PACK-PTQ: ADVANCING POST-TRAINING QUANTIZATION OF NEURAL NETWORKS BY PACK-WISE RECONSTRUCTION

Changjun Li, Runqing Jiang, Zhuo Song, Pengpeng Yu, Ye Zhang, Yulan Guo
Shenzhen Campus, Sun Yat-sen University
Aviation University of Air Force
{lichj68, jiangrq3}@mail2.sysu.edu.cn

ABSTRACT

Post-training quantization (PTQ) has evolved as a prominent solution for compressing complex models, which advocates a small calibration dataset and avoids end-to-end retraining. However, most existing PTQ methods employ block-wise reconstruction, which neglects cross-block dependency and exhibits a notable accuracy drop in low-bit cases. To address these limitations, this paper presents a novel PTQ method, dubbed Pack-PTQ. First, we design a Hessian-guided adaptive packing mechanism to partition blocks into non-overlapping packs, which serve as the base unit for reconstruction, thereby preserving the cross-block dependency and enabling accurate quantization parameters estimation. Second, based on the pack configuration, we propose a mixed-precision quantization approach to assign varied bit-widths to packs according to their distinct sensitivities, thereby further enhancing performance. Extensive experiments on 2D image and 3D point cloud classification tasks, using various network architectures, demonstrate the superiority of our method over the state-of-the-art PTQ methods.

1 Introduction

The remarkable progress of neural networks has propelled significant breakthroughs in computer vision, with notable achievements in image classification [1, 2], object detection [3, 4, 5], and semantic segmentation [6, 7]. However, despite their impressive performance, deploying neural networks on resource-constrained edge devices (e.g., smartphones) remains a significant challenge due to their substantial computational and memory requirements. To handle this, model compression techniques have gained significant attention recently, which aim to compress and accelerate complex neural networks without compromising accuracy.

The vast majority of model compression techniques can be divided into four categories: compact architectures [8, 3], network pruning [9, 10], knowledge distillation [11, 12], and model quantization [13, 14], in which model quantization draws particular attention for its ability to not only reduce model size but also accelerate inference by limiting bit precision. A line of works [15, 16] adopt quantization during model training, which is called quantization-aware training (QAT). While QAT methods achieve high performance, they necessitate end-to-end retraining on the entire training set, resulting in substantial time and computational consumption. To circumvent the heavy retraining, post-training quantization (PTQ) has garnered increasing interest in recent times. PTQ-based methods [14, 17, 18] perform quantization on pre-trained models using a small-scale calibration set, thereby eliminating the need for end-to-end retraining.

Extensive PTQ-based studies [14, 19, 18] have focused on the reconstruction procedure, which seeks to align the outputs between a quantized model and its full-precision counterpart, to recover accuracy after quantization. However, due to the limited availability of calibration samples, traditional network-wise reconstruction often suffers from overfitting, leading to subpar test accuracy. To overcome this limitation, recent studies [14, 19] have proposed employing reconstruction in a block-wise manner, which has been shown to yield improved performance. Unfortunately, as indicated in Fig. 1, these methods still exhibit limited performance in the case of ultra-low bit quantization, as they frequently underestimate the significance of cross-block dependency, resulting in inaccurate quantization parameters.

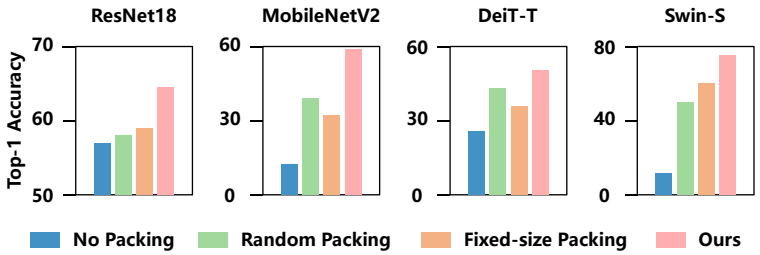

Figure 1: Quantization results of different reconstruction strategies on ImageNet with W3/A3 setting. "No Packing" means employing block-wise reconstruction, "Random Packing" means randomly assigning blocks into packs, and "Fixed-size Packing" means assigning blocks into packs with equal size.

Therefore, it is essential to revisit the reconstruction granularity of quantization, seeking an optimal balance between capturing cross-block dependencies and maintaining optimization efficacy.

To tackle these challenges, this paper introduces Pack-PTQ, a novel post-training quantization method designed to effectively quantize neural networks, even in low-bit scenarios, as illustrated in Fig. 1. First, we propose a Hessian-guided adaptive packing mechanism, which adaptively clusters blocks into distinct packs, serving as the base unit for reconstruction. Specifically, this mechanism starts by calculating a Hessian-guided importance score to assess the impact of each block on its former modules, and then clusters blocks into packs with the guidance of these scores. Second, we recognize that different packs exhibit varied sensitivities to quantization, and thus develop a pack-based mixed-precision quantization strategy. This strategy assigns different bit-widths to different packs, demonstrating improved performance compared to unified precision quantization.

The main contributions of this work are summarized in the following four aspects:

- We propose Pack-PTQ, a novel solution that combines a Hessian-guided adaptive packing mechanism with a pack-based mixed-precision quantization strategy, enabling high-performance post-training quantization of neural networks, particularly in low-bit cases.
- We devise a Hessian-guided adaptive packing mechanism to divide a neural network into non-overlapping packs, which provides a new reconstruction granularity, effectively capturing cross-block dependency.
- With the pack configuration, we develop a pack-based mixed-precision strategy that takes into account the distinct sensitivities of each pack to quantization, thereby enabling a more efficient allocation of bits.
- Through a comprehensive experimental evaluation on 2D image and 3D point cloud classification tasks, leveraging diverse neural network architectures, we demonstrate the consistent superiority of Pack-PTQ over several state-of-the-art PTQ methods, showcasing its robustness and effectiveness in real-world applications.

2 Related Works

2.1 Post-training Quantization

Model quantization is a technique aiming at accelerating inference and reducing memory footprint by limiting the bit-widths of network parameters [13]. A traditional approach to mitigating the accompanying accuracy loss is to perform end-to-end retraining of the model, a method known as quantization-aware training (QAT) [15, 20, 21]. While QAT-based methods achieve high accuracy, they raise significant concerns regarding privacy and incur substantial time overhead due to the extensive retraining process.

To address the challenges above, post-training quantization (PTQ) has emerged as a preferred method for generating quantized models without requiring extensive retraining, garnering significant attention in the realm of model compression. A notable advancement in PTQ is BRECQ [14], which effectively addresses overfitting through block-wise reconstruction. By partially reproducing the Hessian matrix of the original network to serve as a regularizer, BRECQ achieves improved performance in PTQ, making it a promising approach for efficient model compression. Subsequent research has continued to advance the field of PTQ. Q-Drop [22] enhances model robustness by integrating quantized and original data, while NoisyQuant [23] introduces random noise to reduce reliance on fully quantized data, thereby preserving the connectivity of the original model. More recently, SPQ [24] has made significant contributions to this field by leveraging dataset interconnections to improve quantization, particularly in scenarios with limited calibration data. However, these methods have primarily focused on local data connections, potentially overlooking global interdependencies within the network. To address this gap, PD-Quant [25] has introduced global monitoring mechanisms

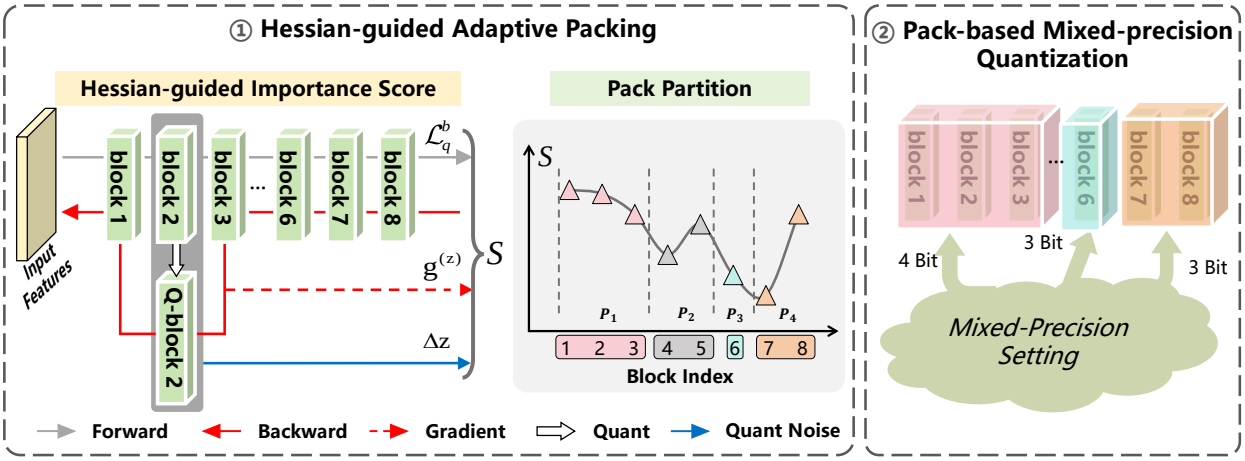

Figure 2: Overview of our proposed Pack-PTQ method. We begin by computing individual block scores, which take into account the model loss, block output gradients, and block local loss. Subsequently, we employ our novel packing mechanism to cluster these blocks into packs. Finally, we assign diverse bit-widths to each pack, thereby achieving optimal pack reconstruction and facilitating efficient post-training quantization of neural networks.

that enhance the accuracy of compressed models. Despite these advancements, challenges persist in fully capturing cross-block relationships, highlighting the ongoing need for improved optimization strategies in PTQ.

2.2 Mixed-precision Quantization

It is commonly acknowledged that quantizing neural networks to 8-bit formats has achieved remarkable results with minimal accuracy degradation [26, 17]. However, quantizing to lower bit-widths (i.e., 3-bit) still remains challenging because of significant accuracy loss. To overcome this hurdle, mixed-precision quantization has emerged as a promising solution, which involves assigning different bit-widths to various layers or components of the models [27]. This approach has shown great potential in balancing accuracy and computational efficiency.

One notable approach in this domain is HAWQ [28], which introduces an automated method to determine optimal mixed-precision settings based on layer sensitivity derived from the Hessian spectrum. Building upon this, HAWQ-V2 enhances this method by using the average Hessian trace as a more accurate sensitivity metric compared to the top eigenvalue [29]. Additionally, HAWQ-V2 employs an automated Pareto frontier method for layer-wise bit precision selection and extends Hessian-based analysis to mixed-precision activation quantization. A recent study further contributes to this field by proposing a novel framework that formulates mixed-precision quantization as a discrete constrained optimization problem [27]. This framework introduces a Multiple-Choice Knapsack Problem (MCKP), which is solved using a greedy search algorithm to optimize bit-widths assignments. This approach provides a systematic way to balance the trade-off between model accuracy and computational efficiency. Moreover, APTQ takes into account the nonlinear effects of attention outputs on the entire model, providing a more nuanced approach to quantization [30]. By considering the impact of attention mechanisms, APTQ offers a refined method for determining the optimal bit-widths.

3 Preliminaries

3.1 Quantization

The uniform quantizer is widely adopted due to its superior hardware compatibility, which converts input full-precision numbers to several fixed-points:

$$x_q = \text{clamp}\left(\left\lfloor \frac{x}{s} + z \right\rceil, 0, 2^k - 1\right), \tag{1}$$

where x and x_q stand for the input full-precision number and its quantized value, respectively. The s, z, and k respectively denote the quantization scale, zero point, and bit-widths. $\lfloor \cdot \rceil$ denotes the rounding operation. In this paper, following BRECQ [14], we employ uniform quantizers for both weights and activations.

3.2 Taylor Expansion

In line with previous efforts [14, 19], quantization imposed on weights can be regarded as a specific instance of weight perturbation. Therefore, the loss degradation caused by quantization can be approximated by Taylor series expansions, which is formulated as:

$$\mathcal{L}_q = \mathbb{E}[\mathcal{L}(\mathbf{w} + \Delta\mathbf{w})] - \mathbb{E}[\mathcal{L}(\mathbf{w})] \approx \Delta\mathbf{w}^\top \mathbf{g}^{(\mathbf{w})} + \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}^{(\mathbf{w})}\Delta\mathbf{w}, \tag{2}$$

where $\mathcal{L}$ is task-specific loss, $\Delta\mathbf{w}$ represents the noise introduced by quantization, $\mathbf{g}^{(\mathbf{w})}$ is the first-order derivative of the loss function. $\mathbf{H}^{(\mathbf{w})}$ is the second-order derivative of the loss function, known as the Hessian matrix.

4 Method

4.1 Hessian-guided Adaptive Packing Mechanism

Block reconstruction strategy resorts to minimizing the quantization errors by aligning the outputs of the full-precision blocks and their quantized counterparts. This paradigm approximates the Hessian matrix with a block-diagonal matrix, in which off-block entries are set to zero. Despite its efficiency, such a skewed estimation of Hessian underestimates the hidden relationships across different blocks, ultimately leading to inferior reconstruction results. To illustrate this, we modify the reconstruction unit from a single block to a stack of blocks (denoted as packs), which allows us to capture the inter-block dependencies within a pack. As shown in Fig. 1, such a simple modification largely outperforms the widely used block-wise paradigm, which indicates the significance of exploring the cross-block relationships. In addition, the final performance differs significantly with various packing strategies (i.e., random packing and fixed-size packing), which motivates us to automate the packing configurations.

To achieve this, we develop a block importance metric that ranks the blocks and subsequently packs consecutive blocks based on this metric. The cornerstone of our approach is to assess the impact of each block's output on the preceding blocks. In this paper, we propose utilizing the Hessian matrix to form this metric, which inherently captures the second-order information of the proceeding modules. Specifically, we first rewrite Eq. (2) to adapt to a single block's output:

$$\mathcal{L}_q^b = \mathbb{E}[\mathcal{L}(\mathbf{z} + \Delta\mathbf{z})] - \mathbb{E}[\mathcal{L}(\mathbf{z})] \approx \Delta\mathbf{z}^\top \mathbf{g}^{(\mathbf{z})} + \frac{1}{2}\Delta\mathbf{z}^\top \mathbf{H}^{(\mathbf{z})}\Delta\mathbf{z}, \tag{3}$$

where $\mathbf{z}$ and $\Delta\mathbf{z}$ represent the output of a block and the change of $\mathbf{z}$ incurred by quantization, respectively. To fully consider the contributions of all elements in the Hessian matrix, we adopt the average value of the Hessian matrix as the block importance metric. However, directly computing $\mathbf{H}^{(\mathbf{z})}$ is infeasible due to the computational complexity of neural networks. Alternatively, we approximate the mean of the Hessian matrix, as detailed below.

Theorem 1. *Assume that $\Delta\mathbf{z}$ is a random vector where each component is i.i.d. as $\mathcal{N}(0, \sigma^2)$. Then the expectation $\mu_{\mathbf{H}^{(\mathbf{z})}}$ of all the elements in the Hessian matrix $\mathbf{H}^{(\mathbf{z})}$ can be approximated as:*

$$\mu_{\mathbf{H}^{(\mathbf{z})}} \approx \frac{\mathbb{E}[\Delta\mathbf{z}^\top \mathbf{H}^{(\mathbf{z})}\Delta\mathbf{z}]}{\mathbb{E}[\Delta\mathbf{z}^\top\Delta\mathbf{z}]}. \tag{4}$$

Proof. Utilizing the knowledge of expectations of quadratic forms and the properties of the covariance matrix of random vectors, we have:

$$\mathbb{E}[\Delta\mathbf{z}^\top \mathbf{H}^{(\mathbf{z})}\Delta\mathbf{z}] = \text{tr}(\mathbf{H}^{(\mathbf{z})}\mathbb{E}[\Delta\mathbf{z}\Delta\mathbf{z}^\top]) = \text{tr}(\mathbf{H}^{(\mathbf{z})}\sigma^2 I) \approx \mu_{\mathbf{H}^{(\mathbf{z})}}\sigma^2 n. \tag{5}$$

At the same time, the $\mathbb{E}[\Delta\mathbf{z}^\top\Delta\mathbf{z}]$ can be concluded as:

$$\mathbb{E}[\Delta\mathbf{z}^\top\Delta\mathbf{z}] = \text{tr}(\mathbb{E}[\Delta\mathbf{z}\Delta\mathbf{z}^\top]) = \text{tr}(\sigma^2 I) = \sigma^2 n. \tag{6}$$

Finally, combining Eq.(5) and Eq.(6), the $\mu_{\mathbf{H}^{(\mathbf{z})}}$ can be obtained by:

$$\mu_{\mathbf{H}^{(\mathbf{z})}} = \frac{\mu_{\mathbf{H}^{(\mathbf{z})}}\sigma^2 n}{\sigma^2 n} \approx \frac{\mathbb{E}[\Delta\mathbf{z}^\top \mathbf{H}^{(\mathbf{z})}\Delta\mathbf{z}]}{\mathbb{E}[\Delta\mathbf{z}^\top\Delta\mathbf{z}]}. \tag{7}$$

□

According to Eq.(3) and Eq.(7), we obtain the importance score for each block as:

$$S = \mu_{\mathbf{H}^{(\mathbf{z})}} \approx \frac{\mathbb{E}[2 \times (\mathcal{L}_q^b - \Delta \mathbf{z}^\top \mathbf{g}^{(\mathbf{z})})]}{\mathbb{E}[\Delta \mathbf{z}^\top \Delta \mathbf{z}]}. \tag{8}$$

Notably, this formulation simplifies the computation of the Hessian-based block importance metric to achieve efficiency. A low value of S indicates that perturbations to the corresponding block have minimal impact on subsequent blocks and even the final output.

After that, given a set of blocks $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ with their corresponding importance scores $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, we aim to assign different blocks to distinct packs. We initiate this process by setting two indicators, $t_s = 1$ and $t_e = n$, which define the search scope for the current pack. Then, we identify the block with the lowest importance score within this scope, whose index is denoted as $t_{min} \in [t_s, t_e]$. Based on the above requirements, a new pack is formed by grouping the blocks from t_{min} to t_e, which is formally defined as:

$$P = \left\{ \bigcup_{t=t_{\text{min}}}^{t_e} B_t \;\middle|\; t_{\text{min}} = \arg \min_{t_s \le t \le t_e} S[t] \right\} \tag{9}$$

where t is the index of a block, and $\mathcal{S}[t]$ denotes the t-th entry of $\mathcal{S}$. Once a pack is formed, the t_e is updated to $t_{min} - 1$, and the process is repeated until each block is meticulously assigned to a unique pack. This approach partitions the network into non-overlapping packs, where the first block in each pack is characterized by a lower importance score, reflecting its limited influence on the former modules. The blocks within the same pack are tightly coupled, implying they are highly interdependent. By jointly optimizing these blocks, we can derive more accurate quantization parameters, as the correlated behaviors within each pack are effectively captured. We also explore diverse packing partition strategies, which are detailed in Sec. 5.4.

4.2 Pack-based Mixed-precision Quantization

As previous studies [28, 27] have shown, different layers exhibit varying sensitivities to quantization, implying that a unified bit-widths assignment across all layers is suboptimal. Specifically, some blocks are relatively insensitive to low-bit quantization and can tolerate aggressive compression, while others require higher precision to maintain accuracy. Motivated by this, we propose a novel approach that assigns different bit-widths to different packs, striking a balance between performance and memory footprint. To be specific, assume that the constraint of memory footprint is C and the pre-defined candidate bit-widths set is $\mathcal{K} = \{k_1, k_2, ..., k_n\}$, and our goal is to determine the optimal bit-widths configuration that satisfies the memory constraint while maximizing model accuracy. To achieve this, we formulate the problem as an optimization objective, which can be mathematically expressed as:

$$\max_{\{b_1, b_2, \ldots, b_M\}} \sum_{j=1}^{M} b_j \cdot \Omega_j, \quad \text{s.t.} \quad \sum_{j=1}^{M} b_j \cdot p_j \le C, \tag{10}$$

where M and p_j denote the number of packs and parameters in the j-th pack, respectively. The $b_j \in \mathcal{K}$ is the bit-width for the j-th pack, and the Ω_j is the mean sensitivity for the j-th pack, which is defined as:

$$\Omega_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (S_j[i] \cdot \mathcal{L}_j^q[i]), \tag{11}$$

where n_j denotes the number of blocks in j-th pack, $S_j[i]$ represents the i-th importance score of the block in the j-th pack, and $\mathcal{L}_j^q[i]$ denotes the quantization loss of the i-th block in the j-th pack. By solving Eq. (10), packs with higher sensitivity are allocated higher bit-widths, thereby preserving their accuracy, while packs with lower sensitivity are subjected to more aggressive quantization, thereby reducing memory footprint. Such a way enables flexible and efficient mixed-precision quantization, improving network performance while adhering to memory limitations.

4.3 Optimization

To recover the accuracy of quantized models, we leverage the packs obtained in Sec. 4.1 as the base unit for reconstruction. Specifically, we define a pack-wise reconstruction loss, which is formulated as:

$$\mathcal{L}_{rec} = \mathbb{E}[||P_q^{(l)}(x^{(l)}) - P^{(l)}(x^{(l)})||_{\text{F}}], \tag{12}$$

where $x^{(l)}$ is the input for the l-th pack and $||\cdot||_{\mathrm{F}}$ represents the Frobenius norm. $P_q^{(l)}(x^{(l)})$ and $P^{(l)}(x^{(l)})$ denote the outputs of the l-th quantized pack and full-precision pack, respectively. Notably, the reconstruction granularity can differ among packs since the packs have varied sizes, enabling the model to adaptively learn cross-block dependencies and capture nuanced relationships between blocks. Thus, such a manner facilitates the learning of more accurate quantization parameters, which is essential for preserving the model's accuracy during the quantization process.

5 Experiments

5.1 Experimental Setups

5.1.1 Datasets and Competing Methods

We conduct comprehensive experiments on image classification and point cloud classification tasks. For image classification, Pack-PTQ is evaluated on the ImageNet dataset [31], which consists of 1,000 categories and 50,000 test samples. We use both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) as network architectures to validate the effectiveness of our method. To be specific, in the case of CNNs, we compare Pack-PTQ with several state-of-the-art methods, including BRECQ [14], RAPQ [32], MRECG [33], and Genie [34]. For ViTs, the evaluation is conducted against BRECQ [14], QDrop [19], PTQ4ViT [17], PD-Quant [25], and RepQ-ViT [35]. As for the point cloud classification, we evaluate our Pack-PTQ on ModelNet40 [36] which consists of 40 categories with a test set of 2,468 samples. To the best of our knowledge, there are no previous works dedicated to post-training quantization on ModelNet40. Therefore, we reimplement the MinMax [18] and BRECQ [14] with their official codes as our competing methods.

5.1.2 Implementation Details

In our experiments, we utilize pre-trained full-precision transformer models from the Timm library[1] and CNN models from BRECQ [14]. We apply uniform quantization to all model weights and activations. We adopt the Adam optimizer with an initial learning rate of 4e-5, no weight decay, and a cosine decay schedule to adjust the learning rate. All experiments are conducted with a batch size of 32 across 2,000 training iterations, following the protocol outlined in prior work [37]. For the image classification task, Pack-PTQ is evaluated on two architecture families: CNNs and ViTs. For CNNs, the evaluation spans various architectures, including ResNet [38] (standard convolution), MobileNetV2 [8] (depthwise separable convolution), RegNet [39] (group convolution), and MNasNet [40]. For ViTs, it includes ViT [1], DeiT [2], and Swin Transformer [3]. For the point cloud classification task, evaluations are conducted exclusively on PointNet [41]. All experiments are performed using a single NVIDIA TITAN RTX GPU. The source code is released in our supplementary materials.

5.2 Results on Image Classification

Table 1 shows the test accuracy of various methods on ImageNet for different CNN models and bit-widths. As shown in Table 1, our Pack-PTQ consistently outperforms existing methods in both W3/A3 and W4/A4 quantization settings. Notably, without mixed-precision (w/o MP), Pack-PTQ surpasses Genie, the second-best performer, by 8.15% on MobileNetV2 under W3/A3 quantization. When using mixed-precision, the improvement increases to 12.86%, demonstrating the effectiveness of bit-width allocation. Furthermore, our method significantly outperforms BRECQ, which achieves an accuracy of only 12.27% under the same setting. The performance gap between our method and BRECQ is particularly pronounced on MobileNetV2, primarily due to the absence of cross-block dependency in BRECQ. Although our method ranks second on some architectures under W2/A4 quantization, the performance differences are minor. For instance, our Pack-PTQ achieves 61.29% accuracy on ResNet18, only 0.21% behind BRECQ, the best performer. However, our method demonstrates superior generalization performance across all architectures. Moreover, our results show minimal performance degradation when using lower bit-widths, highlighting the robustness of our Pack-PTQ.

We further evaluate our Pack-PTQ on several representative ViT models, as shown in Table 2. Overall, our Pack-PTQ achieves state-of-the-art performance in almost all cases, with the smallest gap with the full-precision model. Notably, in the W3/A3 setting, most competing methods exhibit significantly degraded performance compared to full-precision models. For instance, RepQ-ViT and PTQ4ViT achieve only 0.14% and 0.01% accuracy on ViT-B and DeiT-S, respectively, highlighting the challenges in quantizing ViTs in low-bit cases. In contrast, our method achieves accuracy of 61.65% and 50.31% without mixed-precision, and further reaches 64.83% and 58.19% with mixed-precision, respectively. Although Adalog outperforms Pack-PTQ in terms of W4/A4 quantization on ViT-S

[1] https://github.com/rwightman/pytorch-image-models

Table 1: Quantization results for CNNs on the ImageNet dataset. The top-1 accuracy (%) is reported. "Bit. (W/A)" represents the bit width for weights and activations. **Bold** font indicates the highest value. Underlined data represents the second highest value.

Method	Bit. (W/A)	ResNet18	ResNet50	MNV2	Reg600M	Reg3.2G	MNasx2
Full-Precision	32/32	71.08	77.00	72.49	73.71	78.36	76.68
BRECQ [14]	2/4	**61.50**	62.79	21.18	56.49	59.51	49.11
RAPQ [32]	2/4	61.18	59.70	34.47	54.80	57.47	-
MRECG [33]	2/4	59.19	57.96	47.08	26.26	-	-
Genie [34]	2/4	61.27	57.27	50.44	57.77	56.74	**62.24**
Pack-PTQ (Ours) w/o MP	2/4	61.29	**63.47**	**55.86**	**59.88**	**63.14**	59.17
BRECQ [14]	3/3	56.89	61.84	12.27	47.85	51.14	41.00
RAPQ [32]	3/3	55.44	58.71	7.33	44.94	51.04	-
MRECG [33]	3/3	60.01	66.82	48.87	57.75	-	-
Genie [34]	3/3	61.10	67.96	50.68	59.53	67.11	59.59
TexQ [42]	3/3	50.28	25.27	32.80	-	-	-
Pack-PTQ (Ours) w/o MP	3/3	64.46	70.34	58.83	65.12	70.81	62.71
Pack-PTQ (Ours) with MP	3/3	**66.73**	**72.14**	**63.54**	**68.67**	**72.88**	**66.76**
BRECQ [14]	4/4	68.20	73.20	61.93	68.91	74.09	69.57
RAPQ [32]	4/4	68.11	73.08	60.50	68.25	73.64	-
MRECG [33]	4/4	67.69	73.96	66.55	69.22	-	-
Genie [34]	4/4	67.80	74.07	66.85	69.50	75.23	69.83
TexQ [42]	4/4	67.73	70.72	67.07	-	-	-
Pack-PTQ (Ours) w/o MP	4/4	68.74	74.74	68.58	70.96	76.83	72.51
Pack-PTQ (Ours) with MP	4/4	**69.86**	**75.51**	**70.41**	**72.31**	**77.70**	**74.27**

Table 2: Quantization results for ViTs on the ImageNet dataset. The top-1 accuracy (%) is reported. "Bit. (W/A)" represents the bit width for weights and activations. **Bold** font indicates the highest value. Underlined data represents the second highest value.

Method	Bit. (W/A)	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
BRECQ [14]	3/3	0.42	0.59	25.52	14.63	46.29	11.67	1.70
QDrop [19]	3/3	4.44	8.00	30.73	22.67	24.37	60.89	54.76
PTQ4ViT [17]	3/3	0.01	0.01	0.04	0.01	0.27	0.35	0.29
PD-Quant [25]	3/3	1.77	13.09	39.97	29.33	0.94	69.67	64.32
RepQ-ViT [35]	3/3	0.43	0.14	0.97	4.37	4.84	8.84	1.34
Adalog [43]	3/3	10.32	30.10	23.06	19.61	53.33	54.66	62.38
Pack-PTQ (Ours) w/o MP	3/3	47.68	61.65	50.31	53.04	70.13	75.42	76.04
Pack-PTQ (Ours) with MP	3/3	**55.56**	**64.83**	**58.19**	**58.38**	**74.41**	**78.92**	**80.49**
BRECQ [14]	4/4	12.36	9.68	55.63	63.73	72.31	72.74	58.24
QDrop [19]	4/4	21.24	47.30	61.93	68.27	72.60	79.58	80.93
PTQ4ViT [17]	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT [44]	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
PD-Quant [25]	4/4	1.51	32.45	62.46	71.21	73.76	79.87	81.12
RepQ-ViT [35]	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
Adalog [43]	4/4	**72.13**	**79.16**	62.14	71.88	77.47	79.27	80.92
Pack-PTQ (Ours) w/o MP	4/4	61.03	75.71	64.53	71.57	78.45	80.82	82.34
Pack-PTQ (Ours) with MP	4/4	66.83	77.89	**68.24**	**75.33**	**79.41**	**82.18**	**83.75**

and ViT-B, its complex quantizer for post-Softmax activations limits the efficiency of quantized models. Moreover, Adalog's advantages do not generalize to the W3/A3 setting, where our method significantly outperforms it. The results in Table 2 also underscore the importance of cross-block dependencies in maintaining performance during quantization, as evidenced by BRECQ's effectiveness on CNNs but poor performance on ViTs.

5.3 Results on Point Cloud Classification

We also conduct experiments on the 3D point cloud classification task, as shown in Table 3. Overall, our method outperforms all other methods in all settings, achieving impressive performance comparable to that of full-precision models. Notably, our method exhibits negligible performance degradation compared to the full-precision model in the W2/A4 and W3/A3 quantization settings without mixed-precision. Specifically, we observe a slight decrease of 0.45% and 0.17% for W2/A4, and 0.46% and 0.13% for W3/A3 quantization in terms of mAcc and OA, respectively. Moreover, in the W3/A3 setting, our method even surpasses the full-precision performance when using mixed precision, and consistently outperforms the full-precision model in the W4/A4 quantization setting regardless of whether mixed precision is used. This suggests that point cloud networks may inherently possess higher precision redundancy, which mitigates the impact of bit-width on the model's generalization ability.

Table 3: Quantization results on ModelNet40. "mACC" is the mean of class-wise accuracy (%) and "OA" is the overall accuracy (%).

Method	**Bit. (W/A)**	**mAcc**	**OA**
Full-Precision	32/32	92.01	88.54
MinMax	2/4	7.51	12.85
BRECQ [14]	2/4	51.25	43.41
Pack-PTQ (Ours) w/o MP	2/4	**91.56**	**88.37**
MinMax	3/3	78.95	75.77
BRECQ [14]	3/3	67.48	61.18
Pack-PTQ (Ours) w/o MP	3/3	91.55	88.41
Pack-PTQ (Ours) with MP	3/3	**92.28**	**89.53**
MinMax	4/4	91.44	**88.98**
BRECQ [14]	4/4	87.25	82.70
Pack-PTQ (Ours) w/o MP	4/4	**92.37**	88.71
Pack-PTQ (Ours) with MP	4/4	92.21	88.56

BRECQ suffers significant performance degradation in the W3/A3 setting, with a notable drop of 11.47% in mAcc and 14.59% in OA compared to MinMax. This degradation can be attributed to the fact that block-wise overlooks the cross-block relationships, thus leading to inaccurate quantitation parameters. In stark contrast, our method demonstrates remarkable stability and consistency across all scenarios, showcasing its robustness and effectiveness in preserving model performance even at lower bit-widths.

5.4 Ablation Studies

5.4.1 Effectiveness of Key Components of Pack-PTQ

To validate the effectiveness of block packing and mixed-precision (MP) optimization, we conduct ablation studies on ResNet18 and ViT-S models using the W3/A3 quantization setting, as shown in Table 4. We begin by evaluating the block-wise optimization (without any variants) as the baseline, which yields inferior performance, with ViT-S achieving only 0.42% accuracy. We then examine the impact of applying random packing and mixed-precision settings separately, as shown in the second and fourth rows of the table. These components lead to improvements for both architectures, with ViT-S achieving significant increases of 35.10% and 39.93% accuracy, respectively. Notably, replacing the random packing strategy with our HAda mechanism further enhances performance, particularly on the ViT architecture, where accuracy improves to 47.68% (+12.16%), as evident from the comparison between rows 2 and 3. This demonstrates the effectiveness of our HAda mechanism. Furthermore, combining block packing and MP optimization yields even better performance. Among the various configurations, jointly using HAda and MP (ours) proves to be the most advantageous, achieving the highest accuracy of 66.73% and 55.56% on ResNet18 and ViT-S, respectively.

We visualize the learned packs generated by our Hessian-guided adaptive packing mechanism in Figure 3. It reveals an interesting phenomenon that the earlier blocks in the network tend to cluster into a single pack. This suggests that these blocks are tightly coupled and should be optimized together, highlighting the importance of our packing strategy in capturing the complex dependencies within the network.

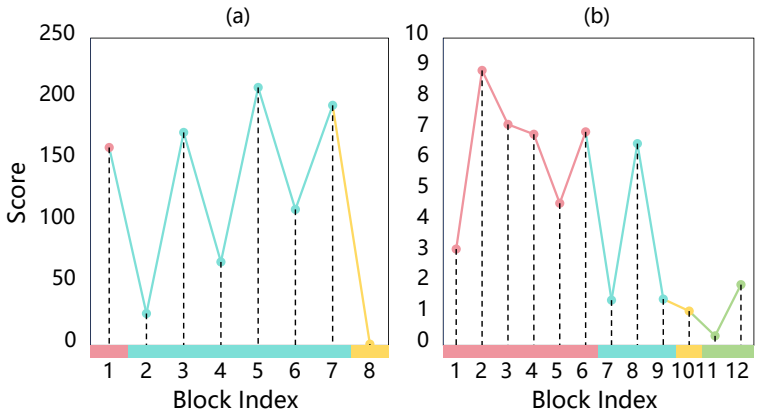

Figure 3: Visualization of important scores of blocks on (a) ResNet18 and (b) ViT-S. Different colors denote different packs.

Table 4: Top-1 accuracy (%) of different variants. "Random", "HAda", and "MP" denote the random packing, Hessian-guided adaptive packing, and mixed-precision respectively.

Method			Accuracy	
Random	HAda	MP	ResNet18	ViT-S
×	×	×	56.89	0.42
✓	×	×	62.81	35.52
×	✓	×	64.46	47.68
×	×	✓	56.82	40.35
✓	×	✓	64.53	47.86
×	✓	✓	**66.73**	**55.56**

5.4.2 Training Efficiency Analysis

As shown in Table 5, Pack-PTQ exhibits high efficiency across various models, highlighting its practicality in execution time. Notably, it only requires 0.79 hours on ResNet18 and 1.80 hours on MNV2. For MNasNetx2, ViT-S and DeiT-S, Pack-PTQ remains competitive, with execution times of approximately 2.3 hours, indicating its scalability. Notwithstanding, the results for more complex models (e.g., Swin-S) indicate that additional optimization efforts should be explored in future research to improve efficiency.

Table 5: Execution time (hours) of Pack-PTQ.

CNN Models		ViT Models	
Model	**Exe. Time**	**Model**	**Exe. Time**
ResNet18	0.79	ViT-S	2.31
MNV2	1.80	DeiT-S	2.27
MNasx2	2.25	Swin-S	6.82

6 Conculsion

This paper presents Pack-PTQ, a novel method for post-training quantization of neural networks. Pack-PTQ assigns a Hessian-based importance score to each block, which enables the network to be divided into non-overlapping packs that serve as the base unit for reconstruction. Furthermore, to promote accuracy, Pack-PTQ introduces a pack-based mixed-precision quantization approach that leverages aggressive low-bit quantization for packs with lower sensitivities and high-bit quantization for those with high sensitivities. Extensive experiments on multiple vision tasks, including image classification and point cloud classification with diverse network architectures (e.g., CNNs, ViTs, and PointNet),

demonstrate that complex neural networks can be compressed into low-bit versions without significantly compromising performance. In future work, we plan to extend our Pack-PTQ to more complicated vision tasks, and focus on exploring more efficient methods for pack reconstruction, to reduce the computational overhead associated with this process.

References

[1] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021.

[3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10012–10022, 2021.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[5] Fenghong Yang, Runqing Jiang, Yan Yan, Jing-Hao Xue, Biao Wang, and Hanzi Wang. Dual-mode learning for multi-dataset x-ray security image detection. *IEEE Transactions on Information Forensics and Security*, 2024.

[6] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7262–7272, 2021.

[7] 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 *Advances in Neural Information Processing Systems*, pages 12077–12090, 2021.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[9] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:10558–10578, 2024.

[10] Runqing Jiang, Yan Yan, Jing-Hao Xue, Biao Wang, and Hanzi Wang. When sparse neural network meets label noise learning: A multistage learning framework. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2208–2222, 2022.

[11] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[12] Runqing Jiang, Yan Yan, Jing-Hao Xue, Si Chen, Nannan Wang, and Hanzi Wang. Knowledge distillation meets label noise learning: Ambiguity-guided mutual label refinery. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2023.

[13] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[14] 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 *International Conference on Learning Representations*, 2021.

[15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[16] 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.

[17] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision*, pages 191–207, 2022.

[18] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206, 2020.

[19] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv preprint arXiv:2203.05740*, 2022.

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

[21] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330, 2022.

[22] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization, 2023.

[23] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. NoisyQuant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2023.

[24] Carlos Victorino Padeiro, Tse-Wei Chen, Takahiro Komamizu, and Ichiro Ide. Lightweight maize disease detection through post-training quantization with similarity preservation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2111–2120, 2024.

[25] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. PD-Quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 24427–24437, 2023.

[26] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1325–1334, 2019.

[27] Weihan Chen, Peisong Wang, and Jian Cheng. Towards mixed-precision quantization of neural networks via constrained optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5350–5359, 2021.

[28] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 293–302, 2019.

[29] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ-v2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems*, pages 18518–18529, 2020.

[30] Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. APTQ: Attention-aware post-training mixed-precision quantization for large language models. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 1–6, 2024.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. pages 1106–1114, 2012.

[32] Hongyi Yao, Pu Li, Jian Cao, Xiangcheng Liu, Chenying Xie, and Bingzhang Wang. RapQ: Rescuing accuracy for power-of-two low-bit post-training quantization. *arXiv preprint arXiv:2204.12322*, 2022.

[33] Yuexiao Ma, Huixia Li, Xiawu Zheng, Xuefeng Xiao, Rui Wang, Shilei Wen, Xin Pan, Fei Chao, and Rongrong Ji. Solving oscillation problem in post-training quantization through a theoretical perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7950–7959, 2023.

[34] Yongkweon Jeon, Chungman Lee, and Ho-young Kim. Genie: Show me the data for quantization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12064–12073, 2023.

[35] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-ViT: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 17227–17236, 2023.

[36] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[37] Yunshan Zhong, Jiawei Hu, Mingbao Lin, Mengzhao Chen, and Rongrong Ji. I&S-ViT: An inclusive & stable method for pushing the limit of post-training vits quantization. *arXiv preprint arXiv:2311.10126*, 2023.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[39] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[40] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[42] Xinrui Chen, Yizhi Wang, Renao Yan, Yiqing Liu, Tian Guan, and Yonghong He. Texq: Zero-shot network quantization with texture feature distribution calibration. In *Advances in Neural Information Processing Systems*, 2024.

[43] Zhuguanyu Wu, Jiaxin Chen, Hanwen Zhong, Di Huang, and Yunhong Wang. AdaLog: Post-training quantization for vision transformers with adaptive logarithm quantizer. In *European Conference on Computer Vision*, pages 411–427, 2025.

[44] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5380–5388, 2022.