical for the prompt deployment of neural networks. PTQ methods [19, 26, 36], on the other hand, calibrate quantization parameters (*e.g.*, quantization intervals, zero-points) from pretrained full-precision models, enabling faster quantization of networks compared to QAT methods with only a

*Corresponding author.

limited number of training samples (usually less than 1k).

Several PTQ methods for transformers [9, 25, 39] apply layer-wise quantization techniques, where a single quantizer is applied to all activation values for efficiency. These methods, however, are not directly applicable for quantizing models using extremely low bit-widths (*e.g.*, 4-bit), due to the significant scale variation on the activations for each channel. Exploiting channel-wise quantizers (*i.e.*, applying different quantizers for each channel) could be a potential solution, but at the expense of computational overheads, due to floating-point summations of channel-wise outputs for matrix multiplication. Group quantization techniques [7, 31] could be an alternative to address this problem, where they divide consecutive channels uniformly into multiple groups, and apply a single quantizer for each group (Fig. 1a). However, we have observed that the channel-wise distributions of activation values vary largely among different samples, making conventional approaches inappropriate for ViTs.

In this paper, we present instance-aware group quantization for ViTs (IGQ-ViT), that effectively and efficiently addresses the variations of channel-wise distributions across different input instances (Fig. 1b). Specifically, we split the channels of activation maps into multiple groups dynamically, such that the activation values within each group share similar statistical properties, and then quantize the activations within the group using identical quantization parameters. We also propose to use the instance-aware grouping technique to softmax attentions, since the distributions of attention values vary significantly according to tokens. In addition, we present a simple yet effective method to optimize the number of groups for individual layers, under a bit-operation (BOP) constraint. IGQ-ViT can be applied to various components in ViTs, including input activations of FC layers and softmax attentions, unlike previous methods [20, 22, 25, 39] that are limited to specific parts of transformer architectures. We demonstrate the effectiveness and efficiency of IGQ-ViT for various transformers, including ViT [10] and its variants [24, 33], and show that IGQ-ViT achieves state-of-the-art results on standard benchmarks. We summarize the main contributions of our work as follows:

- We introduce a novel PTQ method for ViTs, dubbed IGQ-ViT, that splits channels of activation maps into a number of groups dynamically according to input instances. We also propose to use the instance-aware grouping technique to split softmax attentions across tokens.
- We present a group size allocation technique searching for an optimal number of groups for each layer given a BOP constraint.
- We set a new state of the art on image classification [8], object detection, and instance segmentation [21], with various ViT architectures [10, 24, 33].

2. Related work

Network quantization. Network quantization aims at reducing bit-widths of weights and activations of neural networks. QAT methods simulate the quantization process by applying a round function to weights and activations of the network. Since derivatives of the round function is either zero or infinite, they approximate the gradients (*e.g.*, using the straight-through estimator [3]) to train the network with backpropagation. These methods also adjust the derivatives of the round function [17, 18] or train quantization parameters jointly with network weights based on task losses [11, 16]. For better convergence of the training process, many heuristics have been introduced, *e.g.*, progressively shrinking bit-widths [42] or freezing parts of the network weights [28, 41]. Quantized networks using QAT show performance comparable to or even better then full-precision counterparts. However, the quantization process is computationally demanding, requiring a significant amount of training time. PTQ offers an alternative approach to quantizing neural networks. Instead of training full-precision models and simulating the quantization process at training time, PTQ methods calibrate quantization parameters (*e.g.*, quantization intervals) using a subset of training samples. Early efforts focus on optimizing the quantization parameters to minimize the difference between floating-point and quantized values [2, 27]. Another line of research proposes to consider distributions of weights and/or activations to design quantizers. For instance, the work of [12] has observed that network weights follow a bell-shaped distribution. Based on this, it introduces piecewise linear quantizers that assign different quantization intervals according to the magnitudes of activations, performing better compared to uniform quantizers. Recent PTQ methods learn to either round up or down network weights by using a reconstruction error of layer outputs [26] or exploiting the Hessian of training losses [19], and they have proven the effectiveness on CNN architectures (*e.g.*, ResNet [13], MobileNetV2 [30]).

Transformer quantization. While ViTs [10] and the variants [24, 33] have become increasingly popular in computer vision, the unique structure and characteristics of ViT architectures makes network quantization challenging. For example, PTQ methods for CNNs [2, 19, 26, 27] do not perform well on quantizing softmax attentions and GELU activations in transformers, suggesting that directly applying them for ViT quantization results in significant performance degradation [25]. To date, only a limited number of PTQ methods have been developed for ViTs. The work of [25] estimates quantization parameters that maximize similarities between full-precision and quantized outputs of linear operations, and proposes to preserve a relative order of attention values after quantization. APQ-ViT [9] introduces a calibration metric to minimize the discrepancies

between full-precision and quantized outputs, while maintaining the power-law distribution of softmax attentions. PTQ4ViT [39] introduces twin uniform quantizers to handle asymmetric distributions in softmax attentions and GELU activations effectively. Most PTQ methods for ViTs exploit a single quantizer for all channels, suggesting that they do not consider the distributions of activation values across channels, typically having extreme scale variations. Recent works [20, 22] attempt to alleviate the scale variation problem efficiently. FQ-ViT [22] proposes to consider inter-channel scale variations for LayerNorm [1], and exploits channel-wise quantizers with the constraint of the ratio of quantization intervals being power-of-two values. This enables using bit-shift operations, calculating mean and variance of LayerNorm in an integer level. The scale reparameterization technique, introduced by RepQ-ViT [20], allows to use layer-wise quantizers, instead of adopting channel-wise ones, by adjusting the affine factors of LayerNorm and the weights of FC layers. However, this technique applies to the activations for LayerNorm only, and does not fully address the inter-channel scale variations for other layers in transformers.

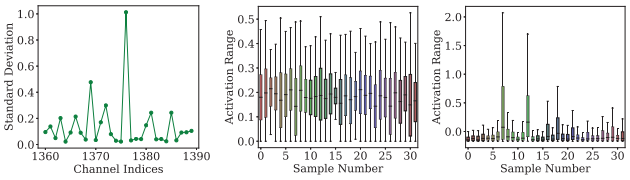
Similar to ours, the works of [4, 7, 31, 35] adopt group quantization techniques for transformers. For instance, Q-bert [31] and VS-quant [7] divide consecutive channels uniformly into a number of groups without considering the dynamic range of each channel, and thus the channels assigned to each group do not follow similar distributions. PEG [4] alleviates this issue by sorting the activations across channels w.r.t. the dynamic ranges during calibration, before grouping the channels. Quantformer [35] proposes to use a differentiable search [6, 23] for QAT in order to group channels of activation maps. The channels assigned to particular groups are however fixed after calibrating pre-trained networks for PTQ in the group quantization techniques [4, 7, 31], which makes them inappropriate for ViTs having diverse channel distributions according to input instances. In contrast, our approach apply group quantization along channels of activation maps and tokens of softmax attentions dynamically at runtime for each input instance, without additional parameters for PTQ.

3. Method

In this section, we provide a brief description of uniform quantizer (Sec. 3.1). We then present our approach in detail, including IGQ-ViT (Sec. 3.2) and a group size allocation technique (Sec. 3.3).

3.1. Uniform quantizer

Given a floating-point value x and the quantization bit-width b, uniform quantizers discretize the inputs into a finite set of values with equally spaced intervals. To this end, it normalizes the floating-point value x using a scale parameter s, calibrate the normalized value with a zero-point z, and clip the output as follows:



(a) (b) (c)

Figure 2. (a) Plots of standard deviations of activations across channels for DeiT-S [33]; (b-c) Boxplots of activation values across different input instances for a particular channel of ResNet-50 [13] and DeiT-S, respectively. We use ImageNet [8] for the visualizations. We have observed that there is a significant scale variation across channels, and the activation ranges for each channel change drastically among different samples for ViTs, in contrast to CNNs.

$$\hat{x} = \text{clip}(\lfloor \frac{x}{s} \rceil + z, 0, 2^b - 1), \tag{1}$$

where the scale parameter s and zero-point z are defined as:

$$s = \frac{u - l}{2^b - 1}, \quad z = \text{clip}(\lfloor -\frac{l}{s} \rceil, 0, 2^b - 1). \tag{2}$$

We denote by u and l upper and lower bounds of the quantizer, respectively. $\lfloor . \rceil$ is a rounding function, and $\text{clip}(., m, n)$ restricts an input to the range with lower and upper bounds of m and n, respectively. The quantized output is then obtained as follows:

$$Q(x; s, z) = s(\hat{x} - z). \tag{3}$$

3.2. IGQ-ViT

Following the work of [25], we quantize all network weights except for the positional embedding. We also quantize input activations of FC layers in the multi-layer perceptron (MLP) block, and the activations for the multi-head self-attention (MSA) block, including queries, keys, values, and softmax attentions. We exploit uniform quantizers for all weights and activations in ViTs.

3.2.1 IGQ for linear operations

In the following, we first provide empirical observations on input activations of FC layers, and explain the details of our IGQ framework for linear operations.

Distributions of activations across channels. Most quantization frameworks [2, 19, 25, 26] exploit layer-wise quantizers for activations, applying a single quantization parameter for all channels for efficient inference. However, we have observed that the input activations of FC layers have significant scale variations across channels (Fig. 2(a)). Similar findings can be found in [20, 22]. This suggests

that layer-wise quantizers degrade the quantization performance significantly, as they cannot handle scale variations across different channels. Although adopting separate quantizers for individual channels could be an effective strategy for overcoming the scale variation problem, this requires a summation of a floating-point output for every channel, which is computationally expensive. We have also found that the ranges of these activations for each channel vary drastically among different input instances (Fig. 2(b, c)), since ViTs do not have preceding BatchNorm [15] layers in contrast to state-of-the-art CNNs (*e.g.*, ResNet [13], MobileNetV2 [30]). Conventional approaches (*e.g.*, [4, 7, 31, 35]) exploit a fixed quantization interval (*i.e.*, from lower to upper bounds of the quantizer) for every input instance, thus cannot adapt to such diverse distributions across different samples.

Instance-aware grouping across channels. We introduce an instance-aware group quantization framework for linear operations that alleviates the scale variation problem, while maintaining efficiency. We split the channels of activation maps into G_1 groups based on statistical properties, where activation values within each group are quantized with identical quantization parameters. We assign the channels of activations to appropriate groups, and optimize the scale parameter s_i and the zero-point z_i for the i-th group. Specifically, given floating-point activations of $\mathbf{X} \in \mathbb{R}^{N \times C}$ and a set of candidate quantizers $\{Q_i\}_{i=1}^{G_1}$, where we denote by N and C as the number of tokens and channels, respectively, we define a distance metric between the c-th channel of the activations $\mathbf{X}$, denoted by $\mathbf{X}_c$, and the quantizer Q_i as follows:

$$d(\mathbf{X}_c, Q_i) = (\min(\mathbf{X}_c) - u_i)^2 + (\max(\mathbf{X}_c) - l_i)^2 \quad (4)$$

where u_i and l_i are upper and lower bounds of the quantizer Q_i, respectively. We then assign each channel of the activation $\mathbf{X}$ to one of candidate quantizers with the minimum distance as follows:

$$\pi(c) = \arg\min_i d(\mathbf{X}_c, Q_i) \quad (5)$$

where we denote by $\pi(c)$ a group index assigned to the c-th channel. The upper and lower bounds of quantizers are then optimized by minimizing the distances as follows:

$$u_i^*, l_i^* = \arg\min_{u_i, l_i} \sum_{\pi(c)=i} d(\mathbf{X}_c, Q_i). \quad (6)$$

We optimize u_i and l_i by solving Eq. (5) and Eq. (6) alternately similar to the expectation-maximization (EM) algorithm, which guarantees the convergence [37]. Finally, we obtain quantization parameters of each group (*i.e.*, $\mathbf{s}$ and $\mathbf{z}$) using Eq. (2). At test time, we fix the quantization parameters, and assign the channels to appropriate groups using Eq. (5).

Table 1. Comparison of BOPs for a 4-bit DeiT-B [33] model using various quantization strategies. We denote by 'Model' the required BOP for layer-wise quantization. In contrast to layer-wise quantization, IGQ-ViT involves additional computations, including (1) computing the min/max values of each channel, (2) assigning channels to quantizers with the minimum distance, and (3) summing the outputs of each group in a floating-point format. The corresponding BOPs for these steps are denoted by 'Minmax', 'Assign', and 'FP sum', respectively.

Methods	Model	Minmax	Assign	FP sum	Total
Layer-wise	340.3G	-	-	-	340.3G
IGQ-ViT(#groups=4)	340.3G	0.99G	0.57G	1.57G	343.4G
IGQ-ViT(#groups=8)	340.3G	0.99G	1.14G	3.66G	346.1G
IGQ-ViT(#groups=16)	340.3G	0.99G	2.28G	7.84G	351.4G

Computational overhead. Compared to layer-wise quantization, our approach requires (1) computing the min/max values of each channel, (2) assigning channels to quantizers with the minimum distance, and (3) summing the floating-point outputs of each group. Specifically, consider a matrix multiplication between a quantized activation $Q(\mathbf{X})$ and the quantized weight $Q(\mathbf{W})$ with a group size of G_1. The quantized activation $Q(\mathbf{X})$ is obtained by partitioning $\mathbf{X}$ into a number of groups across channels using Eq. (5), *i.e.* $\mathbf{X} = [\mathbf{X}_1, ..., \mathbf{X}_{G_1}]$, followed by quantizing $\mathbf{X}_i$ with a scale parameter of s_i, where $i \in \{1, ..., G_1\}$. The matrix multiplication between $Q(\mathbf{X})$ and $Q(\mathbf{W})$ can then be represented as follows:

$$Q(\mathbf{X})Q(\mathbf{W}) = s^w \cdot (\sum_{i=1}^{G_1} s_i \cdot \hat{\mathbf{X}}_i \hat{\mathbf{W}}_i). \quad (7)$$

Note that we omit zero-points for clarity. We denote by s^w the scale parameter for $\mathbf{W}$. $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{W}}_i$ are obtained by applying Eq. (1) to $\mathbf{X}_i$ and $\mathbf{W}_i$, the channels and rows of $\mathbf{X}$ and $\mathbf{W}$ associated with group i, respectively. Computing Eq. (7) requires the summation of floating-point matrices for each group (*i.e.*, $s_i \cdot \hat{\mathbf{X}}_i \hat{\mathbf{W}}_i$), which can be reduced with sufficiently small values of G_1. As the values of G_1, we use no more than 16 in our experiments, which is extremely small compared to the number of channels, usually scaling up to over a thousand. We show in Table 1 BOPs of IGQ-ViT for DeiT-B [33] quantized with 4-bit. We can see that IGQ-ViT introduces only 3.3% additional BOPs for a group size of 16, compared to layer-wise quantization.

3.2.2 IGQ for softmax attentions

Here, we present our observation for the distribution of softmax attentions, and present the details of IGQ for softmax attentions.

Distributions of softmax attentions. ViTs capture correlations between tokens through softmax attentions. The distribution of attention values varies drastically across different tokens (Fig. 3). Therefore, using a single quantization

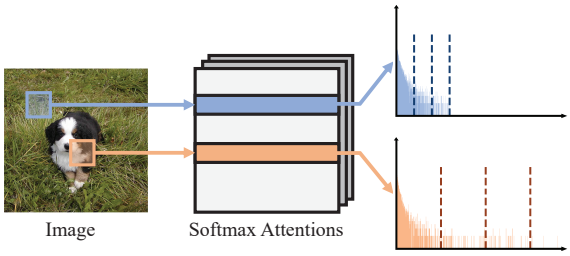


Figure 3. Distributions of softmax attentions across tokens. We can see that the distributions are different significantly across tokens. Our approach can handle this issue by splitting the rows of softmax attentions into several groups and applying separate quantizers for each group, such that the attentions assigned to each group share similar statistical properties.

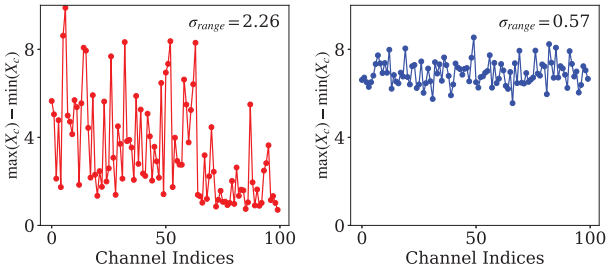


Figure 4. Comparisons for dynamic ranges of activation values across channels, chosen from different layers of ViT-S [10]. σ_{range} is the standard deviation of the dynamic ranges of channels for each layer. We can see that the degree of scale variations across channels varies according to the layer, suggesting that the number of groups for each layer would be adjusted.

parameter to quantize softmax attentions degrades quantization performance severely. Separate quantizers might be exploited for individual tokens to handle the attention values, but this requires a large number of quantizers, and needs to adjust the quantization parameters for each instance.

Instance-aware grouping across tokens. We extend our approach in order to quantize softmax attentions across tokens. We split softmax attentions for tokens (*i.e.*, the rows of the softmax attentions) into G_2 groups, according to the distribution of attention values. Specifically, given softmax attentions $\mathbf{A} \in \mathbb{R}^{H \times N \times N}$ and a set of quantizers $\{Q_j\}_{j=1}^{G_2}$, where we denote by H the number of heads, we define the distance between each row of softmax attentions and a quantizer Q_j as follows:

$$d(\mathbf{A}_n, Q_j) = (\max(\mathbf{A}_n) - v_j)^2 \tag{8}$$

where we denote by $\mathbf{A}_n$ and v_j the n-th row of $\mathbf{A}$ and the upper bound of the quantizer Q_j, respectively. Note that we set lower bounds to 0, as all attention values are positive. We then optimize v_j with the EM algorithm and set the quantization parameters for quantizer Q_j using Eq. (2).

Algorithm 1 IGQ-ViT

Hyperparameter: Number of iterations N_{iter}; update period of group size T.
Input: Pre-trained model; calibration set; target BOPs N_{bop}.
For inputs of FC layers and softmax attentions, compute the distance between their channels/rows and quantizers using Eq. (4) or Eq. (8).
for $k = 1, ..., N_{\text{iter}}$ **do**
  Update the parameters for each group using Eq. (5) and Eq. (6).
  if $k \,\%\, T == 0$ **then**
    Update the group size for each layer using Eq. (10).
  end if
end for
Obtain quantization parameters $\mathbf{s}$, $\mathbf{z}$ using Eq. (2).
Output: Quantized model

3.3. Group size allocation

We observe that activations and softmax attentions in different layers show different amount of scale variations across channels and tokens, respectively, indicating that using the same number of groups for different layers might be suboptimal (Fig. 4). To address this, we search for the optimal group size for each layer that minimizes the discrepancy between the predictions from quantized and full-precision models, under a BOP constraint. However, the search space for finding the optimal group sizes is exponential w.r.t. the number of layers L, which is intractable for a large model. We propose a group size allocation technique that efficiently optimizes the group size for each layer within such a large search space. Concretely, we define a perturbation metric for a particular layer $\psi(.)$ as the Kullback-Leibler (KL) divergence between predictions of the model before and after quantization as follows:

$$\psi(g, l) = D_{\text{KL}}(y_l || y_l^g), \tag{9}$$

where we denote by y_l and y_l^g the predictions of the model before and after quantizing l-th layer with a group size of g, respectively. Note that we quantize all other layers except for the l-th layer for computing the predictions of y_l and y_l^g, to account for the effects of quantization on different layers. For a target BOP N_{bop}, we formulate the group size allocation as a integer linear programming (ILP) problem, and search for the optimal group size for each layer, such that an overall perturbation of the model is minimized as follows:

$$\mathbf{g}^* = \arg\min_{\mathbf{g}} \sum_{l=1}^{L} \psi(g_l, l) \text{ s.t. } B(\mathbf{g}) \leq N_{bop}, \tag{10}$$

where $\mathbf{g} = \{g_l\}_{l=1}^{L}$, and g_l is the group size assigned to l-th layer. We denote by $B(\mathbf{g})$ the BOP of the model with the

Table 2. Quantitative results of quantizing ViT architectures on ImageNet [8]. W/A represents the bit-width of weights (W) and activations (A), respectively. We report the top-1 validation accuracy (%) with different group sizes for comparison. The numbers of other quantization methods are taken from [9, 20]. †: Results without using a group size allocation (*i.e.*, a fixed group size for all layers).

Method	#bits(W/A)	ViT-T	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S	Swin-B
Full-precision	32/32	75.47	81.39	84.54	72.21	79.85	81.80	81.39	83.23	85.27
PTQ4ViT [39]	4/4	17.45	42.57	30.69	36.96	34.08	64.39	-	76.09	74.02
APQ-ViT [9]	4/4	17.56	47.95	41.41	47.94	43.55	67.48	-	77.15	76.48
RepQ-ViT [20]	4/4	-	65.05	68.48	57.43	69.03	75.61	-	79.45	78.32
IGQ-ViT† (#groups=8)	4/4	54.46	72.99	77.91	61.94	74.01	78.85	76.91	80.54	82.88
IGQ-ViT† (#groups=12)	4/4	**55.66**	**73.46**	**79.13**	**62.28**	**74.52**	**79.21**	**77.53**	**80.79**	**83.08**
IGQ-ViT (#groups=8)	4/4	55.29	73.18	78.28	62.25	74.23	79.04	77.19	80.66	83.02
IGQ-ViT (#groups=12)	4/4	**56.01**	**73.61**	**79.32**	**62.45**	**74.66**	**79.23**	**77.77**	**80.98**	**83.14**
IGQ-ViT (upper bound)	4/4	57.59	74.73	79.88	62.55	75.08	79.74	78.58	81.48	83.53
PTQ4ViT [39]	6/6	64.46	78.63	81.65	69.68	76.28	80.25	-	82.38	84.01
APQ-ViT [9]	6/6	69.55	79.10	82.21	70.49	77.76	80.42	-	82.67	84.18
RepQ-ViT [20]	6/6	-	80.43	83.62	70.76	78.90	81.27	-	82.79	84.57
IGQ-ViT† (#groups=8)	6/6	72.90	80.07	83.11	70.71	78.92	81.34	80.23	82.55	84.43
IGQ-ViT† (#groups=12)	6/6	**73.63**	**80.66**	**83.63**	**71.02**	**79.17**	**81.48**	**80.59**	82.66	**84.70**
IGQ-ViT (#groups=8)	6/6	73.19	80.48	83.46	70.92	79.04	81.44	80.48	82.65	84.62
IGQ-ViT (#groups=12)	6/6	**73.77**	**80.76**	**83.77**	**71.15**	**79.28**	**81.71**	**80.89**	**82.86**	**84.82**
IGQ-ViT (upper bound)	6/6	74.61	80.99	84.27	71.42	79.42	81.75	81.20	83.08	85.06

group sizes of $\mathbf{g}$. We solve Eq. (10) with the PULP [29] library, using group size for each layer within the set of $\{4, 6, 8, 10, 12, 16\}$. We allocate the group sizes for every T alternating steps of Eq. (5) and Eq. (6), where T is a hyper-parameter. We show in Algorithm 1 an overall quantization process of our approach.

4. Experiments

In this section, we describe our experimental settings (Sec. 4.1), and evaluate IGQ-ViT on image classification, object detection and semantic segmentation (Sec. 4.2). We then present a detailed analysis of our approach (Sec. 4.3).

4.1. Implementation details

We evaluate our IGQ-ViT framework on the tasks of image classification, object detection, and instance segmentation. We use the ImageNet [8] dataset for image classification, which contains approximately 1.2M images for training, and 50K for validation. We use COCO [21] for object detection and instance segmentation, which includes 118K training, 5K validation, and 20K test images. We adopt various transformer architectures, including ViT [10], DeiT [33], and Swin transformer [24], for image classification. For the tasks of object detection and instance segmentation, we use Mask R-CNN [14] and Cascade Mask R-CNN [5] with Swin transformers as the backbone. Following [9, 20], we randomly sample 32 images from the ImageNet [8] dataset for image classification, and a single image from COCO [21] for object detection and instance segmentation to calibrate the quantization parameters. We apply our instance-aware grouping technique for all input activations of FC layers, and softmax attentions. More detailed settings are available in the supplement.

4.2. Results

Results on ImageNet. We show in Table 2 the top-1 accuracy (%) on the validation split of ImageNet [8] with various ViT architectures. We report the accuracy with an average group size of 8 and 12. We summarize our findings as follows: (1) Our IGQ-ViT framework with 8 groups already outperforms the state of the art except for ViT-B [10] and Swin-S [24] under 6/6-bit setting, while using more groups further boosts the performance. (2) Our approach under 4/4-bit setting consistently outperforms RepQ-ViT [20] by a large margin. Similar to ours, RepQ-ViT also addresses the scale variations between channels, but it can be applied to the activations with preceding LayerNorm only. In contrast, our method handles the scale variations on all input activations of FC layers and softmax attentions, providing better results. (3) Our group size allocation technique boosts the quantization performance for all models, indicating that using the same number of groups for all layers is suboptimal. (4) Exploiting 12 groups for our approach incurs less than 0.9% accuracy drop, compared to the upper bound under the 6/6-bit setting. Note that the results of upper bound are obtained by using a separate quantizer for each channel of activations and each row of softmax attentions.

Results on COCO. We show in Table 3 the quantization results for object detection and instance segmentation on

Table 3. Quantitative results of quantizing Mask R-CNN [14] and Cascade Mask R-CNN [5] using Swin transformers [24] on COCO [21]. We report the box average precision AP^{box} for object detection and the mask average precision AP^{mask} for instance segmentation.

Method	#bits(W/A)	Mask R-CNN				Cascade Mask R-CNN					
		Swin-T		Swin-S		Swin-T		Swin-S		Swin-B	
		AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-precision	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0	51.9	45.0
PTQ4ViT [39]	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5	10.6	9.3
APQ-ViT [9]	4/4	23.7	22.6	44.7	40.1	27.2	24.4	47.7	41.1	47.6	41.5
RepQ-ViT [20]	4/4	36.1	36.0	44.2	40.2	47.0	41.4	49.3	43.1	-	-
IGQ-ViT (#groups=8)	4/4	40.5	38.5	44.7	**41.3**	48.4	42.3	**50.5**	**44.0**	50.4	**44.0**
IGQ-ViT (#groups=12)	4/4	**41.0**	**38.8**	**44.8**	**41.3**	**48.5**	**42.4**	**50.5**	**44.0**	**50.5**	**44.0**
PTQ4ViT [39]	6/6	5.8	6.8	6.5	6.6	14.7	13.6	12.5	10.8	14.2	12.9
APQ-ViT [9]	6/6	45.4	41.2	47.9	42.9	48.6	42.5	50.5	43.9	50.1	43.7
RepQ-ViT [20]	6/6	45.1	41.2	47.8	43.0	50.0	43.5	51.4	44.6	-	-
IGQ-ViT (#groups=8)	6/6	45.4	**41.5**	**48.2**	43.1	**50.4**	43.7	**51.9**	44.9	**51.9**	**45.0**
IGQ-ViT (#groups=12)	6/6	**45.5**	**41.5**	**48.2**	**43.2**	**50.4**	**43.8**	**51.9**	**45.0**	**51.9**	**45.0**

Table 4. Quantitative comparison of our instance-aware group quantization technique with various configurations under a 4/4-bit setting. We denote by 'Linear' and 'Attention' the quantization method for linear operations and softmax attentions, respectively. For applying our method, we use a group size of 8 for all layers.

Linear	Attention	ViT-S	DeiT-B	Swin-T
Layer-wise	Layer-wise	42.82	62.23	55.51
Layer-wise	Ours	53.69	65.05	62.27
Ours	Layer-wise	57.32	75.51	72.89
Ours	Ours	**72.99**	**78.85**	**76.91**
Ours	Row-wise	73.22	78.88	77.19
Channel-wise	Ours	74.56	79.69	78.39
Channel-wise	Row-wise	**74.73**	**79.74**	**78.58**

Table 5. Quantitative comparison of quantizing transformer architectures using various group quantization techniques under a 4/4-bit setting, with a group size of 8 for all linear operations. Note that we use layer-wise quantization for softmax attentions for a fair comparison.

Grouping methods	ViT-S	DeiT-B	Swin-T
No grouping (#groups=1)	42.82	62.23	55.51
Grouping consecutive channels [7, 31]	41.04	65.50	70.39
Sorting before grouping channels [4]	41.26	62.61	56.04
Ours	**57.32**	**75.51**	**72.89**

COCO [21]. We quantize the backbones of Swin transformers [24] and the convolutional layers in the neck and head of Mask R-CNN [14] and Cascade Mask R-CNN [5]. We observe that PTQ4ViT [39] and APQ-ViT [9], that use layer-wise quantizers for activations, do not perform well. In contrast, IGQ-ViT outperforms the state of the art with 8 groups only, and the quantization performance further boosts by exploiting more groups. In particular, it provides the results nearly the same as the full-precision ones for the the 6/6-bit setting. This suggests that scale variations across different channels or tokens are much more critical for object detection and instance segmentation.

4.3. Discussion

Comparison with different quantizers. We compare in Table 4 the results of the variants of our method adopting different types of quantizers on input activations of FC layers and softmax attentions. From the first four rows, we can see that our approach outperforms layer-wise quantization by a large margin, both for linear operations and softmax attentions. This indicates that adopting a single quantization parameter for all channels and rows without considering their individual distributions can severely limit the quantization performance. The last three rows compare the results of our approach with channel/row-wise quantization. We observe that the difference in performance between our approach and channel/row-wise quantization is less than 1.8% for three different models. With a small group size, our framework can achieve comparable performance to the upper bound, while maintaining efficiency.

Table 5 shows the results of quantizing ViT architectures using various group quantization techniques, including [4, 7, 31], and ours. While the works of [7, 31] divide consecutive channels uniformly into a number of groups, the method of [4] first sorts channels w.r.t. the dynamic ranges before partitioning them into groups. In contrast, we dynamically assign channels to groups according to the statistical properties of the channels. We find that our approach outperforms other methods by a large margin, indicating that fixing the channels assigned to each group can degrade the quantization performance significantly. We also observe that sorting the channels w.r.t. their dynamic ranges during calibration does not boost the quantization performance for DeiT-B [33] and Swin-T [24], suggesting that the dynamic range of each channel vary drastically across different input instances.

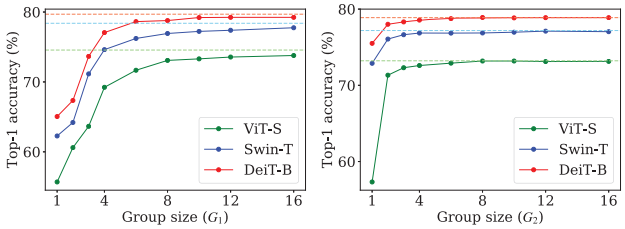


Figure 5. Top-1 validation accuracies on ImageNet [8] w.r.t. group sizes for linear operations (G_1, left) and that for softmax attentions (G_2, right). We set either G_1 or G_2 to 8, while varying the other to compute the accuracies. We report the quantization results of ViT-S [10], Swin-T [24], and DeiT-B [33] under a 4/4-bit setting, with a fixed group size across different layers. We visualize the upper bounds with horizontal stripes of corresponding colors.

Table 6. Quantitative comparison of ViT architectures with and without a group size allocation technique under the 4/4-bit setting.

#groups	Group size allocation	ViT-S	DeiT-B	Swin-T
6		71.44	78.44	76.31
	✓	**71.80**	**78.57**	**76.99**
8		72.99	78.85	76.91
	✓	**73.18**	**79.04**	**77.19**
10		73.34	79.00	77.21
	✓	**73.51**	**79.19**	**77.64**

Analysis on group size. We show in Fig. 5 the results of IGQ-ViT according to the group size for linear operations (left) and softmax attentions (right). We can see that the quantization performance improves as the group size increases, for both linear operations and softmax attentions, demonstrating that using more groups better addresses the scale variation problem for channels and tokens. We also observe that the performance of our approach reaches near the upper bound with a small group size. This suggests that IGQ-ViT can effectively address the variations with a small amount of additional computations.

Convergence analysis. We compare in Fig. 6(top) distances between channels of activation and quantizers in Eq. (4) (rows of softmax attention and quantizers in Eq. (8)) over optimization steps. It shows that our algorithm converges quickly within a small number of optimization steps. We show in Fig. 6(bottom) the dynamic ranges of activations and attentions in a particular layer, along with their assigned groups after convergence. We can see that activations/attentions in each group share similar statistical properties, demonstrating that they can be effectively quantized with a single quantization parameter.

Group size allocation. We compare in Table 6 the results of our approach with/without the group size allocation technique. We can see that the group size allocation improves the quantization performance consistently, suggesting that assigning the same group size for all layers is suboptimal.

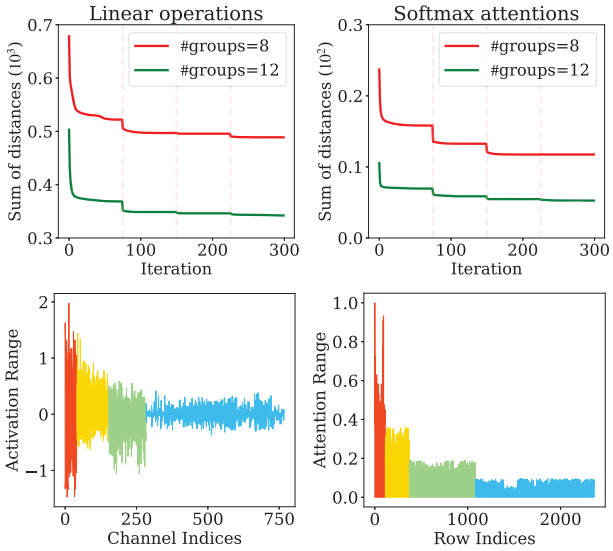


Figure 6. (Top) Distances between channels of activations/rows of softmax attentions and quantizers in a particular layer of DeiT-S [33], that is, the distances as in Eq. (4) and Eq. (8), respectively; (Bottom) Dynamic ranges of activations and attentions in a specific layer of DeiT-S w.r.t. the assigned group of the same color. Note that we sort the activations and attentions based on the group indices for the purpose of the visualization only.

5. Conclusion

We have observed that activations and softmax attentions in ViTs have significant scale variations for individual channels and tokens, respectively, across different input instances. Based on this, we have introduced a instance-aware group quantization framework for ViTs, IGQ-ViT, that alleviates the scale variation problem across channels and tokens. Specifically, our approach splits the activations and softmax attentions dynamically into multiple groups along the channels and tokens, such that each group shares similar statistical properties. It then applies separate quantizers for individual groups. Additionally, we have present a simple yet effective method to assign a group size for each layer adaptively. We have shown that IGQ-ViT outperforms the state of the art, using a small number of groups, with various ViT-based architectures. We have also demonstrated the effectiveness of IGQ-ViT compared with its variants, including layer-wise quantizers, channel/row-wise quantizers, and state-of-the-art group quantizers, with a detailed analysis.

Acknowledgements. This work was supported in part by the NRF and IITP grants funded by the Korea government (MSIT) (No.2023R1A2C2004306, No.RS-2022-00143524, Development of Fundamental Technology and Integrated Solution for Next-Generation Automatic Artificial Intelligence System, and No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv*, 2016. 3

[2] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *NeurIPS*, 2019. 2, 3

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv*, 2013. 2

[4] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. *EMNLP*, 2021. 3, 4, 7

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 6, 7

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

[7] Steve Dai, Rangha Venkatesan, Mark Ren, Brian Zimmer, William Dally, and Brucek Khailany. Vs-quant: Per-vector scaled quantization for accurate low-precision neural network inference. *Proceedings of Machine Learning and Systems*, 2021. 1, 2, 3, 4, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 3, 6, 8

[9] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *ACMMM*, 2022. 2, 6, 7

[10] 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. In *ICLR*, 2021. 1, 2, 5, 6, 8

[11] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *ICLR*, 2020. 1, 2

[12] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. Post-training piecewise linear quantization for deep neural networks. In *ECCV*, 2020. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 4

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 6, 7

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[16] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *CVPR*, 2019. 2

[17] Dohyung Kim, Junghyup Lee, and Bumsub Ham. Distance-aware quantization. In *ICCV*, 2021. 2

[18] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *CVPR*, 2021. 2

[19] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021. 1, 2, 3

[20] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. *ICCV*, 2023. 2, 3, 6, 7

[21] 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*, 2014. 2, 6, 7

[22] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Fully quantized vision transformer without retraining. In *IJCAI*, 2022. 2, 3

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2018. 3

[24] 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 *ICCV*, 2021. 1, 2, 6, 7, 8

[25] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS*, 2021. 2, 3

[26] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *ICML*, 2020. 1, 2, 3

[27] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 2021. 2

[28] Eunhyeok Park and Sungjoo Yoo. Profit: A novel training method for sub-4-bit mobilenet models. In *ECCV*, 2020. 2

[29] JS Roy and SA Mitchell. Pulp is an lp modeler written in python. 2020. 6

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2, 4

[31] 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*, 2020. 1, 2, 3, 4, 7

[32] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 1

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 3, 4, 6, 7, 8

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[35] Ziwei Wang, Changyuan Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. Quantformer: Learning extremely low-precision vision transformers. *TPAMI*, 2022. 3, 4

[36] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2022. 1

[37] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, 1983. 4

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 1

[39] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization framework for vision transformers. In *ECCV*, 2022. 2, 3, 6, 7

[40] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo: Transformer-based yolo for object detection. In *ICCV*, 2021. 1

[41] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. In *ICLR*, 2017. 1, 2

[42] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *CVPR*, 2018. 1, 2