EQ-Net: Elastic Quantization Neural Networks Supplementary Material

A. Proof of Theorem

Theorem 1. *Compared to the typical case of normally-distributed weights, uniformly distributed weight tensors have improved tolerance to quantization and lower sensitivity to quantizer implementation. [6]*

Proof: Let Eq(1) be a uniform b-bit quantizer with quantization step size s that maps a continuous value $\boldsymbol{w} \in \mathbb{R}$ into a discrete representation

$$\hat{\boldsymbol{w}} = \operatorname{clip}\left(\left\lfloor \frac{\boldsymbol{w}}{s} \right\rceil + z, -2^{b-1}, 2^{b-1} - 1\right) \tag{1}$$

We consider the expected mean-squared error (MSE) as a local distortion measure that we aim to minimize, expressed as follows:

$$\operatorname{MSE}(\boldsymbol{w}, s) = \mathbb{E}\left[(\boldsymbol{w} - \hat{\boldsymbol{w}})^2\right] \tag{2}$$

Assuming an optimal quantization step size $\tilde{s}$ and quantizer for a given distribution $\boldsymbol{w}$, we quantify the quantization sensitivity $\Gamma(\boldsymbol{w}, \varepsilon)$ as the increase in the $\operatorname{MSE}(\boldsymbol{w}, s)$, resulting from small changes in the quantization step size around $\tilde{s}$. More specifically, for a given $\varepsilon \geq 0$ and a quantization step size s in proximity to $\tilde{s}$ (i.e.,$|s - \tilde{s}| = \varepsilon$), we compute the following difference:

$$\Gamma(\boldsymbol{w}, \varepsilon) = |\operatorname{MSE}(\boldsymbol{w}, s) - \operatorname{MSE}(\boldsymbol{w}, \tilde{s})| \tag{3}$$

Let $\boldsymbol{w}_u$ and $\boldsymbol{w}_n$ be continuous random variables with a uniform distributed and normal distributions. As demonstrated in the proof of RobustQuant [6], assuming a second-order Taylor approximation, the quantization sensitivity can be expressed as follows:

$$\Gamma\left(\boldsymbol{w}_u, \varepsilon\right) \approx \left|\frac{\partial^2 \operatorname{MSE}\left(\boldsymbol{w}_u, s = \tilde{s}\right)}{\partial^2 s} \cdot \frac{\varepsilon^2}{2}\right| = \frac{\varepsilon^2}{4} \tag{4}$$

$$\Gamma\left(\boldsymbol{w}_n, \varepsilon\right) \approx \left|\frac{\partial^2 \operatorname{MSE}\left(\boldsymbol{w}_n, s = \tilde{s}\right)}{\partial^2 s} \cdot \frac{\varepsilon^2}{2}\right| \geq \frac{11\varepsilon^2}{12} \tag{5}$$

The aforementioned conclusion rigorously establishes the theorem, as we have demonstrated that:

$$\Gamma\left(\boldsymbol{w}_u, \varepsilon\right) < \Gamma\left(\boldsymbol{w}_n, \varepsilon\right) \tag{6}$$

B. Pseudocode of EQ-Net Framework

The overall flow of the EQ-Net framework is illustrated in Algorithm 1. The supernet training process is described in detail in lines 1 to 8, while the mixed-precision search process is presented in lines 11 to 18.

C. The Efficiency of EQ-Net Framework.

DNNs consume vast amounts of energy, resulting in a significant carbon footprint. Quantization has emerged as a promising technique to enhance the energy efficiency of neural networks, benefiting both commodity GPUs and specialized accelerators. The EQ-Net framework takes this approach further by generating a single model that can be deployed across a range of inference chips, eliminating the need for retraining prior to deployment and reducing associated CO_2 emissions. As shown in Table 1, EQ-Net outperforms both uniform and mixed-precision quantization methods when handling multiple deployment scenarios, as its cost is constant, while LSQ/HAQ/EdMIPS scales linearly with the number of deployment scenarios (N). Furthermore, EQ-Net can adapt to a broader range of quantization scenarios compared to the B-OFA method.

D. Comparison with SOTA Methods Details

Figure 1, 2, 3 present a comparison of our proposed EQ-Net utilizing the BGS-OFA method with other approaches across different networks. Approaches with the same quantization method are grouped together and plotted as separate curves. As ResNet18 shares the same architecture as ResNet50 [4], we only present results for ResNet18 here. It is worth noting that the majority of mainstream methods employ per-tensor and symmetric quantization, followed by per-tensor and asymmetric, while the per-channel approach is almost absent.

In the case of per-tensor and symmetric quantization, our EQ-Net achieves the optimal performance under three different networks. Specifically, in ResNet18, our method outperforms RobustQuant [6] and CoQuant [7] by over 10% for bit-widths of 2 and 3, respectively. As the quantization bit-width increases, the performance gap between methods narrows, but our method still outperforms RobustQuant by

Algorithm 1 EQ-Net: Elastic Quantization Neural Networks

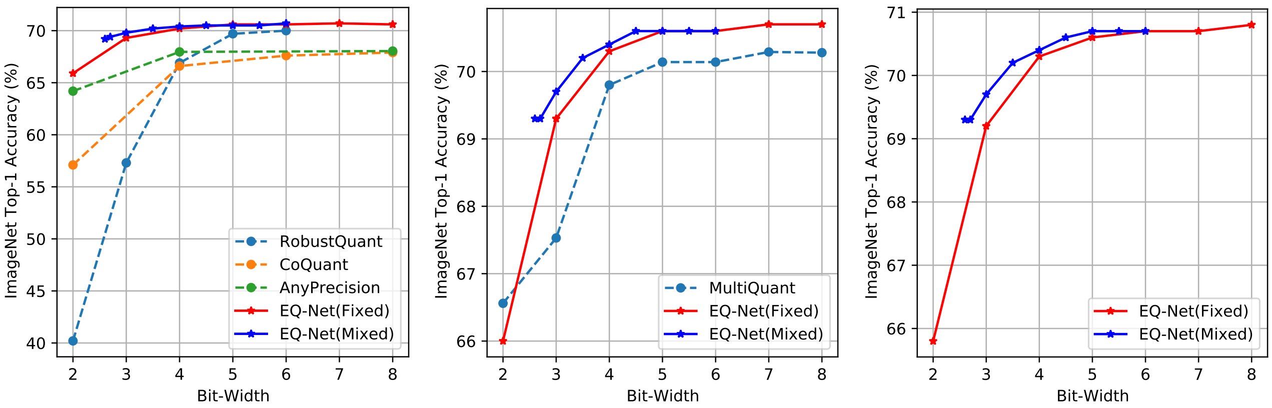
```
Require: Training epochs $E$, Iterations each epoch $I$, Population Size: $\mathcal{S}$, Number of Mutation: $N_M$, Number of Crossover: $N_C$, Max Number of Exploring Iterations: $N$, Gene: $\mathcal{G}_l$, Chromosome: $\mathcal{X}$, Population $\mathcal{O}$, Elite $\mathcal{E}$.
Ensure: Supernet $\text{EQ}_{\text{Net}}$, Each layer bit-width for different quantization configurations of the model $\mathcal{X}_{best}$.
for $e = 0 : E$ do                                                          ▷ Training EQ-Net
    for $i = 0 : I$ do
        Calculate kurtosis and skewness to constrain the weight distribution;   ▷ Eq.(6)
        Forward propagation and loss calculation with $\{\mathcal{L}_H, \mathcal{L}_R, \mathcal{L}_L\}$;   ▷ Eq.(8)
        Compute additional regularization terms and back propagate;
    end for
end for
Get $\text{EQ}_{\text{Net}}$;
Monte Carlo Sampling and Tuning BN to Create Dataset <Config, Accuracy>;
Training Conditional Quantization-Aware Accuracy Predictor for Quantization Model;
$\mathcal{O}_0$=Random($\mathcal{S}$,$\tau_{Bw}$,$\tau_{Ba}$,$\tau_{Gw}$,$\tau_{Sw}$,$\tau_{Sa}$)   ▷ Genetic Algorithm for Mixed-Precision Search
for $n = 0 : N$ do
    $\{\mathcal{O}_n, acc\}$ = CQAP($\mathcal{O}_n$) ;   ▷ Eq.(9)
    $\{\mathcal{E}_n, acc\}$ = TopK($\{\mathcal{O}_0, \mathcal{O}_1, \cdots \mathcal{O}_n; acc\}$);
    $\mathcal{O}_{mutation}$ = Mutation($\mathcal{E}_n, N_M, \tau_{Bw}, \tau_{Ba}$);
    $\mathcal{O}_{crossover}$ = Crossover($\mathcal{E}_n, N_C, \tau_{Bw}, \tau_{Ba}$);
    $\mathcal{O}_{n+1} = (\mathcal{O}_{mutation}, \mathcal{O}_{crossover})$;
end for
$\mathcal{X}_{best}$ = Top1($\{\mathcal{E}_N, acc\}$) $\longrightarrow Acc_{best} = \text{EQ}_{\text{Net}}(\mathcal{X}_{best})$
```

Table 1: Comparison with state-of-the-art quantization method for computation cost on NVIDIA 3090 GPUs. We use N to denote the number of up-coming deployment scenarios. EQ-Net search cost and training cost both stay constant as the number of deployment scenarios grows.

Network	Benchmark	Uniform Quantization	Mixed Quantization	Granularity Quantization	Symmetry Quantization	Search cost (GPU hours)	Training cost (GPU hours)	Total (GPU hours)
ResNet18	LSQ [3]	✓	×	×	×	——	60N	60N
	EdMIPS [2]	×	✓	×	×	10N	60N	70N
	AnyPrecision [11]	✓	×	×	×		76	76
	CoQuant [7]	✓	×	×	×	——	——	——
	RobustQuant [6]	✓	×	×	×	——	214	214
	MultiQuant [10]	✓	✓	×	×	10	134	144
	EQ-Net(Ours)	✓	✓	✓	✓	15	180	195
ResNet50	LSQ [3]	✓	×	×	×	——	240N	240N
	HAQ [9]	×	✓	×	×	95N	192N	287N
	MultiQuant [10]	✓	✓	×	×	48	600	648
	EQ-Net(Ours)	✓	✓	✓	✓	50	672	722
MobileNetV2	LSQ [3]	✓	×	×	×	——	120N	120N
	HAQ [9]	×	✓	×	×	48N	96N	144N
	MultiQuant [10]	✓	✓	×	×	24	310	334
	EQ-Net(Ours)	✓	✓	✓	✓	24	360	384
EfficientNetB0	LSQ [3]	✓	×	×	×	——	130N	130N
	LSQ+ [1]	×	✓	×	×	——	130N	130N
	EQ-Net(Ours)	✓	✓	✓	✓	25	410	435

0.6% at 6-bit and both CoQuant and AnyPrecision [11] by about 2.7% at 8-bit. Similar performance advantages can be observed for MobileNetV2 [5], where our method is nearly 40% better than RobustQuant for 3-bit quantization, with the performance gap shrinking to 2.4% at 6-bit. In EfficientNetB0 [8], our EQ-Net outperforms LSQ [3] by 0.6% and 2.2% for 3-bit and 4-bit quantization, respectively.

When compared with the per-tensor and asymmetry quantization methods, in ResNet18, our method is 0.6% below MultiQuant [10] at 2-bit and exceeds it by more than 0.4% at all other bit-widths, especially at 3 bit-width where our method is 1.8% higher. In MobileNetV2, our approach has the same performance as MultiQuant at 3-bit, and is about 0.8% higher in other bit-widths. We surpass LSQ+ [1] by

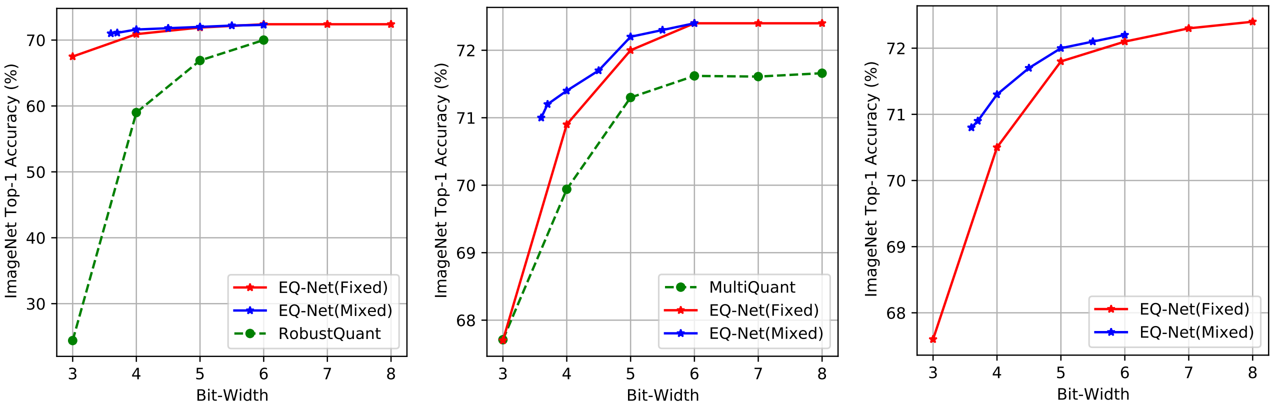


(a) Per-tensor, Symmetry.

(b) Per-tensor, Asymmetry.

(c) Per-channel, Symmetry.

Figure 1: Comparison with other methods for different quantization configuration in ResNet18.

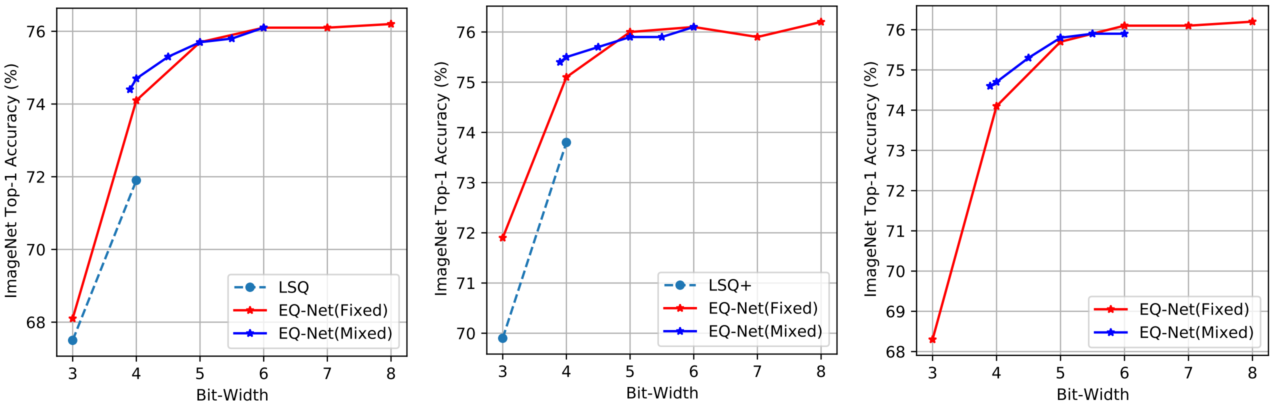


(a) Per-tensor, Symmetry.

(b) Per-tensor, Asymmetry.

(c) Per-channel, Symmetry.

Figure 2: Comparison with other methods for different quantization configuration in MobileNetV2.

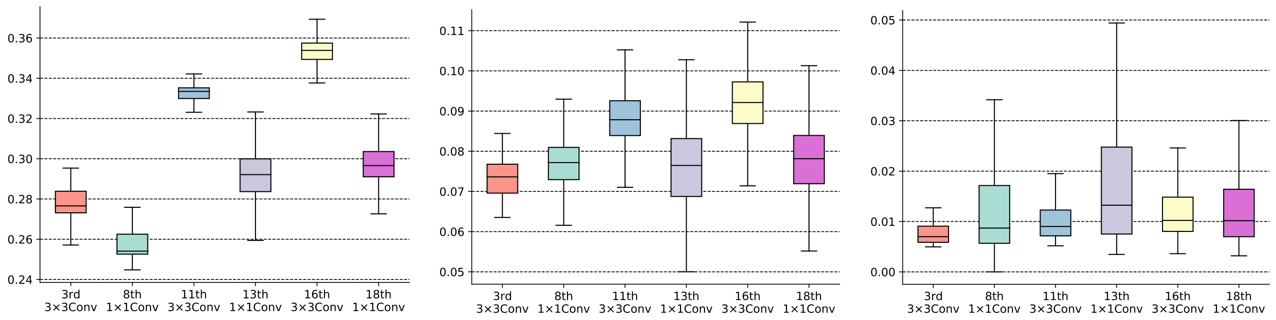


(a) Per-tensor, Symmetry.

(b) Per-tensor, Asymmetry.

(c) Per-channel, Symmetry.

Figure 3: Comparison with other methods for different quantization configuration in EfficientNetB0.

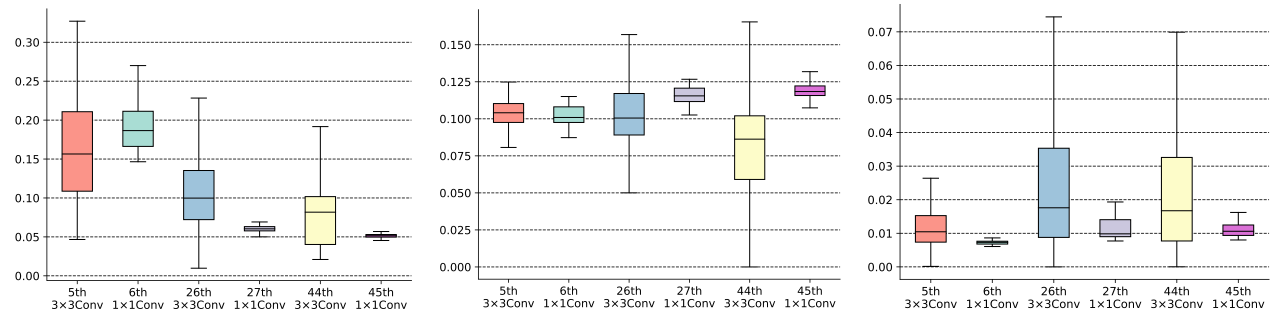


(a) 2bit-width.

(b) 4bit-width.

(c) 8bit-width.

Figure 4: Comparison of step size range for different bit-width in ResNet18.

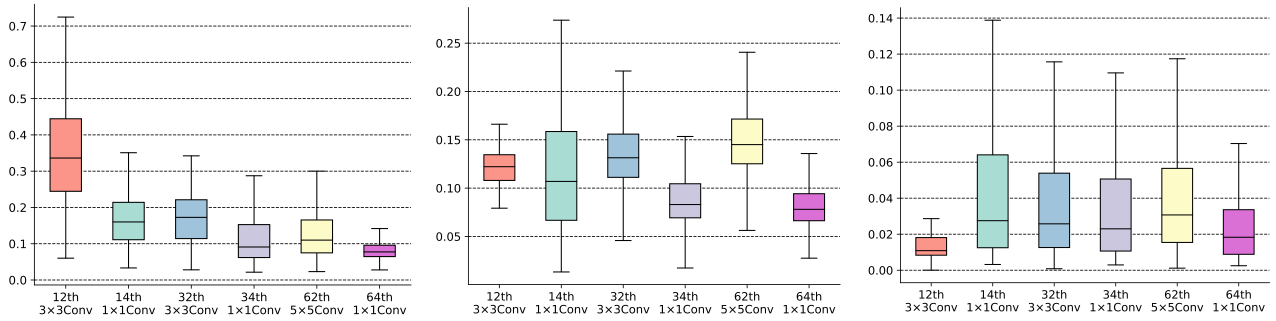


(a) 2bit-width.

(b) 4bit-width.

(c) 8bit-width.

Figure 5: Comparison of step size range for different bit-width in MobileNetV2.



(a) EfficientNetB0 2bit-width.

(b) EfficientNetB0 4bit-width.

(c) EfficientNetB0 8bit-width.

Figure 6: Comparison of step size range for different bit-width in EfficientNetB0.

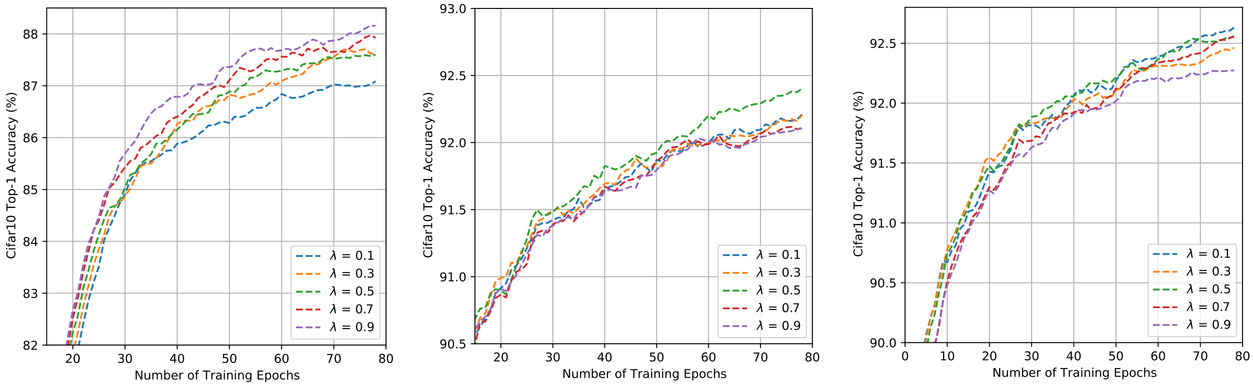
2.0%/1.3% in EfficientNetB0 at 3/4 bit-widths, respectively.

Since there are relatively few methods using per-channel and symmetry quantization methods, we only plot the curves of EQ-Net using this method under three networks.

E. Quantization Step Size Range

We analyzed the distribution range of step size in different networks when using per-channel quantization. Figure 4, 5, 6 shows the box plots of step size for various layers of three kinds of networks respectively. We separately selection the 1×1 and 3×3 convolutions from the front, middle and rear segments of EfficientNetB0 [8], MobileNetV2 [5] and ResNet18 [4] models, specifically, we also select the 5×5 convolution in EfficientNetB0.

In the same model, the step size gets smaller as the quantization bit-width increases, and accordingly its fluctuation range narrower. For example, in the 44th convolutional layer of MobileNetV2, when the quantization bit-width is 2, the



(a) 2 bit-width.

(b) 4bit-width.

(c) 8bit-width.

Figure 7: Sensitivity analysis of λ in ResNet20 on CIFAR10

step size fluctuation range is between 0.02 and 0.20, but the range becomes narrower to 0.00 to 0.07 as the quantization bit-width increases to 8.

At the same bit-width of different networks, ResNet18 exhibits the narrowest range of step size fluctuation, followed by MobileNetV2, while EfficientNetB0 has the largest range of variation. When the bit width is set to 2, the step size of different layers in ResNet18 is majority distributed between 0.24 and 0.38, in MobileNetV2 the distribution is widened to between 0.00 and 0.35, while in EfficientNetB0 the distribution is further stretched to between 0.00 and 0.72. Such phenomenon occurs due to the use of separable convolution in the latter two networks. Notably, as seen in Figure 5, the 1×1 convolution in MobileNetV2 has substantially narrower step size fluctuation range than its 3×3 convolution.

F. Sensitivity Analysis of λ

Figure 7 shows the convergence curves of EQ-Net with different values of λ in Eq.(7) at 2-4-8 bit-widths. λ in Eq.(7) represents the ratio of KL divergence between the subnetwork and the supernet, which is used to distill the knowledge from the supernet to the subnetwork. As shown in the figure, when $\lambda = 0.9$, the performance of EQ-Net is optimal at 2 bit-width, but the performance is worse at 8 bit-width. Conversely, when lambda is 0.1, the performance is optimal at 8 bit-width but worse at 2 bit-width. Therefore, in order to achieve a well-balanced performance at different bit-widths, our λ takes the value of 0.5.

References

[1] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. LSQ+: improving low-bit quantization through learnable offsets and better initialization. In *Proc. of CVPR*, 2020.

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

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

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

[5] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of CVPR*, 2018.

[6] Moran Shkolnik, Brian Chmiel, Ron Banner, Gil Shomron, Yury Nahshan, Alex M. Bronstein, and Uri C. Weiser. Robust quantization: One model to rule them all. In *Proc. of NeuIPS*, 2020.

[7] Ximeng Sun, Rameswar Panda, Chun-Fu Chen, Naigang Wang, Bowen Pan, Kailash Gopalakrishnan, Aude Oliva, Rogério Feris, and Kate Saenko. All at once network quantization via collaborative knowledge transfer. *ArXiv preprint*, 2021.

[8] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. of ICML*, 2019.

[9] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: hardware-aware automated quantization with mixed precision. In *Proc. of CVPR*, 2019.

[10] Ke Xu, Qiantai Feng, Xingyi Zhang, and Dong Wang. Multiquant: Training once for multi-bit quantization of neural networks. In *Proc. of IJCAI*, 2022.

[11] Haichao Yu, Haoxiang Li, Honghui Shi, Thomas S. Huang, and Gang Hua. Any-precision deep neural networks. In *Proc. of AAAI*, 2021.