This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

EMQ: Evolving Training-free Proxies for Automated Mixed Precision Quantization

Peijie Dong[1†] Lujun Li[2†] Zimian Wei[1] Xin Niu[1*] Zhiliang Tian[1] Hengyue Pan[1]
[1] National University of Defense Technology, [2] HKUST
[1]{dongpeijie, weizimian16, niuxin, tianzhiliang, hengyuepan}@nudt.edu.cn, [2]lilujunai@gmail.com

Abstract

Mixed-Precision Quantization (MQ) can achieve a competitive accuracy-complexity trade-off for models. Conventional training-based search methods require time-consuming candidate training to search optimized per-layer bit-width configurations in MQ. Recently, some training-free approaches have presented various MQ proxies and significantly improve search efficiency. However, the correlation between these proxies and quantization accuracy is poorly understood. To address the gap, we first build the MQ-Bench-101, which involves different bit configurations and quantization results. Then, we observe that the existing training-free proxies perform weak correlations on the MQ-Bench-101. To efficiently seek superior proxies, we develop an automatic search of proxies framework for MQ via evolving algorithms. In particular, we devise an elaborate search space involving the existing proxies and perform an evolution search to discover the best correlated MQ proxy. We proposed a diversity-prompting selection strategy and compatibility screening protocol to avoid premature convergence and improve search efficiency. In this way, our Evolving proxies for Mixed-precision Quantization (EMQ) framework allows the auto-generation of proxies without heavy tuning and expert knowledge. Extensive experiments on ImageNet with various ResNet and MobileNet families demonstrate that our EMQ obtains superior performance than state-of-the-art mixed-precision methods at a significantly reduced cost. The code will be released.

1. Introduction

Deep Neural Networks (DNNs) have demonstrated outstanding performance on various vision tasks [20, 25]. However, their deployment on edge devices is challenging due to high memory consumption and computation cost [14]. Quantization techniques [19, 6, 8] have emerged as a promising solution to address this challenge by performing computation and storing tensors at lower bit-widths than floating point precision, and thus speed up inference and reduce the memory footprint.

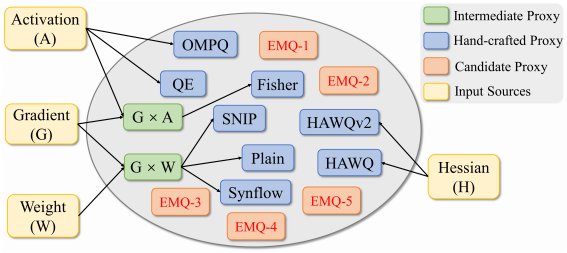

Figure 1. Illustration of the search space for EMQ. Our proposed search space encompasses the handcrafted proxies in mixed-precision quantization, whose input sources are activation(A), gradient (G), weight(W), Hessian(H), as well as their combinations (e.g., $G \times W$). The proposed search space highlights the extensive range of possible combinations, emphasizing the significant effort required to discover new MQ proxies.

Mixed-precision quantization (MQ) [40, 18, 10, 12, 8, 13] is a technique that assigns different bit-widths to the layers of a neural network to achieve a better accuracy-complexity trade-off and allows for the full exploitation of the redundancy and representative capacity of each layer. MQ methods can be categorized into training-based and training-free approaches. **Training-based methods** for MQ present it as a combinatorial search problem and adopt time-consuming Reinforcement Learning (RL) [40], Evolution Algorithm (EA) [41], one-shot [16], or gradient-based [42] methods to find the optimal bit-precision setting. However, these methods can be computationally intensive and require several GPU days on ImageNet [40, 3], limiting their applicability in scenarios with limited computing resources or high real-time requirements. Recently, **training-free approaches** [35, 28, 36, 8, 7, 21] have emerged for mixed-precision quantization, which starkly reduces the heavy computation burden. These approaches aim to reduce the computational burden by building alternative proxies to rank candidate bit-width configurations. For example, QE [35] uses the entropy value of features to automat-

*Corresponding author, † equal contribution.

ically select the bit-precision of each layer. These training-free methods have shown commendable effectiveness in assigning bit-precision to each layer in MQ. However, these training-free methods [8, 7, 44, 35, 28] have **two significant limitations**: (i) Lack of correlation analysis between training-free proxies and quantization accuracy. For instance, HAWQ-V2 [7] report quantitative results, which couple with quantified strategies and proxies. Thus, it is still unclear whether they can accurately predict the performance of different bit configurations. (ii) The discovery processes for proxies require expert knowledge and extensive trial tuning, which might not fully exploit the potential of training-free proxies. These limitations raise **two fundamental but critical questions**: (1) How can we accurately assess the predictive capability of existing proxies? and (2) How can we efficiently devise new proxies?

To address the first question, we develop a benchmark, namely, **MQ-Bench-101**, which comprises numerous bit configurations using the post training quantization strategy. Using this benchmark, we evaluated the performance of several existing training-free proxies, as reported in Tab. 1. Our results demonstrate that the current proxies exhibit limited predictive capabilities. Moreover, we attempt the proxies in training-free NAS and observe that the proxies require bit-weighting for effective quantification [8]. These observations[1] motivate us to devise improved proxies for MQ.

As for the second question, we present a general framework, **Evolving proxies for Mixed-precision Quantization (EMQ)**, whose aim is to use a reformative evolving algorithm to automate the discovery of MQ proxies. Specifically, we devise an elaborate and expressive search space encompassing all existing MQ proxies. As shown in Fig. 2, we formula MQ proxies as branched computation graphs composed of primitive operations and evolve them according to their predictive ability on MQ-Bench-101. We notice the importance of the ranking consistency of the top performing bit-widths rather than the overall rank consistency. To better account for the correlation of the top bit configurations, we introduce $Spearman@topk(\rho_{s@k})$ as the fitness function. To avoid premature convergence and improve search efficiency of the evolution process, we proposed the diversity-prompting selection strategy and compatibility screening protocol, respectively. We validate our framework on quantization-aware training and post-training quantization tasks. The experiments show that our searched MQ proxy is superior to the existing proxies in predictive capacity and quantization accuracy.

Main Contributions:

[1]There are two routines for proxies in MQ: scoring bit configurations as a whole and evaluating layer-wise sensitivity separately. In this paper, we focus on tackling the former and compare both methods in experiments that are discussed in detail in the App. D.1

Table 1. Ranking correlation (%) of training-free proxies on MQ-Bench-101. The $Spearman@topk(\rho_{s@k})$ are adopted to measure the correlation of the top performing bit configurations on MQ-Bench-101. We reported the mean and std of $\rho_{s@k}$ of 5 runs for all MQ proxies. All implementations are based on the official source code. The 'Time' column indicates the evaluation time (in seconds) for each bit-width configuration.

Method	$\rho_{s@20\%}$	$\rho_{s@50\%}$	$\rho_{s@100\%}$	Time(s)
BParams	$28.67_{\pm 0.24}$	$32.41_{\pm 0.07}$	$55.08_{\pm 0.13}$	2.59
HAWQ [8]	$23.64_{\pm 0.13}$	$36.21_{\pm 0.09}$	$60.47_{\pm 0.07}$	53.76
HAWQ-V2 [7]	$30.19_{\pm 0.14}$	$44.12_{\pm 0.15}$	$74.75_{\pm 0.05}$	42.17
OMPQ [28]	$7.88_{\pm 0.16}$	$16.38_{\pm 0.08}$	$31.07_{\pm 0.03}$	53.76
QE [35]	$20.33_{\pm 0.09}$	$24.37_{\pm 0.13}$	$36.50_{\pm 0.06}$	2.15
SNIP [21]	$33.63_{\pm 0.20}$	$17.23_{\pm 0.09}$	$38.48_{\pm 0.09}$	2.50
Synflow [37]	$39.92_{\pm 0.09}$	$44.10_{\pm 0.11}$	$31.57_{\pm 0.02}$	2.23
EMQ(Ours)	$\mathbf{42.59}_{\pm 0.09}$	$\mathbf{57.21}_{\pm 0.05}$	$\mathbf{79.21}_{\pm 0.05}$	**1.02**

- We introduce MQ-Bench-101, the first benchmark for training-free proxies in mixed-precision quantization (Sec. 4.2).
- We propose Evolving training-free proxies for Mixed-precision Quantization (EMQ) framework, which includes the diversity-prompting selection to prevent premature convergence and the compatibility screening protocol to improve the evolution search efficiency (Sec. 3).
- Experimental results demonstrate the superiority of the searched MQ proxy, indicating the effectiveness and flexibility of our proposed approach (Sec. 4).

2. Related Work

2.1. Mixed-precision Quantization

Quantization [30, 17, 32, 5] has been widely investigated as an effective technique to accelerate the inference phase of neural networks by converting 32-bit floating-point weight/activation parameters into low-precision fixed-point values. However, the contribution of each layer to the overall performance is to varying extents, and mixed-precision quantization [40, 27, 42, 46, 8, 7, 2] has been proposed to achieve a better trade-off between accuracy and complexity by assigning different bit-precision to different layers. Existing mixed-precision quantization methods can be classified into four categories: reinforcement learning-based approaches [27, 40, 9], evolutionary algorithm-based approaches [41], one-shot approaches [42, 18, 13] (including differentiable search approaches), and zero-shot approaches [8, 7, 35, 28] (also known as heuristic-based methods). Reinforcement learning-based approaches [40] use hardware feedback to search the bit-precision in discrete space. Evolutionary algorithm-based approaches [41] jointly search the pruning ratio, the bitwidth, and the architecture of the lightweight model from a hypernet. However, these search-based methods require an extremely large amount of computational resources and are time-consuming

due to the exponential search space. One-shot methods, such as DNAS [42] and Adabits [18], alleviate the searching problem greatly by constructing a supernet or hypernet where each layer consists of a linear combination or parallel blocks of outputs of different bit-precisions, respectively. Nevertheless, a differentiable search for mixed-precision quantization [42, 13] still needs a large amount of time due to the optimization of the large hypernet.

To address the bit-precision selection issue, heuristic criterion-based methods utilize zero-cost quantization proxies to rank the importance of layers. One approach is the Hessian-based quantization framework, which uses second-order information as the sensitivity metric. For instance, HAWQ [8] measures the sensitivity of each layer using the top Hessian eigenvalue and manually selects the bit-precision based on the relative sensitivity. HAWQ-V2 [7] proves that the average Hessian trace is a better sensitivity metric and proposes a Pareto frontier-based method for automatic bit-precision selection. Zero-cost proxies are also developed to handle mixed-precision quantization. QE Score [35] evaluates the entropy of the last output feature map without training, representing the expressiveness. OMPQ [28] proposes an Orthogonality Metric (ORM) that incorporates function orthogonality into neural networks and uses it to find an optimal bit configuration without any searching iterations.

These hand-crafted proxies used in previous works require expert knowledge and are often computationally inefficient [8, 7, 28]. These works suffer from major limitations. First, estimating the average Hessian trace using an implicit iterative approach based on the matrix-free Hutchinson algorithm [1] can lead to computational excesses and unstable iterative results. Second, the automatic bit-precision selection can only yield sub-optimal solutions, as the constraint space of the optimization problem is limited. For example, HAWQ-V2 [7] considers only one constraint on memory footprint when drawing the Pareto frontier of accuracy perturbation and model size, limiting the solutions to local optima in low-dimensional spaces. To overcome these challenges, we commence by automatically searching for the most effective training-free quantization proxy, capable of achieving competitive results with hand-crafted solutions.

2.2. Zero-cost Proxies for NAS

Recently, research has been focused on zero-shot/zero-cost neural architecture search (NAS), which estimates the performance of network architectures using zero-cost proxies based on small batches of data. Zero-shot NAS outperforms early NAS since it can estimate model performance without the need for complete training and training of super-networks in a single NAS, and without the need for forward and backward propagation of neural networks, which makes the entire process cost negligible. Zero-shot NAS is classified into architecture-level and parameter-level zero-shot NAS. Architecture-level zero-shot NAS evaluates the discriminative power of different architectures through inference. For example, NWOT [29] found that better-performing models can better distinguish the local Jacobian values of different images and proposed an indicator based on the correlation of input Jacobian for evaluating model performance. Parameter-level zero-cost NAS aims to evaluate and prune redundant parameters from neural networks. Several indicators have been proposed for this purpose, including GradNorm [31], Plain [31], SNIP [21], GraSP [39], etc. While both aim to alleviate the computational burden of traditional NAS, parameter-level zero-shot NAS has gained more attention due to its similarity with existing MQ proxies. Zero-cost proxies operate at the parameter level and are useful in measuring the sensitivity of each layer in a neural network. Parameter-level zero-cost proxies offer a more fine-grained approach to evaluating the performance of different network architectures, which can be used to optimize the overall performance of the system. Inspired by the existing MQ proxies, we adopt the zero-cost proxies in neural architecture search to measure the sensitivity of each layer by weighting the bit-width.

2.3. Revisiting Training-free Proxies

Mixed-precision quantization [40, 27, 42, 46, 8, 7, 2] aims to optimize the bit-width of each layer in a neural network to strike a balance between accuracy and efficiency. To achieve this, the mixed-precision quantization task can be formulated as a search for the best bit-width using training-free proxies. The search objective function is written as the following bi-level optimization form:

$$\begin{aligned} \min_{\mathcal{Q}} \ & \mathcal{L}_{val}(\boldsymbol{W}^*(\mathcal{Q}), \mathcal{Q}) \\ s.t. \ & \boldsymbol{W}^*(\mathcal{Q}) = \arg\min \ \mathcal{L}_{train}(\boldsymbol{W}, \mathcal{Q}) \\ & \Omega(\mathcal{Q}) \leqslant \Omega_0 \end{aligned} \tag{1}$$

where $\boldsymbol{W}$ refers to the quantized network weights, while $\mathcal{Q}$ denotes the quantization policy that assigns different bit-widths to weights and activations in various layers of the network. The computational complexity of the compressed network with the quantization policy $\mathcal{Q}$ is represented by $\Omega(\mathcal{Q})$. The task loss on the training and validation data is denoted by $\mathcal{L}train$ and $\mathcal{L}val$, respectively. The resource constraint of the deployment platform is represented by Ω_0. In order to obtain the optimal mixed-precision networks, the quantization policy $\mathcal{Q}$ and the network weights $\boldsymbol{W}(\mathcal{Q})$ are alternatively optimized until convergence or the maximal iteration number. However, training-free approaches [35, 28] take different routine. we formula the problem as:

$$\mathcal{Q}^* = \max_{\mathcal{Q}} \rho(\mathcal{Q}), \mathcal{Q} \in \mathcal{S} \tag{2}$$

where $\mathcal{Q}^*$ denotes the best MQ proxy in the search space S and ρ denotes the rank consistency of $\mathcal{Q}$. Given a neural

network of L layers, the MQ proxy can measure the sensitivity of i-th layer by $\mathcal{Q}^*(\theta_i)$. Then, the objective function is:

$$b^* = \max_{\mathbf{b}} \sum_{i=1}^{L} \left(b_i \times \mathcal{Q}^*(\theta_j)\right), s.t. \sum_{i=0}^{L} M^{(b_i)} \leq \Omega_0. \quad (3)$$

where $M^{(b_i)}$ denotes the model size of the i-th layer under b_i bit quantization and b^* represents the optimal bit-width configuration under the constraint of Ω_0.

To dive into the design of training-free proxies, we summarizes the existing MQ proxies in Tab. 2, which include the training-free proxies in neural architecture search [21, 36] and mixed precision quantization proxies [35, 28, 8, 7]. The proxies are categorized based on four types of network statistics as follows: (1) **Hessian as input**: HAWQ [8] employ the highest Hessian spectrum as the MQ proxy in Eqn. 2, where H is the Hessian matrix and $\lambda_i(H)$ is the i-th eigenvalue of H. HAWQ-V2 [7] adopt the average Hessian trace as proxy in Eqn. 2, where $tr(H_i)$ denotes the trace of H_i. (2) **Activation as input**: OMPQ [28] take the activation $\{z\}_i^N$ from the the i-th layer as input in Eqn. 2, where $||\cdot||_F$ denotes the Frobenius norm. QE [35] take the variance of the activation σ^2_{act} as input in Eqn. 2, where C_l represents the product of the kernel size K_l and input channel number C_{l-1} for layer l. Fisher [38] take the activation z as input in Eqn. 2. (3) **Gradient as input**: The formula of SNIP [21] is shown in Eqn. 2, where $\mathcal{L}$ is the loss function of a neural network with parameters θ, and $\odot$ is the Hadamard product. Synflow [37] take the weight θ and gradient $\frac{\partial \mathcal{R}}{\partial \theta}$ as input but do not require any data. (4) **Weight as input**: Plain [31], SNIP [21], and Synflow [36] employ the weights as input, as depicted in Eqn. 2, Eqn. 2, and Eqn. 2. For more related work, please refer to App. A.

Table 2. Revisiting mainstream handcrafted training-free proxies for mixed-precision quantization. The proxies are categorized based on four types of network statistics: Hessian matrix (denoted as "H"), activation (denoted as "A"), gradient (denoted as "G"), and weights (denoted as "W").

Type	MQ Proxy	Formula
H	HAWQ [8]	$spectrum(H) = \max_i\{\lambda_i(H)\}$
	HAWQ-V2 [7]	$trace(H) = \frac{1}{n}\sum_{i=1}^{n} tr(H_i)$
A	OMPQ [28]	$orm(z) = \frac{\|\|z_j^T z_i\|\|_z^2}{\|\|z_i^T z_i\|\|_z^2 \|\|z_j^T z_j\|\|_z^2}$
	QE [35]	$qe(\sigma_{\text{act}}) = \sum_{l=1}^{L} \log\left[\frac{C_l \sigma^2 \sigma^2_{\text{act}}}{\sigma^2_{\text{act}}}\right] + \log(\sigma^2_{\text{act}})$
G&A	Fisher [38]	$fisher(z) = \sum_{z_i \in z}\left(\frac{\partial \mathcal{L}}{\partial z} z\right)^2$
G&W	Plain [31]	$plain(\theta) = \frac{\partial \mathcal{L}}{\partial \theta} \odot \theta$
	SNIP [21]	$snip(\theta) = \left\|\frac{\partial \mathcal{L}}{\partial \theta} \odot \theta\right\|$
	Synflow [37]	$synflow(\theta) = \frac{\partial \mathcal{R}}{\partial \theta} \odot \theta, \mathcal{R} = \mathbb{1}^T\left(\prod_{\theta_i \in \theta} \|\theta_i\|\right)\mathbb{1}$

3. EMQ Framework

3.1. EMQ Search Space Design

To ensure the effectiveness and flexibility of our search space, we devise a comprehensive set of primitives that goes beyond the simple combinations of existing MQ proxies. Our search space comprises four input types, 56 primitive operations, and three types of computation graphs. We can construct existing MQ proxies by leveraging these elements, as depicted in Fig. 1. The abundance of operations in our search space enables us to explore a wide range of possible proxies and discover potential ones that previous handcrafted approaches may have overlooked.

Network Statistics as Input. As depicted in Fig. 1 and Tab. 2, the EMQ search space incorporates four distinct input types: activation, gradient, weight, and Hessian of convolutional layers, providing a comprehensive foundation of the sensitivity of each layer. Activation represent the feature map of a convolutional layer, while weight denote the weight of each convolutional layer. Gradient and Hessian matrix are the first and second derivatives of the loss function with respect to the convolution parameters, respectively. By combining these inputs with a diverse set of operations, the search algorithm can explore a vast search space and discover novel MQ proxies.

Primitive Operations. We employ a set of primitive operations encompassing both unary and binary operations to effectively process the neural network statistics. To ensure that the search space is sufficiently expressive and encompasses the existing MQ proxies, it is imperative to develop a varied range of operations. Inspired by AutoML-based methods [34, 11, 26], we provide a total of 24 unary operations and four binary operations to form the EMQ search space. Since the intermediate variables can be scalar or matrix, the total number of operations is 56. To efficiently aggregate information from different types of input, we propose aggregation functions to produce the final scalar output of the computation graph. The opulence of operations in the EMQ framework serves as the cornerstone to construct a diverse and expressive search space that can effectively capture the essence of MQ proxies and yield high-quality solutions. The Appendix F describes all the primitive operations in our search space.

Proxy as Computation Graph. We present each MQ proxy as a computation graph, which can be classified into three structures: sequential structure, branched structure, and Directed Acyclic Graph (DAG) based structure. The sequential structure is a fundamental computation graph, comprising a sequence of nodes with only one input. The branched structure is a hierarchical data structure composed of only one branch with two inputs, as shown in Fig. 2. It

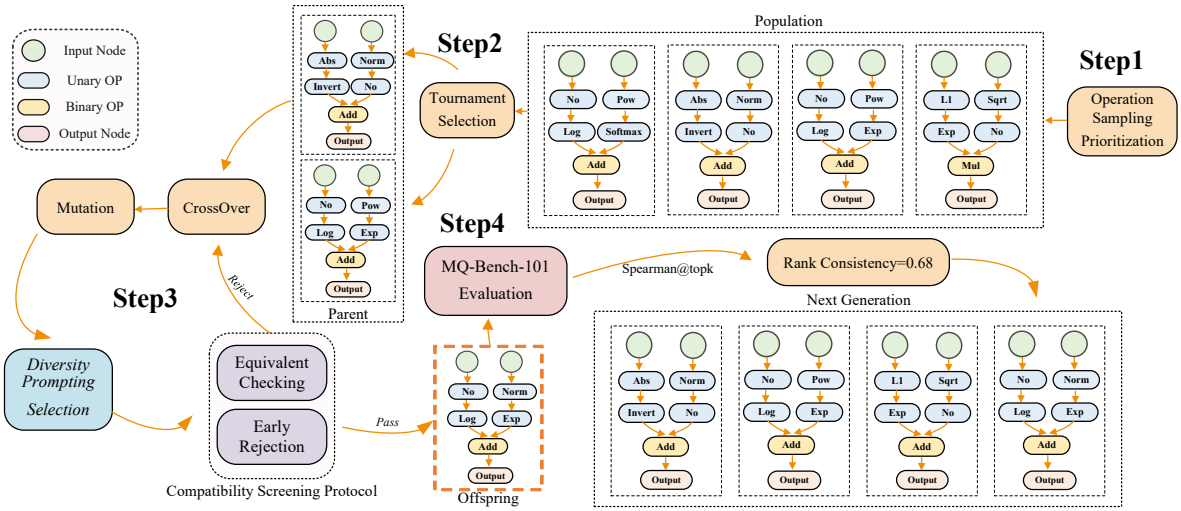

Figure 2. Overview of the Evolving training-free proxies for Mixed-precision Quantization (EMQ) framework. The framework involves four main steps: sampling a population of $|\mathcal{P}|$ candidate proxies from the EMQ search space using operation sampling prioritization (Step 1); generating parent proxies through tournament selection (Step 2); producing offspring via crossover, mutation, diversity-prompting selection and compatibility screening protocol (Step 3); and evaluating the offspring on the MQ-Bench-101 to measure the $Spearman@topk$ as the fitness function (Step 4).

offers a more potent representational capacity than the sequential structure. The DAG-based structure is most complex and expressive one, which allows for representing intricate dependencies between nodes. Each intermediate node is computed based on all of its predecessors, making it highly expressive yet complex. However, the intensive computation may suffer from sparsity resulting from dimension incompatibility issue or mathematical errors. Due to the trade-off between expressive ability and complexity, we predominantly utilize branched structure in the EMQ framework. For more details, please refer to the App. C.

Sparsity of the Search Space. We measure the sparsity of a search space using the validity rate metric, which represents the ratio between the number of valid proxies and the total number of sampled proxies. As shown in Tab. 3, the DAG-based structure achieves a validity rate of only 5.4%, indicating the sparsity of this search space. The sparsity can be attributed to the dimension incompatibility problem and the mathematical invalidity, which presents a challenge when searching for an effective proxy for EMQ. The dimension incompatibility issue arises from the fact that the input tensors for each proxy may have different dimensions, which not all operations can accommodate. The mathematical invalidity issue arises due to conflicting requirements of various operations, leading to violations of fundamental mathematical principles in the proxy representation. To enhance the validity rate of the search space and to improve the effectiveness of EMQ, it is crucial to address these challenges.

3.2. Evolutionary Framework

Inspired by AutoLoss-Zero [23] and AutoML-Zero[34], we introduce the Evolving proxies for Mixed-precision Quantization (EMQ) search algorithm. As depicted in Fig. 2 and Alg. 1, the EMQ pipeline involves several crucial steps. Firstly, we sample $|\mathcal{P}|$ candidate MQ proxies from the search space via operation sampling prioritization strategy. In each evolution, we select two parent proxies using tournament selections with a selection ratio of r. The parents then undergo crossover and mutation with probability p_c and p_m, respectively, to produce offspring. To prevent premature convergence, we propose a diversity-prompting selection (DPS) method to introduce diversity into the population and avoid population degradation. We also employ compatibility screening protocol to ensure the quality of the offspring before evaluating them on MQ-Bench-101. We adopt $Spearman@topk$ as the fitness function to better correlate with the top performing bit-widths. Finally, we only preserve the top-performing proxies within the population at each iteration. This process is repeated to identify the promising proxy for $\mathcal{N}$ generations.

Diversity-prompting Selection To introduce diversity into the population and prevent premature convergence, we implemented a diversity-prompting selection method. Instead of directly adding the offspring into the population, we employ additional random proxies and select the proxy with better performance in the population. There are mainly two benefits: (1) It can explore more candidate proxies with a very small population size and prevent premature convergence. (2) By selecting the best-performing individual among the newly generated individuals and the ran-

Algorithm 1 Evolution Search for EMQ

Input: Search space $\mathcal{S}$, population $\mathcal{P}$, sample ratio r, sampling pool $\mathcal{Q}$, top-k k, selection ratio r, max iteration $\mathcal{N}$.
Output: Best MQ proxy with highest $\rho_{s@k}$.

```
Initialize sampling pool Q := ∅;
P_0 := Initialize population(P_i) with SOP;
for i = 1 : N do
    Clear sampling pool Q := ∅;
    Randomly select r × P subnets P̂_i ∈ P to get Q;
    Candidates {A_i}_k := GetTopk(Q, k);
    Parent A_i^1, A_i^2 := RandomSelect({A_i}_k);
    Crossover A_i^c := CrossOver(A_i^1, A_i^2) with probability p_c;
    Mutate A_i^m := MUTATE(A_i^c) with probability p_m;
    Randomly sample A_i^n from S with OSP;
    // Diversity-prompting selection.
    if ρ_{s@k}(A_i^n) ≤ ρ_{s@k}(A_i^m) then
        Select A_i^m as offspring A_i^o;
    else
        Select A_i^n as offspring A_i^o;
    end if
    // Compatibility screening protocol.
    if CSP(A_i^o) is true then
        Append A_i^o to P;
    else
        Perform line 7;
    end if
    Remove the proxy with the lowest ρ_{s@k};
end for
```

dom individual, the evolution algorithm can converge more quickly and efficiently to an optimal solution.

Spearman@topk as Fitness All individuals are evaluated for rank consistency to determine the fitness function in the EMQ evolutionary algorithm. Intuitively, the correlation of the top-performing bit configurations outweigh the overall rank consistency, because we prioritize the ability to find the optimal bit configuration. To address this, we devise the $Spearman@topk$ coefficient, which is based on the vanilla Spearman coefficient but focuses only on the top-k performing bit-widths. We denote the number of candidate bit-widths as M, the ranking of ground-truth (GT) performance and estimated score (ES) of bit-widths $\{b_i\}_{i=1}^{M}$ are $\{p_i\}_{i=1}^{M}$ and $\{q_i\}_{i=1}^{M}$, respectively.

$$\rho_{s@k} = 1 - \frac{6\sum_{i\in D_k}(p_i - q_i)^2}{k(k^2-1)} \tag{4}$$

where $\rho_{s@k}$ is the Spearman coefficient computed on the top-k performing bit-widths based on the GT performance, and D_k is the set of indices of the top-k performing bit-widths based on GT performance $D_k = \{i | p_i < k \times N\}$.

Compatibility Screening Protocol To address the sparsity issues, we propose Compatibility Screening Protocol (CSP), which includes the equivalent checking and early rejection strategy. Equivalent checking identify distinct structures that are mathematically equivalent, thereby reducing redundant computation. For branched structure, equivalent checking involves the de-isomorphic process, which employ the Weisfeiler-Lehman Test [22] to filter out the equivalent structures. For more details, please refer to the App. D.2. **The early rejection strategy** aims to efficiently filter out invalid MQ proxies. By leveraging the characteristics of MQ proxies, the early rejection strategy employs meticulous techniques to identify and discard invalid proxies before performing a full evaluation on the MQ-Bench-101. This strategy significantly reduce the time cost of the evolution process or accelerate the convergence of the evolving algorithm. The early rejection strategy comprises three techniques: sensitivity perception, conflict awareness, and naive invalid check. **Sensitivity perception** refers to the ability of a proxy to percept whether it is insensitive to the varying of bit-widths, which denotes the incapable of measuring different bit-width and can be rejected at early stage. **Conflict awareness** allows for the identification of conflicting operations during the search process. For instance, the invert operation is in conflict with itself, as is the revert operation. For more detail please refer to App. D.3. **Naive Invalid Check** technique is employed to determine if the estimated score of a proxy is one of $\{-1, 1, 0, nan, inf\}$, indicating that it is indistinguishable. Consequently, such proxies can be rejected at an early stage. For more details, please refer to App. D.4.

Operation Sampling Prioritization When searching for MQ proxies, random operation sampling results in a large number of invalid candidates. To mitigate this issue, we propose Operation Sampling Prioritization (OSP), which assigns different probabilities to different operations. For unary operations, we assign a higher probability to the no_op operation to sparsify the search space. For binary operations, we assign a higher probability to the $element\_wise\_add$ operation to ensure that most cases can function well. The proposed OSP can effectively reduce the number of invalid candidates and improve the efficiency of the search process.

3.3. Effectiveness of EMQ

Searched Training-Free Proxy Here is the formula of the searched MQ proxy:

$$emq(\theta) = log(|\frac{\partial \mathcal{R}}{\partial \theta}|)\sqrt{\frac{\sum_{i=1}^{n}|\theta_i|}{numel(\theta)+\epsilon}} \tag{5}$$

where $numel(\theta) = \prod_{i=1}^{n} d_i$ and it denotes the total number of elements in the weight θ, and d_i is the size of the i-th dimension. The $\mathcal{R} = \mathbb{1}^T\left(\prod_{\theta_i\in\theta}|\theta_i|\right)\mathbb{1}$ denotes synaptic flow loss proposed in Synflow [36]. The input type of proposed proxy is similar to existing MQ proxies [21, 31, 36]. It comprises two components: the logarithm of the absolute value of the derivative of the scalar loss function $\mathcal{R}$, and the

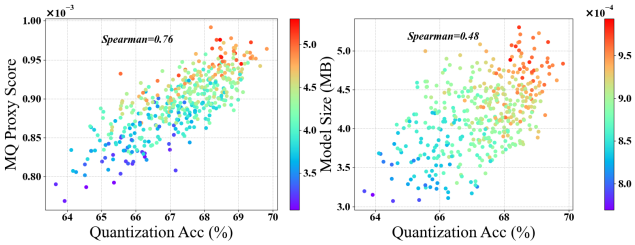

Figure 3. Left: Correlation between the searched EMQ proxy and the quantization accuracy. Right: Correlation between the model size and quantization accuracy.

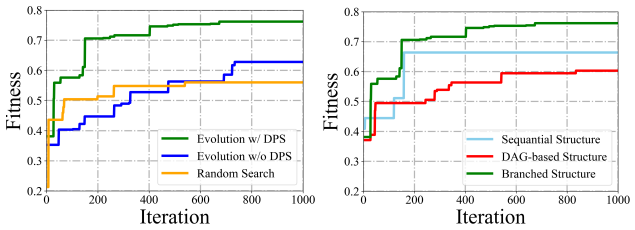

Figure 4. Left: Comparison of the evolutionary search and random search processes, with diversity-prompting selection strategy, denoted as "DPS". Right: Comparison between sequential, branched, and DAG-based structures during the evolution search.

square root of the normalized of the absolute values of the weight θ. Table 1 illustrates the effectiveness of our proposed EMQ, which outperforms the $\rho_{s@100\%}$ of SNIP [21] and Synflow [36] by a substantial margin of 40.73% and $47.64\% \uparrow$, respectively. Additionally, EMQ takes less time to evaluate one bit configuration (about $\times 2$ faster).

Correlation of the Searched EMQ Proxy To evaluate the predictive capability of our searched MQ proxy, we measure the ranking correlation between the searched MQ proxies and the accuracy for bit configurations on MQ-Bench-101. The correlation of the searched EMQ proxy with quantization accuracy is exhibited in Fig. 3. The figure on the left demonstrates an obvious positive correlation between our searched EMQ method and quantization accuracy, with a Spearman correlation coefficient of 76%. The color bar in the figure indicates the corresponding model size of the bit configuration. Conversely, the figure on the right indicates a weak correlation between model size and quantization accuracy, with a Spearman correlation coefficient of only 48%. The results suggest that the EMQ proxy has significantly better predictive capability than the baseline (model size as proxy) by a large margin of $28\% \uparrow$.

Superiority of Branched Structure We present a comparative analysis of the efficiency of three distinct structures: sequential, branched, and DAG-based structure. We assess the validity rate of each search space and investigate the impact of Operation Sampling Prioritization (OSP). Tab. 3 reveals that the sequential structure has the highest validity rate (41.7%) due to the simplicity of its computation graph. Nonetheless, this simplicity limits its expressiveness. The DAG-based structure is theoretically the most expressive search space, but it suffers from a lower validity rate (5.4%), which leads to slower convergence and higher computational costs. As shown in the right of Fig. 4, we observe that the DAG-based structure fails to achieve better performance, while the sequential structure is trapped in premature convergence due to the lower expressiveness of the search space. In contrast, the branched structure balances expressiveness and computational complexity. With two inputs, the branched structure search space can cover most of the existing MQ proxies and achieve a higher validity rate. For further details, please refer to the App. C.

Table 3. Validity rate of different search spaces. After applying the operation sampling prioritization strategy, the validity rate of the search spaces is prompted.

Computation Graph	w/o OSP (%)	w/ OSP (%)
Sequential structure	41.70	45.85
Branched structure	26.40	36.45
DAG-based structure	5.40	6.50

4. Experiments

4.1. Implementation Details

Datasets We perform experiments on the ImageNet dataset, which includes 1.2 million training samples and $50,000$ validation samples. A total of 64 training samples are randomly selected and the data augmentation techniques used are consistent with those employed in ResNet [15].

Evolution Settings In the evolutionary search process, we employ a population size of $|\mathcal{P}| = 20$, and the total number of iteration $\mathcal{N}$ is set to 1000. The selection ratio r for tournament selection is set to 0.25, and the probabilities of crossover and mutation, p_c and p_m, are set to 0.5. If the offspring pass the CSP, we randomly sample 50 bit configurations from MQ-Bench-101 and measure the ranking consistency of the offspring. To determine fitness, we calculated the average of $\rho_{s@20\%}$, $\rho_{s@50\%}$, and $\rho_{s@100\%}$ as the fitness function. During the evolution search, EMQ is extremely efficient, which only needs one NVIDIA RTX 3090 GPU and a single Intel(R) Xeon(R) Gold 5218 CPU. It only occupies the memory footprint of only one neural network during the evolution process.

Bit Assignment with Proxy After obtaining the searched EMQ proxy, we employ it to perform bit assignment by selecting the bit configuration with the highest MQ proxy score. Specifically, we first randomly sample a large number of candidate bit-widths that satisfy the model size constraints. We then traverse these candidate bit-widths and select the one with the highest score as the final bit assignment. The process of performing bit assignment is similar to [35], and it is extremely fast, taking only a few seconds to evaluate one bit configuration (shown in Tab. 1).

QAT Settings. For the QAT experiments, we employed two

NVIDIA Tesla V100 GPUs. The quantization framework excludes any integer division or floating point numbers in the network. We set the learning rate to $4e-4$ and the batch size to 512 for the training process. A cosine learning rate scheduler and SGD optimizer with $1e-4$ weight decay are implemented over 30 epochs. We follow the previous work [28] to keep the weight and activation of the first and last layers at 8 bits, constraining the search space to $\{4, 5, 6, 7, 8\}$.

PTQ Settings. For the PTQ experiments, we perform them on a single NVIDIA RTX 3090 GPU. We combine EMQ with the BRECQ [24] finetuning block reconstruction algorithm. In this experiment, we fix the activation precision of all layers to 8 bits, and limit the search to weight bit allocation in the search space of $\{2, 3, 4\}$.

4.2. MQ-Bench-101

We propose MQ-Bench-101, the first benchmark for evaluating the mixed-precision quantization performance of different bit configurations. To conduct our evaluation, we conduct post training quantization on ResNet-18 and assign each layer one of the bit-widths $b = \{2, 3, 4\}$, while keeping the activation to 8 bits. To manage the computational complexity of the search space, we randomly sample 425 configurations and attain their quantization performance under post-training quantization settings. MQ-Bench-101 enables us to identify high-performing quantization configurations and compare different MQ proxies fairly. For more details, please refer to the App. B.

4.3. Quantization-Aware Training

In this experiment, we conducted quantization-aware training on ResNet-18/50 and compared the results and compression ratios with previous unified quantization methods such as [32, 6, 47] and mixed-precision quantization methods like [40, 5, 44]. The results of our experiments are presented in Tab. 4 and Tab. 5.

Our results indicate that EMQ strikes the best balance between accuracy and compression ratio for ResNet-18 and ResNet-50. For instance, under the bit-width of activation as 6, the searched EMQ proxy achieve a quantization accuracy of 72.28% on ResNet-18 with 6.67Mb and 71BOPs, which achieves a 0.20% improvement over OMPQ [28]. Under the bit-width of activation as 8, EMQ can outperform HAWQ-V3 by 0.75%.

Moreover, compared to HAWQ-V3 [44], EMQ achieve 2.06% higher accuracy while having a slightly smaller BOPs (71 vs 72). EMQ achieve an accuracy of 76.70% on ResNet-50 with a model size of 18.7Mb and 148BOPs, and outperform HAWQ-V3 by 1.31% while having a smaller model size of 17.86Mb and 148BOPs compared to 18.7Mb and 154BOPs.

Table 4. Mixed-precision quantization results of ResNet-18. "Int" means only including integers during quantization. "Uni" represents uniform quantization. W/A is the bit-width of weight and activation. $*$ indicates mixed-precision. $\triangledown$ represents not quantizing the first and last layers. "MS" denotes the model size with bit-parameters and "BOPs" denotes the bit operations.

Method	W/A	Int	Uni	MS(M)	BOPs(G)	Top1(%)
Baseline	32/32	✗	-	44.6	1,858	73.09
RVQuant [33]	8/8	✗	✗	11.1	116	70.01
HAWQ-V3 [45]	8/8	✓	✓	11.1	116	71.56
OMPQ [28]	$*$/8	✓	✓	6.7	97	72.30
EMQ(Ours)	$*$/8	✓	✓	6.69	92	**72.31**
PACT$^{\triangledown}$ [6]	5/5	✗	✓	7.2	74	69.80
LQ-Nets$^{\triangledown}$ [47]	4/32	✗	✗	5.8	225	70.00
HAWQ-V3 [45]	$*$/$*$	✓	✓	6.7	72	70.22
OMPQ [28]	$*$/6	✓	✓	6.7	75	72.08
EMQ(Ours)	$*$/6	✓	✓	6.69	71	**72.28**

Table 5. Mixed-precision quantization results of ResNet-50.

Method	W/A	Int	Uni	MS(M)	BOPs(G)	Top1(%)
Baseline	32/32	✗	-	97.8	3,951	77.72
PACT$^{\triangledown}$ [6]	5/5	✗	✓	16.0	133	**76.70**
LQ-Nets$^{\triangledown}$ [47]	4/32	✗	✗	13.1	486	76.40
RVQuant [33]	5/5	✗	✗	16.0	101	75.60
HAQ [40]	*/32	✗	✗	9.62	520	75.48
Onebit-width [4]	*/8	✗	✓	12.3	494	**76.70**
HAWQ-V3 [45]	*/*	✓	✓	18.7	154	75.39
OMPQ [28]	*/5	✓	✓	18.7	156	76.28
EMQ(Ours)	*/5	✓	✓	17.86	148	**76.70**

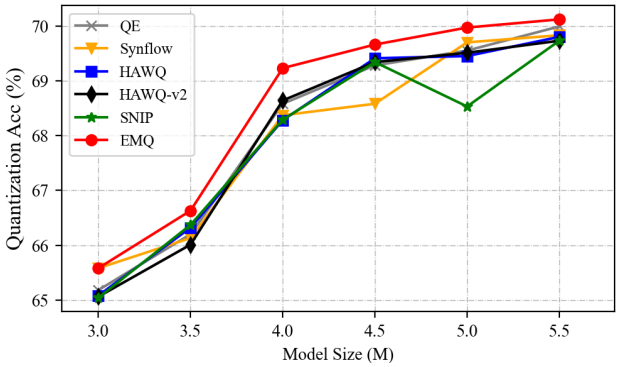

Figure 5. The accuracy and complexity trade-off between MQ proxies and our proposed EMQ approach for ResNet-18.

4.4. Post-Training Quantization

In this experiment, we conduct experiments on ResNet18 and MobileNetV2. Our proposed EMQ approach achieves a better trade-off among different model sizes, as illustrated in Tab. 6 and 7. To achieve this, we adopted the same block reconstruction quantization strategy as OMPQ [28]. Our experiments show that under the constraint of model size $\{4.0, 4.5, 5.5\}$, we achieve competitive results, surpassing OMPQ by 0.97%, 0.74%, and 0.51%, respectively. Moreover, we conducted a series of experiments to evaluate the

Table 6. Mixed-precision post-training quantization results on ResNet-18. † means using distillation in the finetuning process.

Method	W/A	Model size(M)	Top-1 (%)	#Data
Baseline	32/32	44.6	71.08	-
FracBits-PACT [6]	*/*	4.5	69.10	1.2M
OMPQ [28]	*/4	4.5	68.69	64
EMQ(Ours)	*/4	4.5	**69.66**	64
ZeroQ [2]	4/4	5.81	21.20	-
BRECQ† [24]	4/4	5.81	69.32	-
PACT [6]	4/4	5.81	69.20	-
HAWQ-V3 [45]	4/4	5.81	68.45	-
FracBits-PACT [6]	*/*	5.81	69.70	1.2M
OMPQ [28]	*/4	5.5	69.38	64
EMQ(Ours)	*/4	5.5	**70.12**	64
BRECQ [24]	*/8	4.0	68.82	1,024
OMPQ [28]	*/8	4.0	69.41	64
EMQ(Ours)	*/8	4.0	**69.92**	64

Table 7. Mixed-precision post-training quantization results on MobileNetV2.

Method	W/A	Model Size (Mb)	Top-1 (%)	#Data
Baseline	32/32	13.4	72.49	-
BRECQ [24]	*/8	1.3	68.99	1,024
OMPQ [28]	*/8	1.3	69.62	32
EMQ(Ours)	*/8	1.3	**70.72**	64
FracBits [43]	*/*	1.84	69.90	1.2M
BRECQ [24]	*/8	1.5	70.28	1,024
EMQ(Ours)	*/8	1.5	**70.75**	64

performance of different model sizes $\{3.0, 3.5, 4.0, 4.5, 5.0, 5.5\}$ using various quantization proxies, including QE [35], Synflow [36], HAWQ [8], HAWQ-V2 [7], and EMQ. To strike a trade-off between model complexity and quantization accuracy, we plot the quantization accuracy of each proxy against its respective model size, resulting in a pareto front (as shown in Fig. 5). The results demonstrate that our EMQ proxy provides a superior trade-off between model complexity and quantization performance when compared to the existing proxies.

4.5. Ablation Study

As presented in Tab. 3, we observe that the proposed operation sampling prioritization (OSP) technique improves the validity rate of the branched structure by $10.05\% \uparrow$. As illustrated in the left of Fig. 3.3, diversity-prompting selection (DPS) strategy can indeed prevent premature convergence (Blue line) and outperform the random search baseline (Yellow line) by a large margin. These findings suggest that the OSP and DPS strategy are indispensable components of EMQ. Tab. 8 demonstrates the effectiveness of the ablation study on improving efficiency through equivalent checking and early rejection when searching for branched structures. By implementing these strategies, we are able to proactively filter out approximately 97% of failed proxies, resulting in a significant reduction in computational cost.

Table 8. Efficiency improvement with equivalent checking and early rejection strategy on branched structure.

Equivalent Checking	Early Rejection	#Evaluated Proxies
✗	✗	$\sim 1 \times 10^4$
✓	✗	$\sim 9 \times 10^3$
✓	✓	$\sim 3 \times 10^2$

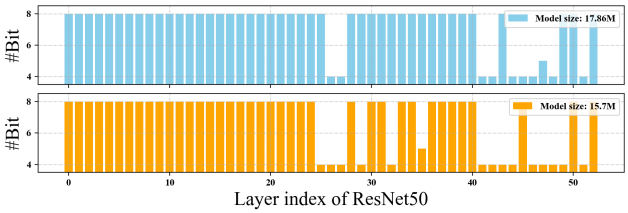

Figure 6. Assignment of bit configurations for weights under 18M and 16M model size constraints for ResNet-50. The bit-widths are searched for configurations of $\{4, 5, 6, 7, 8\}$

4.6. Visualization and Analysis

To intuitively demonstrate the bit-width assignment generated by the searched EMQ proxy, we visualize the quantization strategy of weights in different layers of ResNet50 with model size constraints of 16M and 18M in Fig. 6. We observe that for the bit-width assignment under different model constraints, the 29^{th}, 32^{nd}, 35^{th}, and 49^{th} layers are assigned lower bit-width, indicating that these layers are not as sensitive as others. Additionally, we can see from the bit-width assignment that the first and last layers have higher bit-width to achieve quantization accuracy.

5. Conclusion

In this paper, we present the Evolving proxies for Mixed precision Quantization (EMQ), a novel approach for exploring proxies for mixed-precision quantization (MQ) without requiring heavy tuning or expert knowledge. To fairly evaluate the MQ proxies, we build the MQ-Bench-101 benchmark. We leverage evolution algorithm to efficiently search for superior proxies that strongly correlate with quantization accuracy, using our diversity-prompting selection and compatibility screening protocol. The extensive experiments on the ImageNet dataset on ResNet and MobileNet families demonstrate that our EMQ framework outperforms existing state-of-the-art mixed-precision methods in terms of both accuracy and efficiency. We believe that our work inspires further research in developing efficient and accurate MQ techniques and enables deploying more efficient deep learning models in resource-constrained environments.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62025208) and the Open Project of Xiangjiang Laboratory (No.22XJ01012).

References

[1] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011. 3

[2] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 2, 3, 9

[3] Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2346–2355, 2020. 1

[4] Ting-Wu Chin, Pierce I-Jen Chuang, Vikas Chandra, and Diana Marculescu. One weight bitwidth to rule them all. *ArXiv*, abs/2008.09916, 2020. 8

[5] Ting-Wu Chin, I Pierce, Jen Chuang, Vikas Chandra, and Diana Marculescu. One weight bitwidth to rule them all. In *European Conference on Computer Vision*, pages 85–103. Springer, 2020. 2, 8

[6] 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. 1, 8, 9

[7] Zhen Dong, Zhewei Yao, Yaohui Cai, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. *arXiv preprint arXiv:1911.03852*, 2019. 1, 2, 3, 4, 9

[8] 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/CVF International Conference on Computer Vision*, pages 293–302, 2019. 1, 2, 3, 4, 9

[9] Ahmed T. Elthakeb, Prannoy Pilligundla, FatemehSadat Mireshghallah, Amir Yazdanbakhsh, and Hadi Esmaeilzadeh. Releq : A reinforcement learning approach for automatic deep quantization of neural networks. *IEEE Micro*, 40:37–45, 2020. 2

[10] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. *ArXiv*, abs/1902.08153, 2019. 1

[11] Hongyang Gu, Jianmin Li, Guang zhi Fu, Chifong Wong, Xinghao Chen, and Jun Zhu. Autoloss-gms: Searching generalized margin-based softmax loss function for person re-identification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4734–4743, 2022. 4

[12] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, 2019. 1

[13] Hai Victor Habi, Roy H. Jennings, and Arnon Netzer. Hmq: Hardware friendly mixed precision quantization block for cnns. *ArXiv*, abs/2007.09952, 2020. 1, 2, 3

[14] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*, 2016. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[16] Yiming Hu, Xingang Wang, Lujun Li, and Qingyi Gu. Improving one-shot nas with shrinking-and-expanding supernet. *Pattern Recognition*, 2021. 1

[17] 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. 2

[18] Qing Jin, Linjie Yang, and Zhenyu A. Liao. Adabits: Neural network quantization with adaptive bit-widths. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2143–2153, 2019. 1, 2, 3

[19] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. 1

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 1

[21] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. *ArXiv*, abs/1810.02340, 2018. 1, 2, 3, 4, 6, 7

[22] AA Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968. 6

[23] Hao Li, Tianwen Fu, Jifeng Dai, Hongsheng Li, Gao Huang, and Xizhou Zhu. Autoloss-zero: Searching loss functions from scratch for generic tasks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 999–1008, 2021. 5

[24] 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. *ArXiv*, abs/2102.05426, 2021. 8, 9

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014. 1

[26] Peidong Liu, Gengwei Zhang, Bochao Wang, Hang Xu, Xiaodan Liang, Yong Jiang, and Zhenguo Li. Loss function discovery for object detection via convergence-simulation driven search. *ArXiv*, abs/2102.04700, 2021. 4

[27] Qian Lou, Feng Guo, Lantao Liu, Minje Kim, and Lei Jiang. Autoq: Automated kernel-wise neural network quantization. *arXiv preprint arXiv:1902.05690*, 2019. 2, 3

[28] Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Guannan Jiang, Wei Zhang, and Rongrong Ji. Ompq: Orthogonal mixed precision quantization. *ArXiv*, abs/2109.07865, 2021. 1, 2, 3, 4, 8, 9

[29] Joseph Mellor, Jack Turner, Amos J. Storkey, and Elliot J. Crowley. Neural architecture search without training. *arXiv preprint arXiv:2006.04647*, 2020. 3

[30] Nelson Morgan et al. Experimental determination of precision requirements for back-propagation training of artificial neural networks. In *Proc. Second Int'l. Conf. Microelectronics for Neural Networks*, pages 9–16. Citeseer, 1991. 2

[31] Michael C. Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *NIPS*, 1988. 3, 4, 6

[32] Eunhyeok Park, Sungjoo Yoo, and Peter Vajda. Value-aware quantization for training and inference of neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 580–595, 2018. 2, 8

[33] Eunhyeok Park, Sungjoo Yoo, and Péter Vajda. Value-aware quantization for training and inference of neural networks. *ArXiv*, abs/1804.07802, 2018. 8

[34] Esteban Real, Chen Liang, David R. So, and Quoc V. Le. Automl-zero: Evolving machine learning algorithms from scratch. In *International Conference on Machine Learning*, 2020. 4, 5

[35] Zhenhong Sun, Ce Ge, Junyan Wang, Ming Lin, Hesen Chen, Hao Li, and Xiuyu Sun. Entropy-driven mixed-precision quantization for deep network design on iot devices. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 4, 7, 9

[36] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS*, 2020. 1, 4, 6, 7, 9

[37] Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *ArXiv*, abs/2006.05467, 2020. 2, 4

[38] Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. Faster gaze prediction with dense networks and fisher pruning. *ArXiv*, abs/1801.05787, 2018. 4

[39] Chaoqi Wang, ChaoQi Wang, Guodong Zhang, and Roger Baker Grosse. Picking winning tickets before training by preserving gradient flow. *ArXiv*, abs/2002.07376, 2020. 3

[40] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019. 1, 2, 3, 8

[41] Tianzhe Wang, Kuan Wang, Han Cai, Ji Lin, Zhijian Liu, and Song Han. Apq: Joint search for nerwork architecture, pruning and quantization policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[42] 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. 1, 2, 3

[43] Linjie Yang and Qing Jin. Fracbits: Mixed precision quantization via fractional bit-widths. In *AAAI Conference on Artificial Intelligence*, 2020. 9

[44] 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. 2, 8

[45] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. Hawqv3: Dyadic neural network quantization. In *International Conference on Machine Learning*, 2020. 8, 9

[46] 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. 2, 3

[47] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 365–382, 2018. 8