Where and How to Enhance: Discovering Bit-Width Contribution for Mixed Precision Quantization

Haidong Kang[1] , **Lianbo Ma**[1*] , **Guo Yu**[2] and **Shangce Gao**[3]
[1]College of Software, Northeastern University
[2]Institute of Intelligent Manufacturing, Nanjing Tech University
[3]Faculty of Engineering, University of Toyama
hdkang@stumail.neu.edu.cn, malb@swc.neu.edu.cn, guo.yu@njtech.edu.cn, gaosc@eng.u-toyama.ac.jp

Abstract

Mixed precision quantization (MPQ) is an effective quantization approach to achieve accuracy-complexity trade-off of neural network, through assigning different bit-widths to network activations and weights in each layer. The typical way of existing MPQ methods is to optimize quantization policies (i.e., bit-width allocation) in a gradient descent manner, termed as **D**ifferentiable **MPQ** (DMPQ). At the end of the search, the bit-width associated to the quantization parameters which has the largest value will be selected to form the final mixed precision quantization policy, with the implicit assumption that the values of quantization parameters reflect the operation contribution to the accuracy improvement. While much has been discussed about the MPQ's improvement, the bit-width selection process has received little attention. We study this problem and argue that the magnitude of quantization parameters does not necessarily reflect the actual contribution of the bit-width to the task performance. Then, we propose a **S**hapley-based **MPQ** (SMPQ) method, which measures the bit-width operation's direct contribution on the MPQ task. To reduce computation cost, a Monte Carlo sampling-based approximation strategy is proposed for Shapley computation. Extensive experiments on mainstream benchmarks demonstrate that our SMPQ consistently achieves state-of-the-art performance than gradient-based competitors.

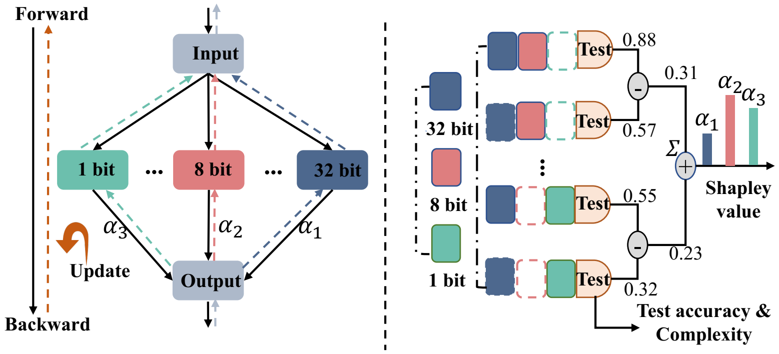
1 Introduction

With the explosive growth in advanced IoT applications, it is challenging to deploy deep neural networks (DNNs) on resource-constrained devices (e.g., MCUs and tiny NPUs) [Kwon *et al.*, 2024; Lu *et al.*, 2023; Aggarwal *et al.*, 2024], which suffer from extremely limited memory (e.g., KB-level SRAM, and MB-level storage) and low computing speed [Wang *et al.*, 2019; Zheng *et al.*, 2022]. This results in a big gap between the computational demands and limited resources. Therefore, it is desired to compress DNNs with no or minor performance degradation for efficient deployment. To achieve this goal, one typical way is to utilize the lower bit-width to quantize the entire network for lightweight and acceleration, a.k.a, fixed-precision quantization (FPQ) [Banner *et al.*, 2019; Bai *et al.*, 2024], where single-precision floating point weights or activations are mapped to lower bit-width ones. The recent developments in inference hardware have enabled variable bit-width arithmetic operations for DNNs, and this leads to the emergence of mixed precision quantization (MPQ) [Cai and Vasconcelos, 2020; Sun *et al.*, 2022; Ma *et al.*, 2023; Dong *et al.*, 2023; Liu *et al.*, 2024; Sun *et al.*, 2024b], which allows different bit-widths for different layers. Mixed precision of MPQ means a fine-grained bit-width allocation manner, where the quantization-insensitive layers can be quantized using much lower bit-widths than the quantization-sensitive layers. This way can naturally obtain more optimal accuracy-complexity trade-off than FPQ.

Limitations of existing DMPQs. As a typical way of existing MPQ methods, differentiable MPQ (DMPQ) aims to learn the optimal bit-width assignment in an end-to-end manner [Cai and Vasconcelos, 2020; Zhang *et al.*, 2021] (as shown in Fig. 1a). It applies continuous relaxation to transform the categorical choice of mixed precision quantization policy into continuous mixed precision quantization parameters [Yu *et al.*, 2020]. In this way, the DMPQ is relaxed to a differentiable quantization search problem, and the bit-width associated with the largest magnitude of learnable bit-width parameters (quantization parameters, α) updated by gradient descent in each layer is selected to form the final mixed precision quantization policy. Such gradient-based bit-width selection process in DMPQ relies on an important assumption that *the value of* α *updated by gradient descent represents the bit-width contribution to the accuracy improvement*. However, little work focuses on the validity of the above assumption of DMPQ.

In this paper, we attempt to understand and overcome this problem. At first, we find that the largest magnitude of learnable bit-width parameters α updated by gradient descent does not truly reflect the bit-width contribution in many cases, which degrades the performance of the derived mixed precision quantization policy. In this work, this phenomenon is defined as **the** α**'s pitfall issue**. Then, we observe that the bit-width operations in the quantization model are not independent of each other, which can be one potential reason of α's pitfall where the underlying relationships between bit-width

[*] Corresponding author



(a) Differentiable MPQ (b) Our SMPQ

Figure 1: The comparison between DMPQ and the proposed SMPQ. (a) DMPQ constructs a differentiable search space consisting of all bit-width candidates, and optimizes the learnable parameters α of bit-width updated by gradient descent, which does not reflect the contribution of bit-width candidates. (b) Our SMPQ directly evaluates the marginal contribution of bit-width candidates to the quantization task, according to the validation accuracy difference each possible bit-width candidate subset and its counterpart without the given bit-width candidate.

operations are ignored.

If the learnable bit-width parameters updated by gradient descent are not an excellent indicator (e.g., for optimization learnable bit-width parameters (α)) of bit-width contribution, how to select optimal quantization policy for each layer during bit-width selection process? To tackle this issue, we propose a Shapley-based mixed precision quantization method, termed as SMPQ, which leverages Shapley value [Ancona *et al.*, 2019; Castro *et al.*, 2009; Sun *et al.*, 2024a] to attribute real contributions to players in cooperative game theory. Fig. 1 shows the differences between our SMPQ and previous DMPQ methods. Shapley value directly measures the contributions of operations according to the validation accuracy difference. Meanwhile, it considers all possible combinations and quantifies the average marginal contribution to handle complex relationships between individual elements. Benefiting from these, Shapley value is effective for obtaining operation importance that is highly correlated with task performance. Instead of relying on the learnable bit-width parameters updated by gradient descent, the bit-width selection of SMPQ considers the actual contributions of bit-width operations to quantization performance, as shown in Fig. 1b, which can be described by "Where" and "How" to search optimal quantization policies.

Contributions. The main novelties of our work include:

- We discover that DMPQ suffers from the issue that the learnable bit-width parameters updated by gradient descent fail to reflect the actual bit-width contribution in many cases, and then we conduct deep analysis on the potential reason for such issue.
- We propose a Shapley-based MPQ (SMPQ) method via capturing actual the contribution of bit-widths on the MPQ task of validation dataset. It is the first attempt in exploring enhanced bit-width selection of DMPQ.
- Extensive experimental results show that SMPQ consistently gets more optimal quantization policies than gradient-based counterparts. Furthermore, we find that SMPQ performs more efficiently than its comparators on resource-constrained devices.

2 Related Work and Preliminaries

2.1 Related Work

Due to the page limit of the main text, the related work is provided in App. A.

2.2 Rethinking the DMPQ Method

Problem Setup. Suppose that a neural network F consists of n convolutional layers denoted as $L_1, ..., L_n$. Each layer L_l $(1 \leq l \leq n)$ has its corresponding set of weights W_l. The training process of the network F is conducted by solving an Empirical Risk Minimization problem. To quantize weights W_l and activations A_l, we define a search space S^α and S^β with n_w and n_a bit-width candidates, respectively. To be specific, S^α represents the search space of weights, and S^β denotes the search space of activations. To this end, the goal of MPQ is to find the optimal bit-width configuration for neural network F. Following the setting of method [Cai and Vasconcelos, 2020], for layer F_l, the quantization function is typically defined as follows:

$$\begin{aligned} Q_l(z) &= \textstyle\sum_{i=1}^{n_w} o_i^\alpha W_l \left(\sum_{j=1}^{n_a} o_j^\beta A_l\left(z\right)\right), \\ \text{s.t.} \quad & \textstyle\sum o_i^\alpha = 1, \sum o_j^\beta = 1, o^\alpha, o^\beta \in 0, 1, \end{aligned} \tag{1}$$

where filter f is parameterized by weight tensor W_i^T, and z represents the filter input. Our goal is to find the optimal bit-width configuration (o_α^* and o_β^*) for the entire neural network F. Then, we can obtain:

$$Y = \sum_{l=1}^{n} Q_l(F_l), \tag{2}$$

where Y represents output of the quantized network. However, we find that the search space is huge. For instance, in the case of ResNet-101, the number of possible bit-width configures reaches $6^{101} = 3.9 \times 10^{78}$ for each input when there are only 6 bit-width candidates. Therefore, a key challenge for the MPQ problem is how to reduce the search time.

The DMPQ Method. To reduce search cost, a representative DMPQ method, EdMIPS [Cai and Vasconcelos, 2020] is proposed and developed, and it can consume less than 10 GPU hours in ImageNet1K. The search space B of DMPQ is represented by a supernet, denoted as G = (V, E), where each node $v_i \in V$ represents a latent representation, and each edge (i, j) is associated with a bit-width $o^{(i,j)}$. The core idea of DMPQ is to transform the selection of discrete bit-width operation into a continuous optimization problem via continuous relaxation, and then optimize the supernet with gradient descent. The intermediate node is computed as a softmax mixture of candidate bit-widths:

$$\bar{o}(i,j)(v_i) = \sum_{o \in \mathcal{S}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{S}} \exp(\alpha_{o'}^{(i,j)})} o(v_i), \tag{3}$$

where $\alpha_o^{(i,j)}$ denotes the mixing weight of candidate bit-width $o^{(i,j)}$. S represents the search space consist of S^α and S^β. $\bar{o}$ is the input and mixed output of an edge. Given the optimal parameters α^*, the mixed-precision network should be derived by discretizing the soft selector variables $\alpha_o^{(i,j)}$ into the binary selectors. With such relaxation, the DMPQ search can be performed by jointly optimizing the model weight W and learnable bit-width parameters α for their corresponding bit-width candidate in a differentiable manner. In this work, we use a "winner-take-all" method to determine which bit-width is selected and others removed, computed as follows:

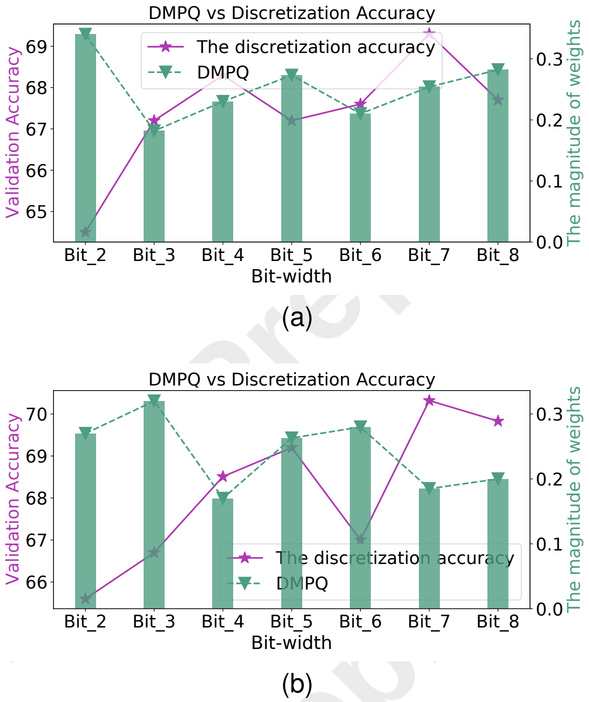
$$o_i^* = \begin{cases} 1, if\ i\ =\ argmax_j\ \alpha_o^{(i,j)}, \\ \quad 0,\ otherwise. \end{cases} \tag{4}$$

At the end of the search, the final mixed precision quantization policy is derived by selecting the maximum magnitude of learnable bit-width parameters α on every edge across all bit-width choices. Notably, our analysis does not limit the scope of EdMIPS, it also can be generalized to other differentiable MPQ (i.e., FracBits-SAT, GMPQ).

3 The Magnitude-based Selection Pitfall Issue

3.1 Rethinking Actual Contribution of Bit-width in DMPQ

Despite the successful application of DMPQ in various model compression scenarios, it is demonstrated in the following that the gradient-based bit-width selection process of DMPQ does not accurately represent the actual contribution of the bit-width, i.e., the α's pitfall. As depicted in Eq. 3, the bit-width selection process of the gradient-based in DMPQ relies on an important assumption that the largest magnitude of the learnable bit-width parameters updated by gradient descent signifies the contribution of the bit-width, i.e., the bit-width that corresponds to the highest probability is used as the optimal quantization strategy. However, due to the lack of comprehensive theoretical support for the rationality of the learnable bit-width parameters updated by gradient descent, we argue that selecting the bit-width based on the learnable bit-width parameters updated by gradient descent does not truly reflect its contribution, which leads to the α's pitfall issue (i.e., the magnitude-based selection pitfall). In the following section, we will substantiate our hypothesis with empirical evidence.



(a)

(b)

Figure 2: **DMPQ** *v.s.* **discretization accuracy** on (a) Random edge 1 and (b) Random edge 2. To be specific, we randomly select 2 edges from a pre-trained EdMIPS supernet on the search space S1.

3.2 Experimentation on α's Pitfall of DMPQ

To verify the aforementioned issue, we first introduce search space S1 (consisting of weight bit $\{2, 3, 4, 5, 6, 7, 8\}$ and activation bit $\{2\}$) to comprehensively explore relevant bit-width configurations and to facilitate the study of different aspects of quantization. Then, we conduct experimentation on learnable bit-width parameters updated by gradient descent does not represent actual bit-width contribution. This is achieved by calculating the discretization accuracy, which is obtained by using gradient descent to retrain quantized network[1] from the search space S1. Fig. 2 demonstrates that allocating small values of α updated by gradient descent to bit-widths can lead to high discretization accuracy at convergence.

These results indicate that the above α's pitfall issue results from the gradient descent process employed by DMPQ. To reveal the reason behind the α's pitfall issue, we argue that the α value from DMPQ is not always consistent with the corresponding contribution to the discretization accuracy in DMPQ. In fact, the contribution of bit-widths cannot solely be determined by the gradient's largest magnitude, and other factors such as the relationship of bit-width should be taken into account for accurate selection.

3.3 Further Analysis

Furthermore, we observe that the bit-widths in the quantization model are not independent but cooperative to each other during the learning of mixed precision quantization policy.

[1] Such network is quantized via selecting a bit-width candidate for a layer while fixing involved bit-widths for other layers

Especially, as illustrated in Eq. 1, the differentiable learning strategy of DMPQ solely focuses on minimizing the loss function of learnable bit-width parameters (α), but ignores the effect of cooperation between different bit-widths on the quantization. In a sense, this can be one potential reason why DMPQ suffers from the issue of α's failure, where the underlying relationships between bit-widths are ignored.

To validate our observation, motivated by method [Abdolrashidi *et al.*, 2021], we conduct further analysis to verify the issue, as shown in Fig. 3. To demonstrate the underlying relationships between bit-widths on different edges, we propose the search space S2, which consists of weight bit $\{1, 2, 3, 4\}$ and activation bit $\{2, 3, 4\}$. To be specific, B0 uses ResNet-18 (pre-trained in the ImageNet1K dataset) as the baseline model, and re-evaluates the discretization accuracy of the final mixed precision quantization policy based on EdMIPS [Cai and Vasconcelos, 2020] method. B1 changes the bit-width of the third edge from 4 bits to 3 bits of B0, and B2 changes the bit-width of the fourth edge from 2 bits to 3 bits of B0. Moreover, B3 alters the bit-widths of both edges of B0 by changing (4 bits, 2 bits) to (3 bits, 3 bits). Finally, we re-evaluate B0, B1, B2, and B3 in the ImageNet1K dataset using the training settings shown in Table 1. As a result, we find the impacts of combinations of the two edges differ from the simple accumulation of their separate influence. Specifically, Fig. 3 shows that the accuracy of B3 at the convergence point exceeds the sum of B1 and B2, which indicates that there is a joint contribution between the 3th edge and the 4th edge for the whole model. That is, the impact of combinations of the two edges differs from the simple accumulation of their separate influence. This observation reveals the complex relationships between different bit-widths on different edges: some bit-widths can collaborate with each other, resulting in a significant joint contribution to the model's performance. Hence, we aim to resolve this issue in this paper by proposing a novel approach that leverages the Shapley value to estimate bit-width contribution from cooperative game theory, which can accurately reflect the contribution of the bit-width.

Conclusion. DMPQs indeed suffer from the issue that the learnable bit-width parameters updated by gradient descent fail to reflect the actual bit-width contribution.

4 Mixed Precision Search via Discovering Bit-width Contribution

4.1 "Where to Search" in SMPQ

"Where to search", i.e., where the optimal mixed precision quantization policy is derived, means defining the search space of MPQ. In this respect, we formulate the MPQ search space B as a layer-wise supernet G (consisting of bit-width sets B), similar to other DMPQ methods (as presented in Section 2.2). The only difference between them is that the updating of our supernet relies on the Shapley evaluation rather than the gradient in terms of α, and thus requires no relaxation functions (e.g., $softmax$) that are typically used by other DMPQs (as shown in Eq. 3). One merit of the supernet is to provide weight sharing, which means that the weights of all quantization models sampled from the supernet directly inherit from those of the supernet without training from scratch. In this way, the search efficiency is greatly improved.

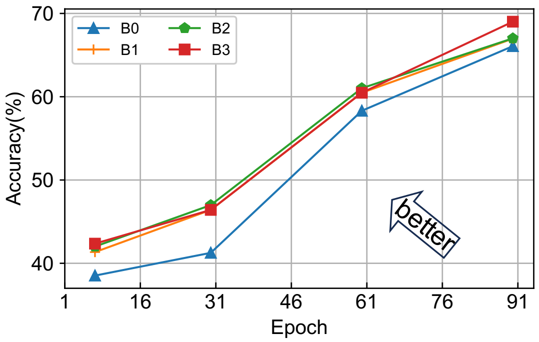


Figure 3: A study of bit-widths relationships. B0 denotes the final model quantized by DMPQ. B1 changes the bit-width of the third edge from 4 bits to 3 bits of B0, and B2 changes the bit-width of the fourth edge from 2 bits to 3 bits of B0. B3 alters the bit-widths of both edges of B0 by changing (4 bits, 2 bits) to (3 bits, 3 bits).

Let $\Omega(Q)$ be the complexity of quantized network through the precision policy Q. Then, the bit-width contribution weights α and network weights W of supernet G can be optimized in a layer-wise manner with the following objective:

$$\begin{aligned} &\min_{D_{val}} J_v(W^*, \alpha), \\ \text{s.t.} \quad &\mathbf{W}^* = \arg\min_{D_{train}} \lim_{\mathbf{W}} (\mathbf{W}, \alpha) , \Omega(Q) \leqslant \Omega_0. \end{aligned} \tag{5}$$

where J_v is validation loss, D_{train} is training datasets, D_{val} is validation datasets, and Ω_0 is the resource constraint of the deployed device.

4.2 "How to Search" in SMPQ

"How to search" refers to how to optimize bit-width parameters α in supernet. Rather than the magnitude of bit-width parameters updated by gradient descent, we focus on their practical contribution to the target quantization performance. Moreover, we observe that the quantization operations in the supernet are often correlative with each other: the combinations of various bit-width quantizations may have different joint impacts on the quantization performance compared to their separate ones. To tackle such complex relationships, we employ the Shapley value to measure the average marginal contribution of the bit-width to the validation performance, where the MPQ search process is mapped into a cooperative game. The main procedures are as follows:

Assume that there are associated N players in the game, and each subset of players $S \subseteq N$ (i.e., is coalition) is mapped to a real value $V(S)$ (as expected payoff of the players for cooperation) via a value function V. For the supernet, each layer has identical cell structures, and each cell has $|\mathcal{E}|$ edges each with $|\mathcal{O}|$ bit-widths. Then, the bit-width set, $N = \mathcal{O} \times \mathcal{E} = {o^{(i,j)}}_{o \in \mathcal{O}, (i,j) \in \mathcal{E}}$, can be regarded as players in the game, where all players cooperate towards the target performance $V(N)$. For bit-width $o^{(i,j)}$, its Shapley value $\psi_o^{(i,j)}$ is calculated by:

$$\psi_o^{(i,j)}(V) = \frac{1}{|N|} \sum_{\mathsf{c}=N|\circ(i,j)} \frac{V\left(S \cup \mathsf{o}^{(i,j)}\right) - V(S)}{\binom{|N|-1}{|S|}}. \tag{6}$$

In our scenario, we utilize the accuracy-complexity trade-off as the value function V to evaluate the quantization performance. However, the direct computation of Eq. 6 needs to enumerate all possible subsets, and thereby requires $2^{|\mathcal{O}|\times|\mathcal{E}|}$ time complexity, which is computationally prohibitive. For efficient evaluation, we adopt Monte-Carlo sampling method [Castro *et al.*, 2009] to get the approximation of the Shapley value, of which a truncated sample technique is also used to clip current sampling if the bit-width results in a significant performance drop. Accordingly, the search objective Eq. 5 needs to be changed as:

$$\begin{aligned} &\alpha_o^{(i,j)} \propto \psi_{D_{val}}(J_v(W^*, \alpha)) \, , \\ \text{s.t.} \quad &\mathbf{W}^* = \arg \min_{D_{train}} \lim_{\mathbf{W}} (\mathbf{W}, \alpha) \, , \Omega(Q) \leqslant \Omega_0, \end{aligned} \tag{7}$$

where ψ is the Shapley value of edge (i, j). Since it is hard to directly solve the above objective, we need to update $\alpha_o^{(i,j)}$ using the Shapley value evaluated by the Monte-Carlo sampling-based approximate method:

$$\alpha_k = {\alpha_o^{(i,j)}}_{k-1} + \xi \cdot \frac{q_k}{||q_k||_2}, \tag{8}$$

where α_t is the contribution weight at the k^{th} iteration during the search, q_k denotes the accumulated Shapley value in the k^{th} iteration, $|| \cdot ||_2$ is the L2 norm, and ξ is the term coefficient. In the bi-level optimization, $\mathbf{W^t}$ is trained by gradient descent $\nabla L_t(w_{t-1}, \alpha_{t-1})$ (as shown in Eq. 7) while α_t is optimized by Shapley value until convergence on training datasets. To alleviate redundant fluctuations induced by the sampling, we incorporate the momentum into the optimization for stabilization in the Monte-Carlo sampling process:

$$\boldsymbol{q}_k = \beta \cdot \boldsymbol{q}_{k-1} + \lambda \cdot \frac{\boldsymbol{\psi}(Acc_v(\boldsymbol{w}_{k-1}, \boldsymbol{\alpha}_{k-1}))}{||\boldsymbol{\psi}(Acc_v(\boldsymbol{w}_{k-1}, \boldsymbol{\alpha}_{k-1}))||_2}, \tag{9}$$

where $\beta + \lambda = 1$, the coefficients β and λ are used to balance the accumulated Shapley value and the current sampling, respectively. Acc_v is the validation accuracy as value function, and $\boldsymbol{w}_{t-1}$ is the supernet weights at $t-1$ iteration. At the end of the search, the final precision policy is derived by selecting the bit-width with the largest contribution on each edge.

4.3 Theoretical Analysis

In this section, we analyze the expected error of SMPQ based on the Shapley theory [Ancona *et al.*, 2019; Castro *et al.*, 2009] from a theoretical analysis perspective.

Theorem 1 *(Upper-bounding of the risk on SMPQ). Given the previous setting for SMPQ, for a set N of n players (bit-widths), according to Eq.7, let ψ be the Shapley value of edge (i, j), SMPQ is able to converge to the optimal point when:*

$$\Delta_\psi \quad = \quad \sum_{i=1}^{n} 50 \times |(\min_{D_{val}} \psi(Jv(W^*, \alpha^{(k)}))| < \epsilon, \tag{10}$$

where ϵ is the expectation error loss on specific datasets, and $abs(\cdot)$ is absolute function. We say that ψ has converged to Δ_ψ and stopped the iterative process.

Assume that after the kth iteration, the weight coefficients are ${\alpha_o^{(i,j)}}_k$, and after the $(k+1)$th iteration, the weight coefficients are ${\alpha_o^{(i,j)}}_{(k+1)}$. To this end, we only need to show that $\psi(Jv(W^*, \alpha^{(k)}))$ will also converge. Therefore, we propose Lemma 1.

Lemma 1 *(Bounding expectation (E) of its marginal contribution).* According to Eq. 6, the Shapley value $\psi_i(\cdot)$ of the game $\langle N, V \rangle$ for each subset of players (bit-widths) $S \subseteq N$ is the expectation (E) of its ***Marginal Contribution*** to players (bit-widths) that can be formulated as:

$$\psi_i(N, V) = E[p_o^{(i,j)}(V)]. \tag{11}$$

The Shapley value can be explained as follows. Assume all the players are arranged in some order, with all orderings being equally likely. Afterward, $\varphi_i(N, v)$ is the anticipated marginal contribution (overall orderings), of player i to the set of players who preceded him. According to the property of Shapley Value, for a mixed precision quantization game, one solution for N players in the quantization game must exist with the biggest contribution $max(N)$ on validation datasets, which obtains best quantization weights for every player. In a mixed precision quantization game, each player (bit-width) only needs to maximize its own Shapley value (i.e., $\psi(J_v(W^*, \alpha_o^{(i,j)})))$ so that $max(N)$ can be achieved as:

$$\max_{D_{val}} V(N) = \sum_{i \in N} \max_{D_{val}} \psi(\mathbf{W}^*, \alpha_o^{(i,j)}). \tag{12}$$

Proof of Lemma 1. According to Eq. 6, we can obtain:

$$\begin{aligned} \psi_o^{(i,j)} &= \mathbb{E}_{S \subseteq N \setminus \{o^{(i,j)}\}} \Big[V(S \cup \{o^{(i,j)}\}) - V(S) \Big] = \\ \sum_{S \subseteq N \setminus \{o^{(i,j)}\}} &\frac{|S|!(|N| - |S| - 1)!}{|N|!} \cdot \Big[V(S \cup \{o^{(i,j)}\}) - V(S) \Big] . \end{aligned} \tag{13}$$

Now, we define marginal contribution of $o^{(i,j)}$ to S as:

$$p_o^{(i,j)}(V, S) := V(S \cup \{o^{(i,j)}\}) - V(S), \tag{14}$$

Based on Eq. 14, we can get:

$$\psi_i(N, V) = \mathbb{E}_S \Big[V(S \cup \{o^{(i,j)}\}) - V(S) \Big] = E[p_o^{(i,j)}(V)]. \tag{15}$$

According to the **"Efficiency"** of Shapley, we can obtain:

$$\max_{D_{val}} V(N) = \sum_{i \in N} \max_{D_{val}} \psi(\mathbf{W}^*, \alpha_o^{(i,j)}), \tag{16}$$

The convergence of $\alpha_o^{(i,j)}$ relies on Eq. 8, 9, namely, $\alpha_k = \alpha_{k-1} + \xi \cdot \frac{q_k}{\|q_k\|_2}$, and $q_k = \beta q_{k-1} + \lambda \cdot \frac{\psi_{k-1}}{\|\psi_{k-1}\|_2}$ are exponential moving average, where $\psi_{k-1} = \psi(\text{Acc}_{val}(\mathbf{w}_{k-1}, \alpha_{k-1}))$. If $\mathbf{W}$ converge in D_{train}, α_k and q_k are stable in D_{val}. Therefore, we propose Lemma 2.

Lemma 2 *(The convergence of quantized neural network).* According to [Arora *et al.*, 2018], we assume that quantized neural network is optimized by gradient descent with the learning rate η, depth L of neural network, and deficiency margin

$c > 0$. For any $\varrho > 0$, the loss of quantized neural network at iteration T can be achieved as follows:

$$\frac{1}{\eta \cdot c^{2(L-1)/L}} \cdot \log\left(\frac{\ell(0)}{\varrho}\right) \leq T. \tag{17}$$

Proof of Lemma 2. Based on [Arora *et al.*, 2018], we have:

$$\frac{1}{\eta \cdot c^{2(L-1)/L}} \cdot \log\left(\frac{\ell(0)}{\varrho}\right) \leq T, \tag{18}$$

Then, supernet weights $\mathbf{W}^* = \arg\min_{D_{\text{train}}} \ell(\mathbf{W}, \alpha)$ converge via gradient descent with learning rate η, depth L, and deficiency margin $c > 0$, for any $\varrho > 0$. The loss $L(\mathbf{W}, \alpha)$ update by:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \nabla L(\mathbf{w}_{t-1}, \alpha_{t-1}), \tag{19}$$

starting from initial loss $\ell(0)$, the loss at iteration T satisfies:

$$\ell(T) \leq \ell(0) \cdot \exp\left(-\eta \cdot c^{2(L-1)/L} \cdot T\right), \tag{20}$$

Then, we can derive:

$$\exp\left(-\eta \cdot c^{2(L-1)/L} \cdot T\right) \leq \frac{\varrho}{\ell(0)}, \tag{21}$$

Finally, we can obtain:

$$T \geq \frac{1}{\eta \cdot c^{2(L-1)/L}} \cdot \log\left(\frac{\ell(0)}{\varrho}\right). \tag{22}$$

As $\mathbf{W} \rightarrow \mathbf{W}^*$, $J_v(\mathbf{W}^*, \alpha^{(k)})$ stabilizes, driving ψ to convergence.

Proof of Theorem 1. First, α is updated using Shapley values. By Lemma 2, for any $\varrho > 0$, we can derive $\ell(W_T, \alpha) \leq \varrho$, where $T \geq \frac{1}{\eta \cdot e^{-2L}} \cdot \log\left(\frac{\ell(0)}{\varrho}\right)$. Thus, $W_T \rightarrow W^*$. In bi-level optimization, W minimizes $\ell(W, \alpha)$, while α minimizes $J_v(W, \alpha)$. By Lemma 1, the Shapley value is:

$$\psi(N, V) = \mathbb{E}_S\left[V(S \cup \{o^{(i,j)}\}) - V(S)\right]. \tag{23}$$

In addition, the validation loss is bounded as follows:

$$\min_{D_{\text{val}}} V(N) \leq \sum_{i \in \mathcal{N}} \max_{D_{\text{val}}} \psi(W^*, \alpha_o^{(i,j)}). \tag{24}$$

As $\alpha_k \rightarrow \alpha^*$ (optimal quantization policy), we can derive $\psi(J_v(W^*, \alpha^{(k)}))$ is stable. Then, we can derive:

$$\min_{D_{\text{val}}} \psi(J_v(W^*, \alpha^{(k)})) \rightarrow 0. \tag{25}$$

Therefore, we have $\Delta_\psi \rightarrow 0$. For any $\epsilon > 0$, we can derive $\Delta_\psi < \epsilon$. Theorem 1 is proven.

Remarks. We further interpret how the above proofs can verify the effectiveness of our proposed SMPQ. First, Lemma 1 proves that SMPQ can obtain bounding expectation of its marginal contribution for any subset of players/bit-widths, e.g., there exists an optimal contribution for any coalition, which proves there is a joint contribution between bit-widths on different edges (as presented in section 3.3). Such joint contribution of different bit-widths is a key to ensure high performance of our SMPQ. Second, Lemma 2 ensures the achieving of $\mathbf{W}^*$. Finally, theorem 1 shows that SMPQ can quickly converge towards a global minimum.

Network	**ResNet-18**	**MobileNetV2**
Phase	MPQ Training	MPQ Training
Epoch	120	120
Batch Size	64	64
Optimizer	Adam	AdamW
Initial *Lr*	1e-3	1e-3
Lr Scheduler	Cosine	Cosine
Weight Decay	-	-
Warmup Epochs	-	30
Random Crop	✓	✓
Random Flip	✓	✓
Color Jittering	✓	-
$\boldsymbol{\xi}$	0.1	0.1
$\boldsymbol{\beta}, \lambda$	(0.8, 0.2)	(0.8, 0.2)
Search Space $S1$	(2, 3, 4, 5, 6, 7, 8), (4)	(2, 3, 4, 5, 6, 7, 8), (4)
Search Space $S2$	(1, 2, 3, 4), (2, 3, 4)	(1, 2, 3, 4), (2, 3, 4)
Search Space $S3$	[2, 8],[2, 8]	[2, 8],[2, 8]

Table 1: Detailed hyper-parameters and training scheme of SMPQ for ResNet-18 and MobileNetV2 on ImageNet1K dataset.

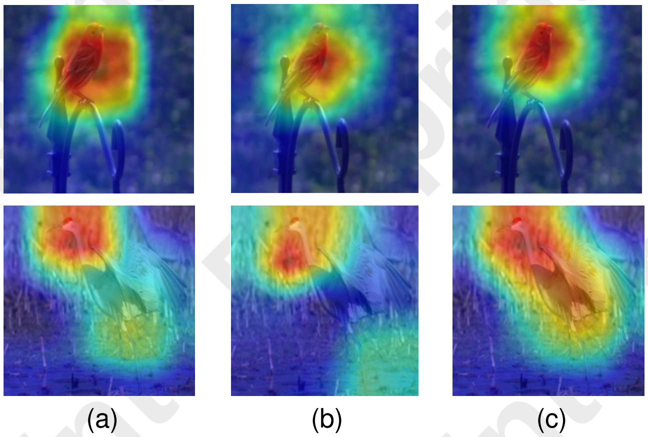


Figure 4: The saliency maps (computed by *Grad-cam* visualization on ImageNet1K. (a) the full precision ResNet-18, (b) MPQ policy searched by EdMIPS, and (c) MPQ policy searched by our SMPQ.

5 Experiments and Discussions

Experimental Settings. Following previous quantization methods [Cai and Vasconcelos, 2020; Chu *et al.*, 2021], we train and evaluate our framework on single hardware platform, i.e., one NVIDIA Tesla A100 GPU with an Intel(R) Xeon(R) Gold 6133 CPU. In this paper, we use the training settings shown in Table 1. Moreover, the training settings of ResNet-50, and Inception-V3 are the same as ResNet-18.

Hardware. All experiments are conducted on one NVIDIA Tesla A100 GPU with an Intel(R) Xeon(R) Gold 6133 CPU. Each experiment is executed on a single GPU at a time.

5.1 Comparison with State-of-the-Art Methods

The experimental results (all models with the same parameter setting are averaged over 10 random repetitions) on ImageNet1K are presented in Tables 2, and 3, respectively. From the reported results, we can have the following observations:

❶ As shown in Table 2, our SMPQ obtains 6.7, 2.6, 2.0, and 1.9 percent improvements over EdMIPS, DNAS, FracBits-SAT, and MetaMix with 2.9±0.07 GPU hours for ResNet-18

Methods	Top-1 (%) ↑	Search space.	Bit (W/A)	BOPs (G) ↓	Cost. ↓	Comp. ↑	Search method
Full-precision	73.09			1853.4			
PACT [Choi *et al.*, 2018]	69.8	{5}/{5}	5/5	35.2	-	52.7×	Fixed-precision
LSQ [Bhalgat *et al.*, 2020]	67.6	{2}/{2}	2/2	23.1	-	80.2×	Fixed-precision
PDQ [Chu *et al.*, 2019]	65.0	{1,2,4,8}/{2}	-/-	-	-	-	Fixed-precision
Hybrid-Net* [Chakraborty *et al.*, 2020]	62.7	{2}/{4}	∗/∗	-	-	-	Post-training Deterministic
HAWQ [Dong *et al.*, 2019]	68.5	{2}/{4}	∗/∗	34.0	15.6	54.5×	Sensitivity
EdMIPS [Cai and Vasconcelos, 2020]	65.9	[1,4]/[2,4]	∗/∗	34.7	9.5	54.5×	Differentiable
SMPQ (ours)	68.70±0.05)	[1,4]/[2,4]	∗/∗	33.4	2.4±0.08	55.5×	Shapley-based
HAWQv3 [Yao *et al.*, 2021]	70.4	{4}/{8}	∗/∗	72.0	-	25.7×	Sensitivity
DNAS [Wu *et al.*, 2018]	70.0	{1,2,4,8,32}/{1,2,4,8,32}	-/-	35.2	-	52.7×	Differentiable
SDQ [Huang *et al.*, 2022]	70.2	-	3.85/3	25.1	-	-	Differentiable
One-Shot [Koryakovskiy *et al.*, 2023]	<70	(2,4,8)	-	-	-	-	-
MetaMix [Kim *et al.*, 2023]	70.7	-	3.85/3	-	-	-	-
HAQ [Wang *et al.*, 2019]	70.4	[2,8]/{32}	∗/32	465	-	4.0×	RL-based
FracBits-SAT [Yang and Jin, 2021]	70.6	[2,8]/[2,8]	∗/∗	34.7	-	53.4×	Differentiable
GMPQ [Chu *et al.*, 2021]	69.9	[2,8]/[2,8]	∗/∗	15.3	0.6	121.0×	Differentiable
SMPQ (ours)	71.0±0.14	[2, 8]/{4}	∗/4	34.2	2.9±0.07	54.2×	Shapley-based
SMPQ (ours)	72.6±0.03	[2, 8]/{8}	∗/8	59.7	4.9±0.02	31.1×	Shapley-based

Table 2: Accuracy and efficiency results for ResNet. The symbol "Top-1" is the Top-1 accuracy of quantized model. The symbol "Search space." represents the search space. The symbol "Bit (W/A)" denotes the average bit-width for weights and activation parameters. The symbol "∗" means mixed-precision quantization. The symbol "Cost." denotes the MPQ policy search time that is measured by GPU hours. The symbol "Comp." means the compression ratio of BOPs. The symbol "BOPs" denotes the bit operations.

Methods	Top-1 (%) ↑	Search space.	Bit (W/A)	BOPs (G) ↓	Cost. ↓	Comp. ↑	Search method
Full-precision	72.5			337.9			
PACT [Choi *et al.*, 2018]	61.4	{5}/{5}	4/4	7.42	-	45.5×	Fixed-precision
LQ-net [Zhang *et al.*, 2018]	64.4	{4}/{4}	4/4	7.42	-	45.5×	Differentiable
RMSMP [Chang *et al.*, 2021]	69.0	-	∗/∗	5.35	-	63.2×	Sensitivity
HAQ [Wang *et al.*, 2019]	71.5	[2,8]/{32}	∗/32	42.8	51.1	7.9×	RL-based
HMQ [Habi *et al.*, 2020]	70.9	{2,3,4,5,6,7,8}/{min(b(X),8)}	∗/∗	5.32	33.5	63.5×	Differentiable
FracBits-SAT [Yang and Jin, 2021]	71.6	{2,3,4,5,6,7,8}/{2,3,4,5,6,7,8}	∗/∗	5.35	-	63.2×	Differentiable
GMPQ [Chu *et al.*, 2021]	70.4	[2,8]/[2,8]	3/∗	7.4	2.6	45.7×	Differentiable
SMPQ (ours)	71.7±0.01	[2,8]/{4}	∗/4	5.35	8.2±0.02	63.2×	Shapley-based
SMPQ (ours)	72.0±0.02	[2,8]/[2,8]	∗/∗	6.83	9.7±0.01	49.5×	Shapley-based

Table 3: Results for MobileNetV2. The red, and orange, cyan indicate the best, second-best, and third-best, respectively.

Method	#Param. (M) ↓	BOPs (G) ↓	Top1 ↑	Bit (W/A)
ResNet-50				
Full-prec.	97.8	3951.0	77.72%	32/32
HAWQ [Dong *et al.*, 2019]	13.1	61.3	75.3%	⋆/⋆
EdMIPS [Cai and Vasconcelos, 2020]	13.9	15.6	72.1%	⋆/⋆
HAQ [Wang *et al.*, 2019]	12.2	50.3	75.5%	⋆/⋆
HMQ [Habi *et al.*, 2020]	15.6	37.7	75.5%	⋆/⋆
BP-NAS [Yu *et al.*, 2020]	11.2	33.2	75.7%	⋆/⋆
EdMIPS-C [Cai and Vasconcelos, 2020]	13.7	16.0	65.6%	⋆/⋆
SMPQ (ours)	12.4	53.0	76.2±0.03%	⋆/⋆
Inception-V3				
Full-prec.	90.9	5850.0	78.9%	32/32
HWGQ [Cai *et al.*, 2017]	29.4	376.2	71.0%	⋆/⋆
Integer Only [Jacob *et al.*, 2018]	20.1	280.0	73.7%	⋆/⋆
EdMIPS [Cai and Vasconcelos, 2020]	29.4	376.2	72.4%	⋆/⋆
SMPQ (ours)	19.6	265.0	74.6±0.14%	⋆/⋆

Table 4: The performance comparison on larger networks.

on ImageNet1K, respectively. As shown in Table 3, SMPQ achieves 72.0±0.02 accuracy, higher than peer competitors with 9.7±0.01 GPU hours for MobileNetV2. These results highlight the effectiveness of SMPQ. This is because that SMPQ methods can utilize the relationships between different bit-widths on different edges by our proposed SMPQ method, which can directly measure bit-width contribution to the performance of the quantized model on the validation sets.

❷ When comparing the accuracy-complexity trade-off of SMPQ with baseline methods across various settings, it is evident that SMPQ provides a competitive accuracy-complexity trade-off under a variety of resource constraints. Importantly, this is achieved with significantly reduced BOPs (G). As shown in Table 2, compared with the chosen baselines for ResNet-18, the compression ratio of SMPQ is 1.7× more than DNAS, 1.8× more than EdMIPS, and 7.8× more than HAQ. This shows the effectiveness of our SMPQ.

❸ **To the best of our knowledge, this work is the first to compare the performance among different search spaces**. As shown in Tables 2, and 3, with bit-width increasing, the performance of models further increases, implying that search space is a key factor to MPQ. By comparing with "EdMIPS" under same search space, our SMPQ achieves competitive results with 68.7 top-1 accuracy and faster search speed, i.e., 2.4 GPU hours. v.s. 9.5 GPU hours. The main reason for better performance of SMPQ lies in that SMPQ can fully leverage relationship between different bit-widths through Shapley value. Shapley value is an excellent indicator of bit-width contribution, enabling accurately selects optimal mixed precision quantization policy for neural networks.

5.2 Effectiveness for Larger Networks

To demonstrate the generalizability of SMPQ, we conduct experiments that scale to larger networks (i.e., ResNet-50, Inception-V3, etc.). The results are listed in Table 4. As shown in Table 4, compared with the chosen baselines for ResNet-50, the Top-1 accuracy of SMPQ is 4.1 more than EdMIPS, 0.7 more than HMQ, and 10.6 more than EdMIPS-C. By comparing with the chosen baselines for Inception-V3,

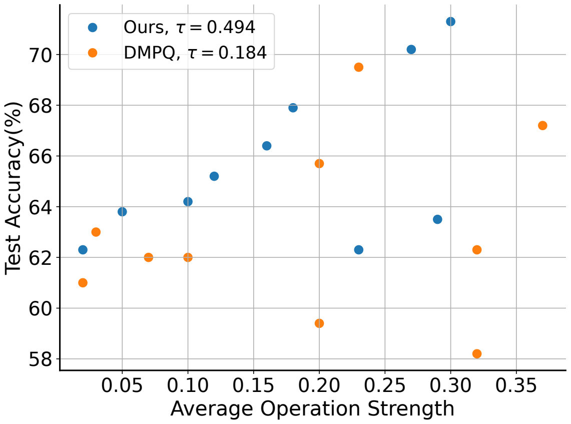


Figure 5: The correlation between test accuracy and average bit-width magnitude of 10 discrete quantization policies, which are sampled from the supernet using DMPQ and our SMPQ on ImageNet1K.

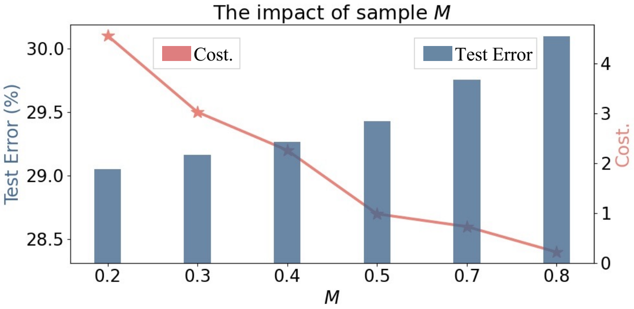


Figure 6: The impact of samples M.

the Top-1 accuracy of SMPQ is 3.6 more than HWGQ, 0.9 more than "Integer Only", and 2.2 more than EdMIPS. We can figure out that our approach is far better than peer competitors, which implies good generalizability on larger networks.

step size ξ	$\beta = 0.25$		$\beta = 0.5$		$\beta = 0.75$		$\beta = 1$	
	Err.(%)	**BOPs.(G)**	**Err.(%)**	**BOPs.(G)**	**Err.(%)**	**BOPs.(G)**	**Err.(%)**	**BOPs.(G)**
0.01	30.5	31.5	30.2	31.7	29.9	33.6	29.4	33.1
0.05	31.1	30.7	30.4	31.4	29.0	34.2	29.5	34.6
0.1	29.4	34.6	29.7	34.8	29.0	34.2	29.5	34.6
0.5	31.4	32.5	31.6	33.7	29.5	35.1	30.1	32.6

Table 5: Ablation studies of β and ξ for ResNet-18, where $\beta = 1 - \lambda$.

5.3 Correlation Analysis

Then, we perform a correlation analysis using Kendall's Tau (τ) coefficient. After the search phase, we select 10 discrete quantization policies from the supernet and calculate their corresponding bit-width strength by averaging the magnitude of the learnable bit-width parameters. We then plot the test accuracy *vs.* average bit-width magnitude obtained by DMPQ and our SMPQ. To measure the correlation between test accuracy and average bit-width magnitude, the results of Kendall's Tau (KTau) coefficient on ImageNet1K are depicted in Fig. 5. The larger the KTau value, the more predicted contribution matches the real contribution. From Fig. 5, we can see that our SMPQ achieves a higher correlation with the test accuracy (τ = 0.494), while DMPQ (the learnable bit-width parameters are updated by gradient descent) shows a worse correlation with the final test accuracy. In summary, the correlation analysis demonstrates that our proposed SMPQ can predict the real contribution to bit-widths more accurately than DMPQ.

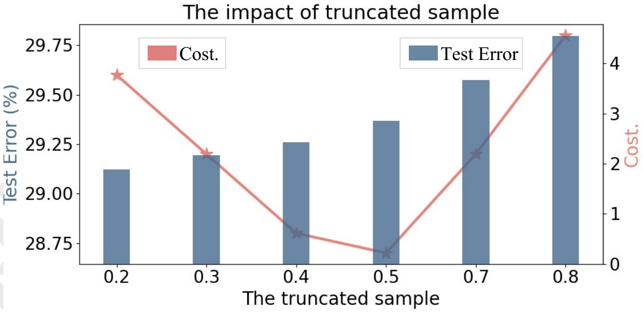


Figure 7: The impact of the truncated sample.

5.4 Ablation Study

Influence of sampling M and truncated sample. In fact, the two factors are crucial for model accuracy and search cost, as shown in Fig. 6 and 7. With the increase of M, the test error degrades significantly, while the search cost increases dramatically. As a result, when M = 10, we obtain accuracy-complexity trade-offs for SMPQ. For truncated sample, medium truncated sample (threshold=0.5) achieves the best accuracy-complexity trade-off.

Influence of β and ξ. Then, by varying β and ξ, we evaluate their influence with respect to the model accuracy. As shown in Table 5, we can find that our SMPQ is insensitive to β and ξ, and when $\beta = 0.75$ and $\xi = 0.05$, we get the optimal results.

6 Conclusion

In this paper, we attempt to understand and enhance MPQ methods from the bit-width selection perspective. We examine the magnitude-based quantization selection process of DMPQ and provide empirical evidence to show the α's pitfall issue, i.e., the value of α cannot well reflect the actual bit-width contribution, and then conduct deep analysis on the potential reason of such issue. To overcome this issue, we propose a Shapley-based MPQ method that directly estimates the bit-width contribution via its contribution to the performance of the quantized model on the validation sets. Specifically, the Shapley value of bit-widths can be efficiently approximated by a Monte-Carlo sampling algorithm with early truncation. The proposed method is able to consistently obtain more optimal quantization policies than magnitude-based MPQs on a set of datasets and several search spaces.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 62472079, Fundamental Research Funds for the Central Universities (N2417008, N2417003), the Key Technologies R&D Program of Liaoning Province (2023JH1/10400082), and Guangdong Basic and Applied Basic Research Foundation (2024A1515012016).

References

[Abdolrashidi *et al.*, 2021] AmirAli Abdolrashidi, Lisa Wang, Shivani Agrawal, Jonathan Malmaud, Oleg Rybakov, Chas Leichner, and Lukasz Lew. Pareto-optimal quantized resnet is mostly 4-bit. *ArXiv*, abs/2105.03536, 2021.

[Aggarwal *et al.*, 2024] Shivam Aggarwal, Kuluhan Binici, and Tulika Mitra. Chameleon: Dual memory replay for online continual learning on edge devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(6):1663–1676, 2024.

[Ancona *et al.*, 2019] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

[Arora *et al.*, 2018] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *ArXiv*, abs/1810.02281, 2018.

[Bai *et al.*, 2024] Jinyu Bai, Sifan Sun, Weisheng Zhao, and Wang Kang. Cimq: A hardware-efficient quantization framework for computing-in-memory-based neural network accelerators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(1):189–202, 2024.

[Banner *et al.*, 2019] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Conference on Neural Information Processing Systems*, 32, 2019.

[Bhalgat *et al.*, 2020] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 696–697, 2020.

[Cai and Vasconcelos, 2020] Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2020.

[Cai *et al.*, 2017] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5918–5926, 2017.

[Castro *et al.*, 2009] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *COMPUT OPER RES*, 36(5):1726–1730, 2009.

[Chakraborty *et al.*, 2020] Indranil Chakraborty, Deboleena Roy, Isha Garg, Aayush Ankit, and Kaushik Roy. Constructing energy-efficient mixed-precision neural networks through principal component analysis for edge intelligence. *Nature Machine Intelligence*, 2(1):43–55, 2020.

[Chang *et al.*, 2021] Sung-En Chang, Yanyu Li, Mengshu Sun, Weiwen Jiang, Sijia Liu, Yanzhi Wang, and Xue Lin. Rmsmp: A novel deep neural network quantization framework with row-wise mixed schemes and multiple precisions. In *International Conference on Computer Vision*, pages 5251–5260, 2021.

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

[Chu *et al.*, 2019] Tianshu Chu, Qin Luo, Jie Yang, and Xiaolin Huang. Mixed-precision quantized neural network with progressively decreasing bitwidth for image classification and object detection. *ArXiv*, abs/1912.12656, 2019.

[Chu *et al.*, 2021] Tianshu Chu, Qin Luo, Jie Yang, and Xiaolin Huang. Mixed-precision quantized neural networks with progressively decreasing bitwidth. *Pattern Recognition*, 111:107647, 2021.

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

[Dong *et al.*, 2023] Peijie Dong, Lujun Li, Zimian Wei, Xin Niu, Zhiliang Tian, and Hengyue Pan. Emq: Evolving training-free proxies for automated mixed precision quantization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17076–17086, 2023.

[Habi *et al.*, 2020] Hai Victor Habi, Roy H Jennings, and Arnon Netzer. Hmq: Hardware friendly mixed precision quantization block for cnns. In *European Conference on Computer Vision*, pages 448–463. Springer, 2020.

[Huang *et al.*, 2022] Xijie Huang, Zhiqiang Shen, Shichao Li, Zechun Liu, Xianghong Hu, Jeffry Wicaksana, Eric Xing, and Kwang-Ting Cheng. Sdq: Stochastic differentiable quantization with mixed precision. *ArXiv*, abs/2206.04459, 2022.

[Jacob *et al.*, 2018] 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 *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[Kim *et al.*, 2023] Han-Byul Kim, JooHyung Lee, Sungjoo Yoo, and Hong-Seok Kim. Metamix: Meta-state precision searcher for mixed-precision activation quantization. In *Association for the Advancement of Artificial Intelligence*, Vancouver, Canada, Nov. 2023.

[Koryakovskiy *et al.*, 2023] Ivan Koryakovskiy, Alexandra Yakovleva, and Valentin Buchnev. One-shot model for mixed-precision quantization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, jun. 2023.

[Kwon *et al.*, 2024] Eunji Kwon, Jongho Yoon, and Seokhyeong Kang. Mobile transformer accelerator exploiting various line sparsity and tile-based dynamic quantization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(6):1808–1821, 2024.

[Liu *et al.*, 2024] Lian Liu, Ying Wang, Xiandong Zhao, Weiwei Chen, Huawei Li, Xiaowei Li, and Yinhe Han. An automatic neural network architecture-and-quantization joint optimization framework for efficient model inference. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(5):1497–1510, 2024.

[Lu *et al.*, 2023] Zhichao Lu, Chuntao Ding, Felix Juefei-Xu, Vishnu Naresh Boddeti, Shangguang Wang, and Yun Yang. Tformer: A transmission-friendly vit model for iot devices. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):598–610, 2023.

[Ma *et al.*, 2023] Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Yongjian Wu, Guannan Jiang, Wei Zhang, and Rongrong Ji. Ompq: Orthogonal mixed precision quantization. In *Association for the Advancement of Artificial Intelligence*, volume 37, pages 9029–9037, 2023.

[Sun *et al.*, 2022] Zhenhong Sun, Ce Ge, Junyan Wang, Ming Lin, Hesen Chen, Hao Li, and Xiuyu Sun. Entropy-driven mixed-precision quantization for deep network design. *Conference on Neural Information Processing Systems*, 35:21508–21520, 2022.

[Sun *et al.*, 2024a] Qiheng Sun, Jiayao Zhang, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. Shapley Value Approximation Based on Complementary Contribution . *IEEE Transactions on Knowledge & Data Engineering*, 36(12):9263–9281, 2024.

[Sun *et al.*, 2024b] Sifan Sun, Jinyu Bai, Zhaoyu Shi, Weisheng Zhao, and Wang Kang. Cim^{2}pq: An array-wise and hardware-friendly mixed precision quantization method for analog computing-in-memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(7):2084–2097, 2024.

[Wang *et al.*, 2019] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019.

[Wu *et al.*, 2018] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018.

[Yang and Jin, 2021] Linjie Yang and Qing Jin. Fracbits: Mixed precision quantization via fractional bit-widths. In *Association for the Advancement of Artificial Intelligence*, volume 35, pages 10612–10620, 2021.

[Yao *et al.*, 2021] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pages 11875–11886. PMLR, 2021.

[Yu *et al.*, 2020] Haibao Yu, Qi Han, Jianbo Li, Jianping Shi, Guangliang Cheng, and Bin Fan. Search what you want: Barrier panelty nas for mixed precision quantization. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020.

[Zhang *et al.*, 2018] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *European Conference on Computer Vision*, pages 365–382, 2018.

[Zhang *et al.*, 2021] Zhaoyang Zhang, Wenqi Shao, Jinwei Gu, Xiaogang Wang, and Ping Luo. Differentiable dynamic quantization with mixed precision and adaptive resolution. In *International Conference on Machine Learning*, pages 12546–12556. PMLR, 2021.

[Zheng *et al.*, 2022] Yang-Lin Zheng, Wei-Yi Yang, Ya-Shu Chen, and Ding-Hung Han. An energy-efficient inference engine for a configurable reram-based neural network accelerator. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(3):740–753, 2022.