This CVPR 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.

One-Shot Model for Mixed-Precision Quantization

Ivan Koryakovskiy
koryakovskiy.ivan1@huawei.com

Alexandra Yakovleva
yakovleva.alexandra@huawei.com

Valentin Buchnev
buchnev.valentin@huawei.com

Temur Isaev
isaev.temur@huawei.com

Gleb Odinokikh
odinokikh.gleb@huawei.com

Huawei Technologies Co. Ltd.

Abstract

Neural network quantization is a popular approach for model compression. Modern hardware supports quantization in mixed-precision mode, which allows for greater compression rates but adds the challenging task of searching for the optimal bit width. The majority of existing searchers find a single mixed-precision architecture. To select an architecture that is suitable in terms of performance and resource consumption, one has to restart searching multiple times. We focus on a specific class of methods that find tensor bit width using gradient-based optimization. First, we theoretically derive several methods that were empirically proposed earlier. Second, we present a novel One-Shot method that finds a diverse set of Pareto-front architectures in O(1) time. For large models, the proposed method is 5 times more efficient than existing methods. We verify the method on two classification and super-resolution models and show above 0.93 correlation score between the predicted and actual model performance. The Pareto-front architecture selection is straightforward and takes only 20 to 40 supernet evaluations, which is the new state-of-the-art result to the best of our knowledge.

1. Introduction

In recent years, neural network quantization [31] has become a popular hardware-friendly compression technique. It is common to quantize linear and convolutional layer operands while leaving vector operands unchanged. Modern algorithms achieve lossless quantization into fixed 8-bit integer values in many applications [45, 15, 40, 35, 49, 25, 5]. At higher compression rates, mixed-precision is often needed [22, 44]. For example, models often require 8-bit precision for the first and last layers, while the middle layers can tolerate lower precision [15, 40]. In addition, the selected precision may depend on a quantized operation [7] or a hardware at hand [41]. This motivates many vendors to support mixed-precision models in hardware.

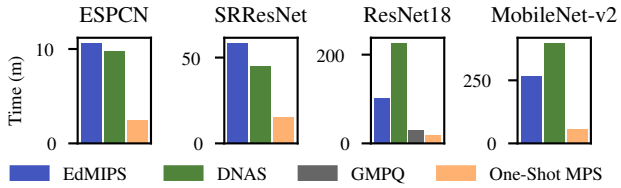


Figure 1. The searching time taken by each algorithm to discover a single bit width architecture belonging to a Pareto front. EdMIPS [7], DNAS [44], GMPQ [42], and One-Shot MPS (our) use a proxy dataset for ResNet-18 and MobileNet-v2. Bayesian Bits [38] and HAQ [41] roughly take 100 times more searching time compared to our method.

To attain the best mixed-precision performance, it is crucial to find an optimal precision for each matrix multiplication factor. Unfortunately, all possible bit width combinations cannot be examined since the search space scales exponentially with the number of multiplications. The inability to predict the influence of individual loss coefficients on the resulting compression rate furthermore exacerbates the difficulty of searching. Existing methods [44, 7, 38, 9] require multiple restarts of the searching process until a satisfactory bit width allocation is found. This results in $O(N)$ searching time, where N is the number of restarts.

The authors of EdMIPS [7] mention that "sometimes, it is mysterious why and how an architecture is found by Neural Architecture Search (NAS)". We answer this question in the context of EdMIPS and DNAS [44] methods. To do so, we simplify and generalize Bayesian Bits [38], where a variational inference (VI) approach is used to derive the loss function for a hierarchical supernet. Then, we demonstrate how the EdMIPS and DNAS loss functions can be derived.

Next, using our derivation, we propose a novel One-Shot Mixed-Precision Search (One-Shot MPS) method that finds a diverse set of Pareto-front architectures in $O(1)$ time. We extend the commonly used supernet transforma-

tion of a floating-point model with a set of trainable functions that predict the bit width probability depending on a hardware regularization parameter. For ImageNet [11], we observe at least 0.96 correlation score between *child* models sampled from a One-Shot model and *standalone* fine-tuned models, *cf*. previous result attains a maximum score of 0.55 [16]. Such a high score allows one to plot a Pareto front of performance versus hardware resources using a linear sweep over the regularization parameter, and select the most promising precision given hardware constraints *before fine-tuning*. The Pareto-front architecture selection is straightforward and takes only 20 to 40 supernet evaluations while existing One-Shot methods require at least 1000 evaluations [16, 10].

To sum up, our contribution is twofold. First, we provide a theoretical derivation of the earlier empirically-found state-of-the-art searching methods. Second, we propose to augment a supernet with a bit width prediction model that allows searching for Pareto-front bit width combinations corresponding to different compression rates in a constant time. We validate the benefits of the approach on several widely-used models including mobile-friendly architectures. To the best of our knowledge, the proposed predictor is not described in the existing literature.

2. Related work

The existing search methods can be split into four groups. The first group [12, 28, 8, 34] uses suitable proxy metrics that reflect model sensitivity to quantization. In [12], it is shown that an average Hessian trace can suggest a relative bit width for each layer. The Bayesian approach [28] uses a posterior uncertainty to identify and remove insignificant bits.

The second group of methods leverages Reinforcement Learning (RL) [14, 41, 27]. Such methods assume that to converge in a reasonable time, the optimal solution should abide by resource constraints while attaining the best performance after only several fine-tuning epochs. The advantage of RL is that it works with non-differentiable feedback from a target device, thereby finding a solution adapted to all the particular features of the hardware at hand.

The third and largest group of methods uses various techniques to relax the discrete problem of choosing integer bit widths into a continuous problem. This enables updating architecture parameters using Stochastic Gradient Descent (SGD). After searching, the most likely tensor precision is selected. Methods inspired by differentiable NAS construct a supernet in which each matrix multiplication factor is represented as a weighted sum of mixed-precision quantizers applied to an original FP32 factor [44, 7, 42]. A similar approach is to take a linear combination of two integer quantizers [46, 19] or bit widths [13]. In [37, 9], the authors reparameterize quantizers with a trainable step and dynamic range, achieving a legitimate bit width after rounding. The Bayesian Bits [38] method recursively decomposes a residual error between the quantized and FP32 tensors. The process involves the relaxation of stochastic gates that, if open, double a tensor precision. Methods in this group typically use a regularization parameter in a loss function to balance a task performance and a compression rate. The parameter has to be selected *in advance*, and generally, several restarts should be made before a suitable parameter is found.

Finally, the fourth group of methods uses One-Shot models that can predict the performance of any bit width configuration. These methods may look very similar to supernet methods [44, 7] described above. However, the crucial difference is that One-Shot models allow for the decomposition of the search process into two steps. In the first step, the One-Shot model is trained such that child architectures sampled from it can predict the performance without fine-tuning. After training, the architecture is selected via Evolutionary Search (ES) [16] or a heuristic algorithm [6, 10]. Weight co-adaptation is a well-known issue [1]. Solving this issue is essential for achieving good-quality predictions in the first step. A high correlation between a child and standalone model performance was first achieved by a dropout technique [1]. Later, single-path sampling [16] and varieties of progressive training [6, 10] were shown to be effective. In fact, the latter methods are so successful in combating co-adaptation that the sampled child models are already optimal, *i.e*. they do not require fine-tuning at all.

3. The derivation of DNAS and EdMIPS

3.1. Supernet with one-hot gates

In this section, we show that EdMIPS and DNAS methods can be derived from Bayesian principles using VI. The proposed derivation is inspired by Bayesian Bits [38] where authors use a hierarchical transformation of quantization operations. Our derivation uses a much simpler transformation that is generic and easier to apply. In particular, Bayesian Bits supports only power-of-two bit width options, while our method supports any bit width.

We assume that we are given a regular model with FP weights ω. The model is trained to maximize the log-likelihood $\log p_\omega(\mathcal{D})$ on a supervised dataset $\mathcal{D}$. Our goal is to find the optimal bit width of weights $b^\omega \in \mathcal{B}^\omega$ and activations $b^x \in \mathcal{B}^x$ subject to hardware constraints such as a limit on memory or bit operations (BOPs). For simplicity, we often use a Cartesian product of bit options $\mathcal{B} = \mathcal{B}^x \times \mathcal{B}^\omega$.

We consider the class of methods that find a suitable architecture using gradient descent. The proposed supernet block for a single matrix-multiplication operation is shown in Figure 2. In the block, we place discrete categorical random variables $z \in \{0, 1\}^{|\mathcal{B}^*|}$ encoded as one-hot vectors in front of each quantizer Q. The $*$ symbol is used as

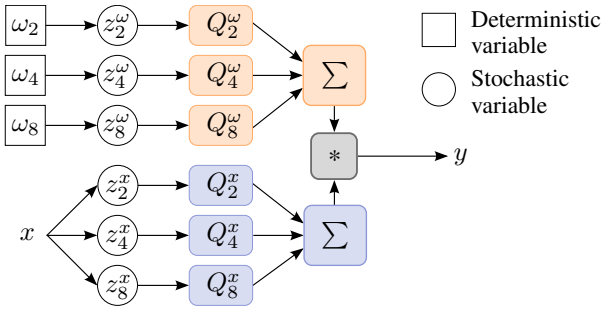


Figure 2. The transformation of a matrix multiplication operation $y = \omega^T x$ in a supernet. One-hot gates z, placed in front of each quantizer Q, determine the weight ω or input tensor x precision. The example is given for quantizers in 2, 4, and 8 bits.

a wildcard for $\{\omega, x\}$. The aforementioned goal can now be specified as finding a posterior probability distribution $p_\omega(z|\mathcal{D}) = p_\omega(\mathcal{D}, z)/p_\omega(\mathcal{D})$ of the gates being open, such that the supernet maximizes the log-likelihood subject to hardware constraints. Calculating the marginal density of observations $p_\omega(\mathcal{D}) = \int p_\omega(\mathcal{D}, z)dz$ is intractable. Thus, we resort to the VI approach of approximating $p_\omega(z|\mathcal{D})$ by some variational distribution $q_\pi(z)$ with parameters π. In other words, we aim at minimizing the KL-divergence between the variational distribution and the posterior,

$$\mathrm{KL}(q_\pi(z)||p_\omega(z|\mathcal{D})) = \mathbb{E}_{z\sim q_\pi(z)}\left[\log \frac{q_\pi(z)}{p_\omega(\mathcal{D}, z)}\right] + \log p_\omega(\mathcal{D}) = -\mathcal{F}(\omega, \pi) + \log p_\omega(\mathcal{D}), \quad (1)$$

where $\mathcal{F}(\omega, \pi)$ is known as the evidence lower bound (ELBO). Minimizing the left-hand side equals maximizing ELBO because the log-evidence term $\log p_\omega(\mathcal{D})$ is constant w.r.t. q. Therefore, our goal of finding the posterior $p_\omega(z|\mathcal{D})$ reduces to maximizing ELBO,

$$\mathcal{F}(\omega, \pi) = \mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right] - \mathrm{KL}(q_\pi(z)||p_\omega(z)). \quad (2)$$

The first term minimizes task loss. This leads to a higher bit width selection and a smaller quantization error. The second term is a prior-based regularizer. We select it to encourage modest resource utilization by penalizing the opening of high-precision gates. The prior probability of a gate z_k being open in layer k at bit width b depends on the amount of resources each gate adds to the total consumed resources. Let us denote this amount as $h_{k,b}$. The Bayesian approach is to view $h_{k,b}$ as state energy, where the state is defined by the bit width allocation. Hence, we propose to model the gate-opening prior by a Boltzmann distribution,

$$p_{k,b} = \frac{e^{-\eta h_{k,b}}}{\sum_{b\in\mathcal{B}} e^{-\eta h_{k,b}}}. \quad (3)$$

Here, $\eta \geq 0$ is a hardware penalty that plays a role of an inverse temperature. Intuitively, (3) promotes bit widths that result in smaller resources h for $\eta > 0$. The prior does not give any preference to any particular gate when $\eta = 0$. Note how simple and generic it becomes to model the prior probability using any hardware metric.

The joint prior over the gates in layer k is modeled as a Categorical distribution because only a single gate z can be opened per a multiplication factor,

$$p_\omega(z_k) = \mathrm{Cat}(z_k; p_{k,b}) = \prod_{b\in\mathcal{B}} p_{k,b}^{I(z_k = z_{k,b})}. \quad (4)$$

We use the mean-field approximation for the variational distribution $q_\pi(z) = \prod_{k=1}^K q_\pi(z_k)$ defined by parameters $\{\pi_k\}_{k=1}^K$. Once again, we choose a Categorical distribution for independent factors of $q_\pi(z)$,

$$q_\pi(z_k) = \mathrm{Cat}(z_k; \pi_k) = \prod_{b\in\mathcal{B}} \pi_{k,b}^{I(z_k = z_{k,b})}. \quad (5)$$

The KL-divergence in (2) is calculated as

$$\mathrm{KL}(q_\pi(z)||p_\omega(z)) = \sum_{k=1}^K \mathrm{KL}(q_\pi(z_k)||p_\omega(z_k)), \quad (6)$$

where

$$\begin{aligned}
&\mathrm{KL}(q_\pi(z_k)||p_\omega(z_k)) = \mathbb{E}_{z_k\sim q_\pi(z_k)}\left[\log q_\pi(z_k) - \log p_\omega(z_k)\right] \\
&= \mathbb{E}_{z_k\sim q_\pi(z_k)}\left[\sum_{b\in\mathcal{B}} I(z_k = z_{k,b}) \log \pi_{k,b} - \sum_{b\in\mathcal{B}} I(z_k = z_{k,b}) \log p_{k,b}\right] \\
&= \sum_{b\in\mathcal{B}} \mathbb{E}_{z_k\sim q_\pi(z_k)}\left[I(z_k = z_{k,b})\right] \log \pi_{k,b} - \sum_{b\in\mathcal{B}} \mathbb{E}_{z_k\sim q_\pi(z_k)}\left[I(z_k = z_{k,b})\right] \log p_{k,b} \\
&= -H(\pi_k) - \sum_{b\in\mathcal{B}} \pi_{k,b} \log p_{k,b}. \quad (7)
\end{aligned}$$

By putting everything together, we obtain the novel Variational Inference Mixed-Precision Search (VIMPS) method

$$\mathcal{F}(\omega, \pi) = \mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right] + \sum_{k=1}^K \sum_{b\in\mathcal{B}} \pi_{k,b} \log p_{k,b} + H(\pi), \quad (8)$$

where $H(\pi) = \sum_{k=1}^K H(\pi_k)$ for simplicity. VIMPS is used further to derive DNAS, EdMIPS, and One-Shot MPS.

3.2. DNAS and EdMIPS loss functions

We would like to maximize (8) with respect to the model parameters ω and variational parameters π. However, it

is problematic to propagate gradients through the expected conditional distribution $\mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right]$ to update π.

The first approach is to approximate the posterior $q_\pi(z)$ using a differentiable Concrete distribution [20, 29],

$$\mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right] \triangleq \mathbb{E}_{g\sim\text{Gumbel}(0,1)}\left[\log p_\omega(\mathcal{D}|z^g)\right], \quad (9)$$

where $z^g = \text{Softmax}\left((l+g)/\tau\right)$ is a sample from the Concrete distribution, l is trainable architecture parameters, g is a Gumbel sample, and τ is a temperature. This approach leads to the DNAS bit width searching method [44].

Another approach is to discard sampling and simply use a Softmax function as a proxy for the gate probabilities,

$$\mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right] \triangleq \log p_\omega(\mathcal{D}|\,\text{Softmax}(l)) \quad (10)$$
$$= \log p_{\tilde{\omega}}(\mathcal{D}), \quad (11)$$

where the convoluted weights and activations are calculated as $\tilde{\omega} = \sum_{b\in\mathcal{B}^\omega}\text{Softmax}(l_b^\omega)Q_b^\omega(\omega)$ and $\tilde{x} = \sum_{b\in\mathcal{B}^x}\text{Softmax}(l_b^x)Q_b^x(x)$. This approximation is crude, but it works well in practice. This approach results in EdMIPS [7]. Note that due to the reparametrizations (9) and (11), logits l become the actual trainable parameters of the variational distribution $q_\pi(z)$.

Finally, let us write the exact form of (8) that, for example, minimizes BOPs. As explained in Section S3.2 of Supplementary Materials, BOPs in layer k depend on both weight and activation bit width, $\text{BOPs}(k) = b_k^\omega b_k^x\,\text{MACs}(k)$. When gate $z_{k,b}^\omega$ ($z_{k,b}^x$) is open, the weight (activation) tensor in layer k will be quantized in b^ω (b^x) bits. Therefore, the opening probability should reduce if a tensor bit width is increasing. This can be formalized as

$$h_{k,b} = b_k^\omega b_k^x\,\text{MACs}(k). \quad (12)$$

In such a case, (8) now will have the following form:

$$\mathcal{F}(\omega,\pi) = \mathbb{E}_{z\sim q_\pi(z)}\left[\log p_\omega(\mathcal{D}|z)\right] - \eta\sum_{k=1}^{K}\tilde{b}_k^\omega\tilde{b}_k^x\,\text{MACs}(k) + H(\pi), \quad (13)$$

where $\tilde{b}^* = \sum_{b^*\in\mathcal{B}^*}\pi_{b^*}b^*$ is the expected bit widths of weights and activations, and z is sampled from the Concrete distribution similarly to DNAS. The exact derivation of (13) is given in Section S1 of Supplementary Materials.

Let us summarize the differences between our theoretically derived loss (13) and DNAS and EdMIPS losses.

1. Both DNAS and EdMIPS do not use entropy.
2. DNAS uses a multiplicative hardware loss.
3. EdMIPS uses a crude approximation of the expected conditional distribution.

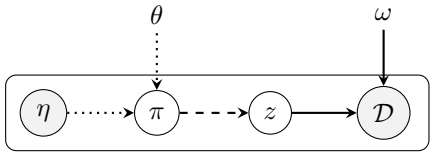


Figure 3. The graphical model of the proposed One-Shot MPS method. Solid lines denote the generative model $p_\omega(\mathcal{D}|z)p_\omega(z)$, dashed line denotes the variational approximation $q_\pi(z)$, and dotted lines denote the discriminative model $f_\theta(\eta)$.

Note that the provided derivation is generic in that it applies to any differentiable NAS methods where branch probabilities sum up to one.

4. One-Shot Mixed-Precision Search

4.1. The bit width probability model

To discover the range of models subject to various compression rates at once (in $O(1)$ time), we propose to expand the supernet block from Figure 2 with a discriminative model $\pi(\eta) = \text{Softmax}(f_\theta(\eta))$ parametrized by weights θ. We call it the bit width probability model because, given the hardware penalty η, the model outputs the opening probability of each gate in the supernet. The graphical model of the proposed supernet modification is shown in Figure 3. After training, one can select Pareto front architectures simply by sweeping the hardware penalty through the model linearly. ELBO can now be written as

$$\mathcal{F}(\omega,\theta) = \mathbb{E}_{\eta\sim p(\eta)}\left[\mathbb{E}_{z\sim q_{\pi(\eta)}(z)}\left[\log p_\omega(\mathcal{D}|z)\right] + \sum_{k=1}^{K}\sum_{b\in\mathcal{B}}\pi_{k,b}(\eta)\log p_{k,b}(\eta) + H(\pi(\eta))\right], \quad (14)$$

where the penalty η is sampled from some distribution $p(\eta)$.

4.2. One-Shot MPS loss function

Although the original EdMIPS or DNAS models do not use the entropy term, we have found that using this term is beneficial for One-Shot MPS. The intuition of its usefulness is the following. The Softmax function used in the bit width model fails to learn when logits become large, causing gradients to vanish. The entropy prevents Softmax saturation, and therefore it facilitates gradient flow. However, too strong entropy regularization leads to a discrepancy between a supernet and its child. Experimentally, we have found that weighting the entropy by some small and constant value λ works the best. Also, note that the effect of the temperature τ in (9) is different from the entropy loss. While the former works approximately as a gradient multiplier, the latter works as an additive regularizer that prevents Softmax saturation.

The One-Shot model is expected to work at different bit width options within the same layer. For this reason, the model has an independent set of weights for each layer, *cf*. Figure 2. Experimentally, independent weights may result in One-Shot model collapse because of a positive feedback loop in early training: 8-bit weights perform much better than 2-bit weights due to a smaller quantization error after initialization. The positive feedback loop reinforces 8-bit gate opening, and the model is not trained well. Our solution is to add an extra term called kernel similarity that helps a quicker 2-bit weights adjustment *at the start of training*.

This leads us to the One-Shot MPS loss function,

$$\mathcal{L}(\mathcal{D};\omega,\theta) = -\mathbb{E}_{\eta\sim p(\eta)}\Bigg[\mathbb{E}_{z\sim q_{\pi(\eta)}(z)}\left[\log p_\omega(\mathcal{D}|z)\right] + \sum_{k=1}^{K}\sum_{b\in\mathcal{B}}\pi_{k,b}(\eta)\log p_{k,b}(\eta) + \lambda H(\pi(\eta))\Bigg] + \mu\sum_{i\in\mathcal{B}^\omega}\sum_{\substack{j\in\mathcal{B}^\omega\\ j>i}}\|\omega_i-\omega_j\|_2, \quad (15)$$

where the last term is the kernel similarity weighted by some penalty μ. We reduce μ to 0 during training using a cosine decay. Throughout our experiments, we use the same constant entropy λ and initial kernel similarity μ_0 penalties.

Loss (15) is differentiable with respect to model weights ω due to the Straight-Through Estimator (STE) [2]. It is also differentiable with respect to bit width model parameters θ by the supernet construction. We update both ω and θ simultaneously, using a training dataset.

Finally, what distribution should we choose for $p(\eta)$? Since the scales of terms in (15) are not known in advance, the proposed method requires the upper η_1 and lower η_0 bounds on the hardware penalty. Given the bounds, we draw samples exponentially from the interval $[\eta_0, \eta_1]$. Such a procedure is often used for hyperparameter optimization [3]. For $\eta_1/\eta_0 \gg 1$, which is the case in all One-Shot MPS experiments, the samples approximately follow the exponential distribution. This distribution is a maximum entropy distribution on $[0, +\infty)$, which means that we make the fewest assumptions about the true distribution of η. In practice, we sample $\hat{\eta} \sim \text{Uniform}(\ln\eta_0, \ln\eta_1)$, and use a normalized value of $\hat{\eta}$ as input to the bit width model, $l = f_\theta((\hat{\eta} - \ln\eta_0)/(\ln\eta_1 - \ln\eta_0))$. The hardware penalty in (15) is then computed as $\eta = e^{\hat{\eta}}$.

4.3. Conditional Batch Normalization and biases

Conditional Batch Normalization (CBN) is another feature we propose to use in One-Shot MPS. Several authors have noticed that changing model architecture or bit width while training results in serious performance degradation if a network contains Batch Normalization (BN) layers [47, 21, 6]. This degradation is attributed to a shift in feature statistics. Note that the feature distribution in DNAS or EdMIPS is stationary because η is fixed.

To cope with the changing statistics, we expand BN layers by the factor of $|\mathcal{B}|$ and use the incurred tensor product $c = \pi^x \otimes \pi^\omega$, combined with a moving average momentum m, to "softly" update the BN layers,

$$x_{k+1} = \text{CBN}(x_k, \pi^x, \pi^\omega) = \sum_{c_i\in\{\pi^x\otimes\pi^\omega\}} c_i\,\text{BN}_i(x_k, c_i m), \quad (16)$$

where BN statistics depend on input and weight precision probabilities c_i used in a previous matrix multiplication operation. Note that $\sum_i c_i = 1$.

Similarly, we expand biases whenever they are used in an FP32 model. The described expansion results in a negligible number of extra parameters, and can be fused with convolutional and fully-connected layers at inference [47, 31].

5. Experiments

5.1. Experiment setup

We consider searching for hardware-friendly architectures with power-of-two bit width options. For the convenience of comparison with Bayesian Bits, we consider 2, 4, or 8 bit quantizers for all layers in a model, including the first and last layer. We set the lower range value for input x to zero if it follows a ReLU activation function. Residual connections are not quantized [18, 15].

The proposed method is evaluated on ResNet-18 [17] and MobileNet-v2 [32], two widely-used classification models, and on two super-resolution (SR) models. The ESPCN [33] network is a real-time SR network with three convolutional layers. We use it to fine-tune all $3^6 = 729$ bit width combinations and demonstrate the optimal One-Shot MPS performance. Quantizing the second SR model, SRResNet [23], is known to be a challenging task [40, 18].

For ResNet-18 (MobileNet-v2) training, we use the ImageNet [11] dataset with a standard augmentation pipeline [17] and 512 (256) batch size. The top-1 accuracy is calculated on a validation dataset. Proxy ImageNet is prepared according to [41]. For SR models, we use 32×32 pixel low-resolution images from DIV2K [36] dataset with augmentations from [40]. Images are upscaled 4 times. Models are tested on Set5 [4] and Set14 [48] datasets.

For all experiments with the One-Shot model, we use initial kernel similarity $\mu_0 = 1$ with cosine annealing, a constant entropy weight $\lambda = 10^{-3}$, and a constant temperature $\tau = 1$. For modeling the bit width probability π, we use a densely-connected n-layered neural network $f_\theta(\eta)$ with 128 hidden layers and a swish activation function. For ESPCN, we set $n = 1$, while for other networks, we use $n = 2$. Please refer to Section S3 of Supplementary Materials for extra details about the experiments and Section S4 for extra results.

5.2. Evaluation criteria

We evaluate the performance of the One-Shot MPS method using three criteria.

1. The primary evaluation criterion is the time required for searching the Pareto front models.

2. The second criterion is the correlation plots and the corresponding Kendall's Tau correlation values between child and standalone models [1, 16]. The values range from -1 to 1. The higher value indicates that child models sampled from a supernet can better predict the relative performance of standalone models without fine-tuning.

3. Finally, we evaluate the quality of architectures by fine-tuning the models found by One-Shot MPS, EdMIPS, DNAS, and Bayesian Bits [38]. The plots show the trade-off between the compression rate and the quantized model performance. We use BOPs to compare our results with Bayesian Bits. However, BOPs cannot directly reflect the latency due to many factors, including memory costs [41, 43]. Thus, we additionally use the total random-access memory (RAM) metric to compare SR Pareto fronts. Two versions of the One-Shot MPS method are compared. The "One-Shot (Lloyd)" version uses a Lloyd quantizer for searching and fine-tuning. The "One-Shot (Trainable Lloyd)" version uses a Lloyd quantizer for searching and a Trainable Lloyd quantizer for fine-tuning.

5.3. Searching costs analysis

The Pareto searching time of all methods is shown in Figure 1. The One-Shot MPS method is run only once. For large models, ResNet-18 and MobileNet-v2, the searching time is 5 times smaller than existing methods.

One-Shot MPS does not share weights within a layer. Therefore, DNAS and One-Shot MPS consume approximately $|\mathcal{B}^x|$ more RAM than EdMIPS and Bayesian Bits.

5.4. Correlation analysis

Kendall's Tau correlation results are presented in Table 1. Child models sampled from One-Shot MPS in all experiments have correlation scores above or equal 0.93, and the result is always significant. In particular, the scores are much higher than those in literature, *e.g.* [16] attains a maximum score of 0.55 on ImageNet. On the other hand, EdMIPS in most cases has smaller and often non-significant correlation values. We explain this by the crude cost function approximation that does not use architecture sub-sampling. Contrary to EdMIPS, DNAS uses architecture sub-sampling by the means of a Concrete distribution. We observe that DNAS attains higher correlation scores, although they are still smaller than for One-Shot MPS.

Table 1. The Kendall's Tau correlation score is calculated between the child and standalone models' performance. The *child* model is derived from a supernet, where the branch with the highest π is taken. The *standalone* model is a fine-tuned model of the same bit width. The Lloyd quanitzer is used in these experiments. The boldface marks the statistically significant result (p-value ≤ 0.05).

Network	Method	Kendall's Tau correlation measured for Set14 PSNR	for Set5 PSNR
ESPCN	One-Shot	**0.97**	**0.97**
	EdMIPS	**0.71**	0.52
	DNAS	**0.86**	**0.93**
SRResNet	One-Shot	**0.93**	**0.93**
	EdMIPS	0.07	0.14
	DNAS	**0.91**	**0.91**
		for ImageNet Top-1 accuracy	
ResNet-18	One-Shot	**0.97**	
	EdMIPS	0.29	
	DNAS	0.52	
MobileNet-v2	One-Shot	**0.96**	
	EdMIPS	**0.93**	
	DNAS	0.29	

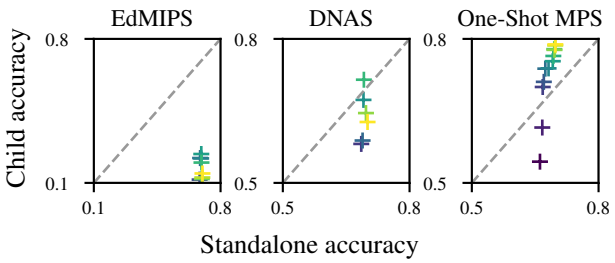
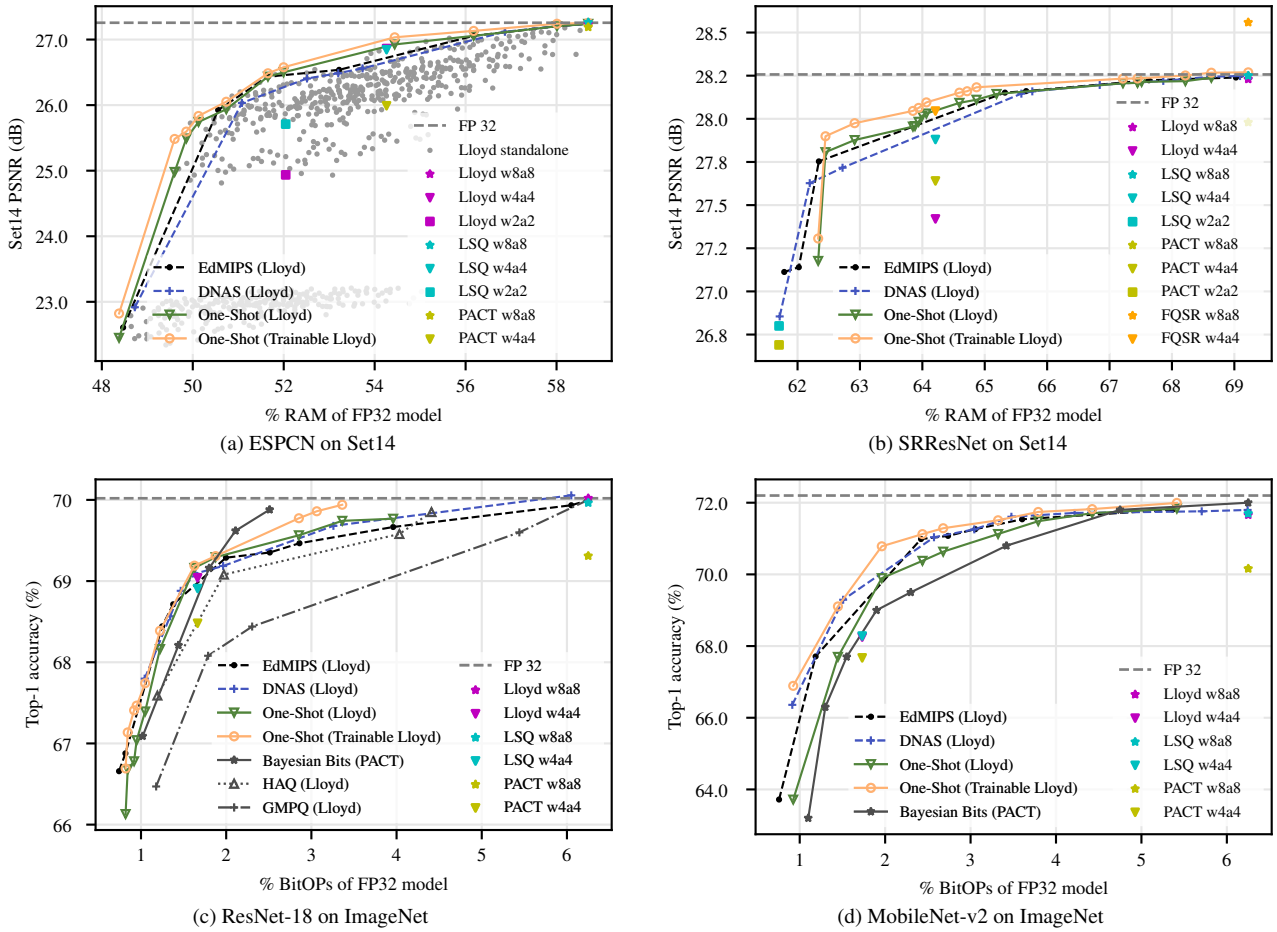


Figure 4. The plots visualize the correlation between the child and standalone models in terms of top-1 accuracies for ResNet-18. Brighter colors depict models of higher BOPs. The standalone model is trained and evaluated on a full ImageNet, while the child model is taken from a One-Shot supernet that is trained and evaluated on a proxy ImageNet. Note the difference in axis ranges.

Figure 4 depicts the correlation between the child and standalone models in terms of top-1 accuracies. It can be seen that One-Shot MPS finds child architectures that correlate well with standalone architectures.

5.5. The quality of found architectures

The architecture quality found by One-Shot MPS is studied in Figure 5, where the performance of fine-tuned models is compared. ESPCN plot shows that One-Shot MPS finds the optimal architectures with respect to performance vs RAM trade-off. In most cases, One-Shot MPS attains performance that is similar to or higher than existing methods. Thus, training the bit width model does not hamper the quality of found architectures. Furthermore, found architectures are generic w.r.t. a fine-tuning quantizer, *cf*. "Lloyd" and "Trainable Lloyd". Finally, note that One-Shot MPS finds a richer set of models compared to Bayesian Bits.



(a) ESPCN on Set14

(b) SRResNet on Set14

(c) ResNet-18 on ImageNet

(d) MobileNet-v2 on ImageNet

Figure 5. The verification of architecture qualities found by One-Shot MPS. For a cleaner visualization, we removed models which rest inside Pareto fronts. The notation "wXaY" indicates a fixed bit width architecture with 8-bit model input, X-bit weights, and Y-bit activations. The FP32 ESPCN and SRResNet models' RAM is 1.28 MB and 26.75 MB, respectively. The FP32 ResNet-18 and MobileNet-v2 model' BOPs are 1857.6 GBitOPs and 308.0 GBitOPs, respectively. The FQSR result is taken from [40]. The Bayesian Bits [38] result is taken from the arXiv article. All other results, including HAQ [41] and GMPQ [42], are obtained by the authors.

Fixed bit width architectures attain peak signal-to-noise ratio (PSNR) or Top-1 accuracy similar to mixed-precision networks only at 8 bits. For smaller bit widths, One-Shot MPS and conventional mixed-precision methods typically find better-performing architectures.

5.6. Ablation studies

Ablation results in Figure 6 are obtained for a shorter fine-tuning time and a smaller number of architecture samples. In the *left* plot, we compare the exponential temperature decay $5.0 \rightarrow 0.5$ suggested in [44] with a faster and stronger exponential decay of $5.0 \rightarrow 10^{-3}$, and a constant value. The plot shows that there is no difference in temperature choices. In the *middle* plot, we see that $\lambda = 10^{-3}$ and $\mu_0 = 1$ diversify found architectures, but do not alter their performance substantially. Note, how the absence ($\lambda = 0$) of the entropy term in (14) reduces the spectrum of found models. In the *right* plot, we see that a child model with CBN makes a better PSNR prediction than with BN. Kendall's Tau scores for One-Shot MPS with the CBN and BN layers are 0.81 and 0.52, respectively.

Figure 7 investigates the bit width model $f_\theta(\eta)$. In the *left* plot, the deeper ($n = 2$) bit width model has a higher confidence of a selected bit width (probabilities closer to 1) and a greater architecture diversity (2, 4, and 8 bits are selected for different penalty values η). Such behavior appears because the linear model ($n = 1$) has less capacity (high bias) to learn the dependency of a bit width distribution on penalty η compared to a nonlinear model ($n = 2$).

The *right* plot shows the experiment in which the bit width model was fit to the whole penalty range and its halves in a log-uniform space. As we see, a wider penalty range does not hamper the quality of found architectures.

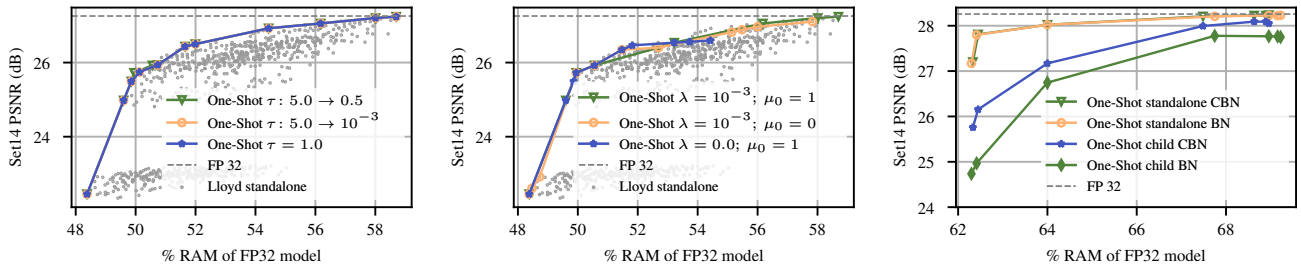


Figure 6. The influence of temperature τ (*left*), entropy penalty λ, and initial kernel similarity penalty μ_0 (*middle*) on ESPCN performance. The notation "$X \rightarrow Y$" in temperature indicates the exponential schedule, with X and Y being the starting and final temperature values. The *right* plot studies the influence of CBN vs. BN on child and standalone SRResNet models quantized.

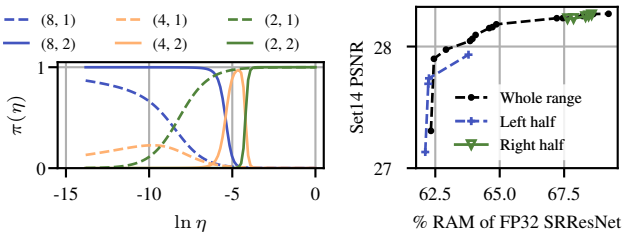


Figure 7. The *left* plot depicts the bit width probability $\pi(\eta)$ depending on the number of layers in a bit width model. The notation "(b, n)" denotes the bit width b probability for the bit width model of n layers. The *right* plot shows that a wider hardware penalty range does not hamper the quality of found architectures.

6. Discussion

Our results demonstrate that the proposed One-Shot MPS method can find good-quality architectures at once. Compared to the conventional methods, our method

1. allows finding a richer set of bit width combinations,
2. improves a Kendall's tau correlation which is useful for predicting the fine-tuned model performance, and
3. does not hamper the quality of found architectures.

Benefits 1 and 2 arise because the bit width model imposes a structure on the selected architectures, *cf*. Figure 7 (*left*). After searching, one may use an educated guess or a binary search to select one or several models that satisfy both performance and compression constraints.

One-Shot MPS requires two extra hyperparameters in loss, *i.e*. the kernel similarity μ and entropy λ penalties. However, our results, obtained in four very different model architectures, show that the default values work well.

During experiments, it was easy to choose the regularization bounds $[\eta_0, \eta_1]$ for all models except SRResNet, *i.e*. moderate compression rates were not covered by the selected architectures. We attribute this to the largest residual connection between the low-level feature extractor and the up-sampler parts [24]. Using the two-layered perceptron allowed us to obtain more diverse bit width combinations.

Mixed-precision searching methods are sensitive to hyperparameters. Although during fine-tuning, we used the same learning rates for models of different bit widths, we observed that tuning the learning rates for a particular bit width may increase model performance. Thus, we would like to stress the necessity of developing quantizers and optimizers that would adapt and perform equally well in various mixed-precision configurations.

Limitations. First, the One-Shot MPS approach, similar to other searching methods considered here, assumes that hardware metrics can be calculated by an equation, and the equation itself is differentiable w.r.t. bit width. Modeling some interesting metrics such as latency or power consumption is usually non-trivial and may pose a difficulty for a differentiable NAS [39]. Second, the proxy dataset used for searching the architectures of ResNet-18 and MobileNet-v2 by One-Shot MPS may not match the full dataset well. Thus, the trained One-Shot model may result in biased predictions. Finally, the memory grows linearly with the number of bit width options. This may become a limiting factor for applications to very large models such as transformers.

Potential negative societal impact. Quantization may reduce robustness against adversarial attacks [30, 26].

7. Conclusion

We theoretically derived the DNAS and EdMIPS bit width searching methods using variational inference. Using our derivation, we proposed and experimentally verified the novel bit width searching method, One-Shot MPS. The method uses the Boltzmann distribution for hardware constraints modeling and CBN for improving the correlation scores between the child and standalone models. One-Shot MPS finds good-quality architectures in $O(1)$ time compared to conventional methods for which the searching time depends linearly on the number of architectures.

References

[1] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and Simplifying One-Shot Architecture Search. In *Proceedings of the International Conference on Machine Learning*, pages 550–559, 2018.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv:1308.3432*, 2013.

[3] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-line Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *Proceedings of the British Machine Vision Conference*, 2012.

[5] Alex Bie, Bharat Venkitesh, Joao Monteiro, Md Akmal Haidar, and Mehdi Rezagholizadeh. A Simplified Fully Quantized Transformer for End-to-end Speech Recognition. *arXiv:1911.03604*, 2020.

[6] Adrian Bulat and Georgios Tzimiropoulos. Bit-Mixer: Mixed-Precision Networks with Runtime Bit-Width Selection. In *Proceedings of IEEE International Conference on Computer Vision*, 2021.

[7] Zhaowei Cai and Nuno Vasconcelos. Rethinking Differentiable Search for Mixed-Precision Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2020.

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

[9] Vladimir Chikin, Kirill Solodskikh, and Irina Zhelavskaya. Explicit model size control and relaxation via smooth regularization for mixed-precision quantization. In *European Conference on Computer Vision*, pages 1–16. Springer, 2022.

[10] Yufei Cui, Ziquan Liu, Wuguannan Yao, Qiao Li, Antoni B. Chan, Tei-wei Kuo, and Chun Jason Xue. Fully Nested Neural Network for Adaptive Compression and Quantization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2021.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[12] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian Aware Trace-Weighted Quantization of Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 18518–18529, 2020.

[13] Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable Model Compression via Pseudo Quantization Noise. *arXiv:2104.09987*, 2021.

[14] 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(5):37–45, 2020.

[15] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned Step Size Quantization. In *Proceedings of the International Conference on Learning Representations*, 2020.

[16] 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 *Proceedings of the European Conference on Computer Vision*, pages 544–560, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Cheeun Hong, Heewon Kim, Sungyong Baik, Junghun Oh, and Kyoung Mu Lee. DAQ: Channel-Wise Distribution-Aware Quantization for Deep Image Super-Resolution Networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 913–922, 2022.

[19] Xijie Huang, Zhiqiang Shen, Shichao Li, Zechun Liu, Hu Xianghong, Jeffry Wicaksana, Eric Xing, and Kwang-Ting Cheng. SDQ: Stochastic Differentiable Quantization with Mixed Precision. In *International Conference on Machine Learning*, pages 9295–9309, 2022.

[20] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of the International Conference on Learning Representations*, 2017.

[21] Qing Jin, Linjie Yang, and Zhenyu Liao. AdaBits: Neural Network Quantization With Adaptive Bit-Widths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[22] Griffin Lacey, Graham W Taylor, and Shawki Areibi. Stochastic Layer-Wise Precision in Deep Neural Networks. *arXiv:1807.00942*, 2018.

[23] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[24] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. PAMS: Quantized Super-Resolution via Parameterized Max Scale. In *Proceedings of the European Conference on Computer Vision*, pages 564–580, 2020.

[25] 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:2102.05426*, 2021.

[26] Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. In *Proceedings of the International Conference on Learning Representations*, 2019.

[27] Qian Lou, Feng Guo, Minje Kim, Lantao Liu, and Lei Jiang. AutoQ: Automated Kernel-Wise Neural Network Quantization. In *Proceedings of the International Conference on Learning Representations*, 2020.

[28] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian Compression for Deep Learning. *Proceedings of the Neural Information Processing Systems*, 2017.

[29] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *Proceedings of the International Conference on Learning Representations*, 2017.

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the International Conference on Learning Representations*, 2018.

[31] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A White Paper on Neural Network Quantization. *arXiv:2106.08295*, 2021.

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[34] Chen Tang, Kai Ouyang, Zhi Wang, Yifei Zhu, Wen Ji, Yaowei Wang, and Wenwu Zhu. Mixed-Precision Neural Network Quantization via Learned Layer-Wise Importance. In *Computer Vision - ECCV 2022: 17th European Conference*, pages 259–275, Berlin, Heidelberg, 2022. Springer-Verlag.

[35] G.K. Thiruvathukal, Y.H. Lu, J. Kim, Y. Chen, and B. Chen. *Low-Power Computer Vision: Improve the Efficiency of Artificial Intelligence*. Taylor & Francis Limited, 2022.

[36] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, Xintao Wang, Yapeng Tian, Ke Yu, Yulun Zhang, Shixiang Wu, Chao Dong, Liang Lin, Yu Qiao, Chen Change Loy, Woong Bae, Jaejun Yoo, Yoseob Han, Jong Chul Ye, Jae-Seok Choi, Munchurl Kim, Yuchen Fan, Jiahui Yu, Wei Han, Ding Liu, Haichao Yu, Zhangyang Wang, Honghui Shi, Xinchao Wang, Thomas S. Huang, Yunjin Chen, Kai Zhang, Wangmeng Zuo, Zhimin Tang, Linkai Luo, Shaohui Li, Min Fu, Lei Cao, Wen Heng, Giang Bui, Truc Le, Ye Duan, Dacheng Tao, Ruxin Wang, Xu Lin, Jianxin Pang, Jinchang Xu, Yu Zhao, Xiangyu Xu, Jinshan Pan, Deqing Sun, Yujin Zhang, Xibin Song, Yuchao Dai, Xueying Qin, Xuan-Phung Huynh, Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, Vishal Monga, Cristovao Cruz, Karen Egiazarian, Vladimir Katkovnik, Rakesh Mehta, Arnav Kumar Jain, Abhinav Agarwalla, Ch V. Sai Praveen, Ruofan Zhou, Hongdiao Wen, Che Zhu, Zhiqiang Xia, Zhengtao Wang, and Qi Guo. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1110–1121, 2017.

[37] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier García, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed Precision DNNs: All You Need Is a Good Parametrization. In *Proceedings of the International Conference on Learning Representations*, 2020.

[38] Mart Van Baalen, Christos Louizos, Markus Nagel, Rana Ali Amjad, Ying Wang, Tijmen Blankevoort, and Max Welling. Bayesian Bits: Unifying Quantization and Pruning. *Advances in Neural Information Processing Systems*, 33:5741–5752, 2020.

[39] Manoj Rohit Vemparala, Nael Fasfous, Lukas Frickenstein, Alexander Frickenstein, Anmol Singh, Driton Salihu, Christian Unger, Naveen-Shankar Nagaraja, and Walter Stechele. Hardware-aware mixed-precision neural networks using in-train quantization. In *Proceedings of the British Machine Vision Conference*, 2021.

[40] Hu Wang, Peng Chen, Bohan Zhuang, and Chunhua Shen. Fully Quantized Image Super-Resolution Networks. *arXiv:2011.14265*, 2021.

[41] 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.

[42] Ziwei Wang, Han Xiao, Jiwen Lu, and Jie Zhou. Generalizable mixed-precision quantization via attribution rank preservation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5291–5300, 2021.

[43] Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An Insightful Visual Performance Model for Multicore Architectures. *Communications of the ACM*, 52(4):65–76, Apr. 2009.

[44] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed Precision Quantization of Convnets via Differentiable Neural Architecture Search. *arXiv:1812.00090*, 2018.

[45] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. *arXiv:2004.09602*, 2020.

[46] Linjie Yang and Qing Jin. FracBits: Mixed Precision Quantization via Fractional Bit-Widths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[47] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable Neural Networks. In *Proceedings of the International Conference on Learning Representations*, 2019.

[48] Roman Zeyde, Michael Elad, and Matan Protter. On Single Image Scale-Up Using Sparse-Representations. In *Proceedings of the Curves and Surfaces*, pages 711–730, 2012.

[49] Kang Zhao, Sida Huang, Pan Pan, Yinghan Li, Yingya Zhang, Zhenyu Gu, and Yinghui Xu. Distribution Adaptive

INT8 Quantization for Training CNNs. *arXiv:2102.04782*, 2021.