MultiQuant: Training Once for Multi-Bit Quantization of Neural Networks

Ke Xu [1,2] , **Qiantai Feng** [3,4] , **Xingyi Zhang** [1,2] , **Dong Wang** [3,4] *

[1] School of Artificial Intelligence, Anhui University, Hefei, China

[2] Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Hefei, China

[3] Institute of Information Science,Beijing Jiaotong University, Beijing, China

[4] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China

xuke@ahu.edu.cn, qtfeng@bjtu.edu.cn, xyzhanghust@gmail.com, wangdong@bjtu.edu.cn

Abstract

Quantization has become a popular technique to compress deep neural networks (DNNs) and reduce computational costs, but most prior work focuses on training DNNs at each individual fixed bit-width and accuracy trade-off point. How to produce a model with flexible precision is largely unexplored. This work proposes a multi-bit quantization framework (MultiQuant) to make the learned DNNs robust for different precision configuration during inference by adopting Lowest-Random-Highest bit-width co-training method. Meanwhile, we propose an online adaptive label generation strategy to alleviate the problem of vicious competition under different precision caused by one-hot labels in the supernet training. The trained supernet model can be flexibly set to different bit widths to support dynamic speed and accuracy trade-off. Furthermore, we adopt the Monte Carlo sampling-based genetic algorithm search strategy with quantization-aware accuracy predictor as evaluation criterion to incorporate the mixed precision technology in our framework. Experiment results on ImageNet datasets demonstrate MultiQuant method can attain the quantization results under different bit-widths comparable with quantization-aware training without retraining.

1 Introduction

Deep neural networks have achieved remarkable performances in the fields of computer vision, machine translation, speech recognition, etc. However, the high computational demand and storage cost have posed great challenges for deployment of DNN-based algorithms, especially when embedded setting with limited hardware resource were considered. DNN quantization [Gholami *et al.*, 2021; Qin *et al.*, 2020] is a very effective approach to address this problem. By converting the floating-point-valued weights into low precision fixed-point values, quantization can shrink DNN model's memory footprint without changing the original network architecture. Moreover, the expensive floating-point matrix multiplications between weights and activations can be efficiently implemented on low-precision arithmetic circuit with much lower hardware costs and power consumption.

It is often necessary to adjust the parameters in the neural networks after quantization. This can either be performed by retraining the model, a process that is called Quantization-Aware Training (QAT) [Esser *et al.*, 2020; Gong *et al.*, 2019; Choi *et al.*, 2018; Zhou *et al.*, 2016], or done without retraining, a process that is often referred to as Post-Training Quantization (PTQ) [Li *et al.*, 2021; Nagel *et al.*, 2020; Nahshan *et al.*, 2019; Migacz, 2017]. PTQ usually requires a small subset of training data but produces less powerful quantized models than QAT, especially in low-precision quntization. However, QAT often involve simulating the quantization process during training, making the trained model highly dependent on the target bit-width. If the target bit-width is changed, the model needs to be retrained, which is time-consuming and resource-intensive. To address this issue, we introduce Lowest-Random-Highest bit-width Co-Training (LRH Co-Training) method, which provides a collaborative training strategy for multiple quantization models to enhance the robustness of weights in quantization scenarios. The model only needs to be trained once and then can flexibly and directly set its layers to different bit-widths and support mixed-precision quantization without retraining. It can achieve deployment efficiency faster than PTQ and the quantization accuracy is close to QAT method.

This paper makes the following contributions: (i) We propose a novel MultiQuant framework to train one-shot weight-sharing multi-bit supernet under the benchmark model to support subnets with uniform and mixed-precision quantization without retraining. (ii) We find the problem of vicious competition between high and low bit-widths in supernet training, and further design an online adaptive label to alleviate it. (iii) We propose Monte Carlo sampling instead of uniform sampling combined with genetic algorithm and quantization-aware accuracy predictor to improve the correlation and efficiency of mixed precision search.

2 Related Works

All-in-Once Network Architecture Search. It is well known that the use of proper parameter training techniques can make the model capable of multi-tasking at the same time. This practice has recently been widely applied to neural architecture search (NAS). OFA[Cai *et al.*, 2020], BigNAS [Yu *et al.*,

*Corrsponding Author

2020] and AttentiveNAS [Wang *et al.*, 2020a] push the envelope forward in network architecture search by introducing diverse architecture space (stage depth, channel width, kernel size and input resolution). These methods propose to train a single over-parameterized supernet from which we can directly sample or slice different candidate architectures for instant inference and deployment. Specifically, OFA propose progressive shrinking to reduce the interference between different sub-networks in weight-sharing supernet training, while BigNAS put forward sandwich rule to achieve one-stage model training. AttentiveNAS further propose attentive sampling of networks on Pareto-best and Pareto-worst to improve the performance. The above supernet training method, especially the sandwich rule, inspires us to train quantization supernet with the LRH Co-Training strategy. The work of APQ [Wang *et al.*, 2020b] used genetic algorithms by collecting quantized data points to realize the joint search of network architecture-pruning-quantization. However, APQ collect a quantized NN dataset for training the predictor is difficult (needs finetuning), and propose the predictor-transfer technique to make up for the lack of data which is sub-optimal and time-consuming. This work remedies the problem of difficult collection of quantized data points, and trains a quantization-aware predictor using sufficient quantized NN dataset to achieve mixed accuracy search with strong correlation.

All-in-Once Quantization of Neural Networks. Recent studie [Alizadeh *et al.*, 2020] models the quantization errors of weights and activation as additive l_∞ bounded perturbations and uses first-order approximation of loss function to derive a gradient norm penalty regularization that encourage the network's robustness to any bit-width quantization. RobustQuant [Shkolnik *et al.*, 2020] prove that compared to the typical case of normally-distributed weights, uniformly distributed weight tensors have improved tolerance to quantization with a higher signal-to-noise ratio and lower sensitivity to specific quantizer implementation and introduce Kurtosis regularization to uniformize the distribution of weights and improve their quantization robustness. CoQuant [Sun *et al.*, 2021] propose a novel collaborative knowledge transfer approach for training the all-in-once quantization network that can flexibly choose the bit-width during inference, without need of additional storage or re-training. AnyPrecision [Yu *et al.*, 2021] method training model with DoReFa [Zhou *et al.*, 2016] quantization constraints but save as floating-point form. Further, the floating-point model in runtime can be flexibly and directly set to different bit-widths, by truncating the least significant bits. OQAT [Shen *et al.*, 2021] present the bit inheritance mechanism under the OFA framework to reduce the bit-width progressively so that the higher bit-width models can guide the search and training of lower bit-width models, but limits its quantization policy search space to fixed-precision quantization policies. The above approaches are all implemented by implicitly constraining the weight distribution from the perspective of design loss function or knowledge transfer, and rarely discuss the interaction of different bit widths on weights-sharing quantization. This work identified the problem of vicious competition between high and low bit-widths in supernet training due to the hard label and further designed an online adaptive label to support subnets with arbitrary mixed-precision and uniform quantization policy.

3 Approach

In this section, we will explain the proposed training once for multi-bit quantization scheme in detail from three perspectives: multi-bit quantization modeling, training strategy, and Pareto frontier search based on mixed-precision.

3.1 Multi-Bit Quantization Modeling

We start with formalizing the problem of training the all-in-once network that supports versatile bit-width configurations. Assumed the quantization configuration of a model can be represented as $\mathcal{B} = \{(b_1^w, b_1^a), \ldots, (b_l^w, b_l^a), \ldots, (b_L^w, b_L^a)\}$, and b_l^w, b_l^a represent bit-width of weights and activation of the layer l respectively. Given floating-point weights $\mathbf{w}$ and activation $\mathbf{v}$, learnable quantization step size set $\mathbf{s} = \{s_{l,b}^w, s_{l,b}^a\}$ and zero-point set $\mathbf{z} = \{z_{l,b}^w, z_{l,b}^a\}$. Then the problem can be formalized as

$$\min_{\mathbf{w}^*, \mathbf{s}^*, \mathbf{z}^*} \sum_{\mathcal{B}} \mathcal{L}_{val}\left(Q\left(\mathbf{v}, \mathbf{w}, \mathbf{s}, \mathbf{z}, \mathcal{B}\right)\right) \tag{1}$$

where $Q(\cdot)$ denotes quantization function. Multi-bit quantization aims to learn the robust weights distribution, stand-alone quantization step size and zero-point set of activation and weight under different bit-width configurations. To enable efficient training of quantized models, a learnable quantization function is adopted from the recent low-bit quantization scheme LSQ [Esser *et al.*, 2020]. Taking activation quantization to b-bit as an example, the weights-sharing multi-bit quantization function is defined as follows:

$$Q(\mathbf{v}, \mathbf{s}, \mathbf{z}, b) = \left(\text{clip}\left(\left\lfloor \tfrac{\mathbf{v}}{\mathbf{s}} \right\rceil + \mathbf{z}, 0, 2^b - 1\right) - \mathbf{z}\right) \times \mathbf{s} \tag{2}$$

where all operations for $\mathbf{v}$ are element-wise operations, $\text{clip}(v, m, n)$ returns v with values below m set to m and values above n set to n, $\lfloor v \rceil$ rounds v to the nearest integer. The step size set $\mathbf{s}$ are learned by back-propagation and initialized by considering current layer-by-layer quantization methods where the quantization step size within each layer is optimized to accommodate the dynamic range of the tensor while keeping it small enough to minimize quantization noise [Nahshan *et al.*, 2019]. Zero-point set $\mathbf{z} = \left\lfloor -\frac{\mathbf{v}_{min}}{\mathbf{s}} \right\rceil$ are integer set, ensuring that zero is quantized with no error. This is important to ensure that common operations like zero padding do not cause quantization error. Note that the quantization form uses an asymmetric, per-tensor strategy.

3.2 Training the Multi-Bit Quantization Supernet

Lowest-Random-Highest Bit-Width Co-Training. Before training, given a mini-batch of data, we first initialize the quantization step size and zero-point sets of the model by the PTQ [Nahshan *et al.*, 2019] strategy. In each training step, we sample the lowest bit-width model, the highest bit-width model and randomly sampled bit-width models. It then aggregates the gradients from all sampled child models before updating the weights of the single-stage model. The lowest bit-width model is set to the lowest quantization bit-width for each layer, except for the first and last layers. In contrast, the highest bit-width model is set to the highest quantization bit width of each layer. The quantization bit widths of each layer in the highest and lowest bit-width models are 8-bit and 2-bit, respectively. The motivation is to improve the robustness of the model on all bit widths in search space simultaneously, by pushing up both the performance lower bound (the lowest bit-width model) and the performance upper bound (the highest bit-width model) across all child models. In random model, the quantization bit width of each layer can be randomly selected. The random quantization model is added mainly to improve the robustness of the supernet for mixed precision quantization. Note that to ensure the effectiveness of the quantization model, the quantization bit width of the first and last layers of the model are both set to 8-bit.

Online Adaptive Label. The difference between the proposed MultiQuant framework and OFA-like NAS is that the number of weights of the biggest and smallest child models of NAS are different, while the weights of MultiQuant are completely shared under different quantization configurations. Therefore, NAS can isolate precision conflicts under different configurations by training differentiated parameters, but MultiQuant can only coordinate the distribution of weights to adapt different bit-width configurations. In general, the variance of the confidence distribution of the quantization model prediction results is gradually decreasing from highest bit model to lowest bit model. By using the cross-entropy loss function measure with hard label, the loss under 2-bit will be much larger than the loss under 8-bit. After gradient accumulation, the model parameter learning is mainly to adapt to the 2-bit quantization distribution, which in turn impairs the 8-bit quantization accuracy. To alleviate these problems, we propose an online adaptive label method. We generates soft labels based on the statistics of the LRH quantization model prediction for the target category, which are subsequently used to supervise the supernet. Compared with hard labels and label smoothing [Szegedy *et al.*, 2016], this strategy can dynamically and adaptively adjust label distribution during supernet training. Given a dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x_i}, y_i)\}$ with N classes, where $\boldsymbol{x_i}$ denotes the input image and y_i denotes the corresponding ground-truth label. Formally, let E denote the number of training epochs. We define $\mathcal{A} = \left\{A^0, A^1, \cdots, A^e, \cdots, A^{E-1}\right\}$ as the collection of the class-level soft labels at different training epochs. Here, A^e is a matrix with N rows and N columns, and each column in A^e corresponds to the soft label for one category. At the beginning of the e_{th} training epoch, we initialize the soft label A^e as a zero matrix. When an input sample $(\boldsymbol{x_i}, y_i)$ is correctly classified by any quantization model, we utilize its predicted scores $\{p_L(\boldsymbol{x_i}), p_R(\boldsymbol{x_i}), p_H(\boldsymbol{x_i})\}$ to update the y_i column in A^e, which can be formulated as:

$$A^e_{y_i,n} = A^e_{y_i,n} + \frac{p_L(n \mid \boldsymbol{x_i}) + p_R^m(n \mid \boldsymbol{x_i}) + p_H(n \mid \boldsymbol{x_i})}{3} \tag{3}$$

At the end of the e_{th} training epoch, we normalize the cumulative A^e column by column as represented by $A^e_{y_i,n} \leftarrow \frac{A^e_{y_i,n}}{\sum_{n=1}^{N} A^e_{y_i,n}}$. We can now obtain the normalized soft label A^e for all N categories, which will be used to supervise the model at the next training epoch. Since the quantization at the beginning can generates large noise and lacks accurate labels. Thus, we utilize both the hard label and soft label as supervision to train the model. The loss based online adaptive label can be represented by

$$L_{OAL} = -\sum_{n=1}^{N}\left((1-\zeta)q(n \mid \boldsymbol{x_i}) + \zeta A^{e-1}_{y_i,n}\right)\log p(n \mid \boldsymbol{x_i}) \tag{4}$$

where ζ is used to balancing hard label and soft label that is usually set to 0.5 in practice. The proposed L_{OAL} constructs a more robust distribution between the lowest-bit model and highest-bit model to co-training supernet for multi-bit quantization task scenario, as shown in Figure 2.

3.3 Search Pareto Frontier for Mixed-Precision

Monte Carlo Sampling. In the mixed precision search phase, we need search out mixed precision subnets from the weight-sharing model, according to the given average bit-width constraint. At the beginning of the search, we need to sample a sufficient number of models that meet the mixed bit constraint to construct the initial population. However, a large number of models are concentrated in the middle bit-width, e.g. , all-in-once quantization model from 2 to 8 bit-width contained a plenty of subnet around 5 average bit-width. When we need to search low or high bit-width models, it takes a lot of time if we try the traditional uniform sampling. Therefore, we adopt the Monte Carlo sampling strategy to estimate distributions of bits width configuration under different average bit-width constraints for sampling to improve sampling efficiency. Given the average bit-width constraint of weights and activations τ_w and τ_a , the empirical approximation of $\pi(\mathcal{B}|\tau_w, \tau_a)$ is $\hat{\pi}(\mathcal{B}|\tau_w, \tau_a)$. In order to facilitate statistics, $\hat{\pi}(\mathcal{B}|\tau_w, \tau_a)$ can be relaxed as shown below:

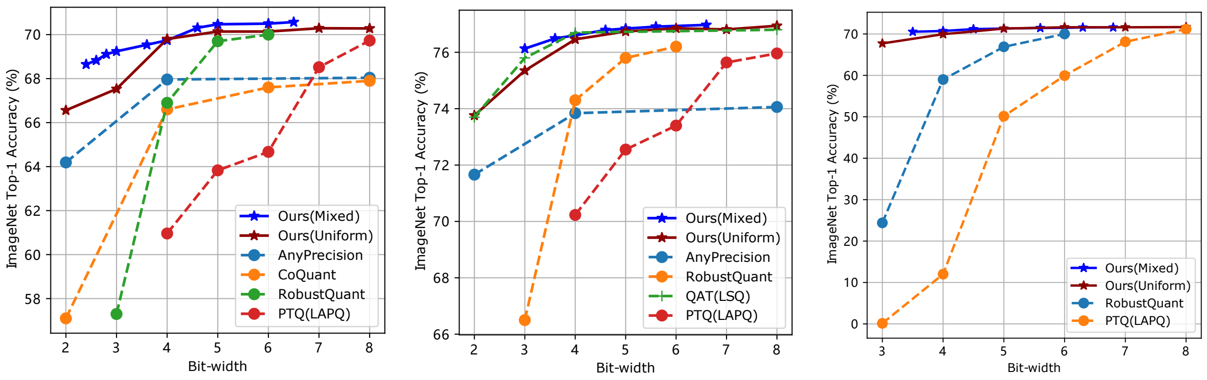
$$\hat{\pi}(\mathcal{B}|\tau_w, \tau_a) \propto \prod_{l}^{L} \hat{\pi}(b_l^w|\tau_w)\hat{\pi}(b_l^a|\tau_a) \tag{5}$$

To get the distribution above, we randomly get a large number of mixed-presicion structure and average bit-width data pairs$\{\mathcal{B}, (\tau_w, \tau_a)\}$ from the search space to construct a model average bit-width sampling pool. Let $\#(\tau_w = \tau_0)$ denote the total number of subnet with τ_0 in the sampling pool, and $\#(b_l^w = k, \tau_w = \tau_0)$ denote the number of time that the pair of $(b_l^w = k, \tau_w = \tau_0)$ appears in the sampling pool, then the $\hat{\pi}(b_l^w|\tau_w)$ can be estimated as:

$$\hat{\pi}(b_l^w|\tau_w) = \frac{\#(b_l^w = k, \tau_w = \tau_0)}{\#(\tau_w = \tau_0)} \tag{6}$$

$\hat{\pi}(b_l^a|\tau_a)$ can be acquired in the same way. Sampling the subnet in the above distribution can greatly improve the possibility of satisfying the average bit-width constraints to improve the efficiency of the search.

Quantization-Aware Accuracy Predictor. In the search process, it is very important to accelerate the evaluation procedure of the searched model. We use accuracy predictor for efficient performance estimation, which can predict the accuracy of a model given its configuration. More specifically, it is a 7-layer feed-forward neural network with each embedding



(a) ResNet18. (b) ResNet50. (c) MobileNetV2.

Figure 1: Comparison with All-in-Once (AnyPrecision [Yu *et al.*, 2021],CoQuant [Sun *et al.*, 2021] and RobustQuant [Shkolnik *et al.*, 2020]), QAT(LSQ [Esser *et al.*, 2020]) and PTQ(LAPQ [Nahshan *et al.*, 2019]) methods for different bit-width and different model.

dim equaling to 150. The bit-width configuration is encoded into an one-hot vector as the input, which is fed into the predictor to get the predicted accuracy as output. Different from the previous work, e.g. OFA, we use Monte Carlo Sampling to generate the structure-accuracy pair dataset, which can avoid the imbalance of the data set and improve the prediction performance of lower and higher bit-width. In addition, with the all-in-once quantization supernet we can quickly obtain quantization data pairs without predictor-transfer technique like APQ, allowing for more efficient and accurate prediction of the quantization model.

Genetic Algorithm for Mixed-Precision Search. The mixed-precision search is designed to explore the best candidate bit-width configuration for each layer of supernet. In the search phase, we use the bit-width of each layer explored by the genetic algorithm[Whitley, 1994] as input to accuracy predictor and obtain the accuracy, which is then used as the evaluation standard to sort and select the best candidates for the exploration results. The genetic algorithm first randomly generates several chromosomes as the initial Pareto solution set. Secondly, the accuracy of all candidate quantization networks produced by predictor are evaluated as the fitness scores. Finally, the chromosomes with the highest fitness scores are preserved and added to the elitists, which are then selected for mutation and crossover to obtain a new population according to a predefined probability. The selection-mutation-crossover procedure is repeatedly conducted until the algorithm reaches a satisfactory Pareto solution that meets the weights and activations average bit-width targets.

4 Experimental Results

In this section, we conduct extensive experiments to show that our approach outperforms many strong baselines while achieving comparable performance with individual quantization models on ImageNet datasets. We also perform comprehensive ablation experiments and visualization analysis to verify the effectiveness of MultiQuant for all-in-once quantization.

4.1 Implementation Details

We present results of using the pre-trained benchmark models by TorchVision, including ResNet18, ResNet50 [He *et al.*, 2016] and MobileNetV2 [Sandler *et al.*, 2018] on the ImageNet [Deng *et al.*, 2009] dataset. We train the models for 90 epochs by using Adam [Kingma and Ba, 2015] with a cosine learning rate decay. The batch size is set as 256, base learning rate is set as 0.001 and weight decay of 0. We train the ResNet18 and ResNet50 with bit-width candidates $\{2, 3, 4, 5, 6, 7, 8\}$. Since MobileNetV2 is a compact model, 2-bit quantization lead to worse performance, so the bit-width candidates are set to $\{3, 4, 5, 6, 7, 8\}$. We sample 5K mixed precision configurations and test their accuracy as the accuracy dataset<BitSet,Acc> for training the accuracy predictor. In the genetic algorithm for mixed-precision search, the size of population in each generation is 100 (50 each for mutation and crossover), and the number of exploring iteration is set to 500.

4.2 Comparison with State-of-the-Art Methods

Figure 1 presents the comparison with fixed quantization, mixed precision and all-in-once quantization methods in different bit-width and different model. It can be observed that the proposed scheme consistently outperforms the quantization results of state-of-the-art all-in-once works with either fixed or mixed-precision. Specifically, when the bit-width is set to 2, the proposed scheme show 2.3%/2.1% accuracy boosts over AnyPrecision (from 64.19%/66.56% to 71.66%/73.76%) for the ResNet18/ResNet50 models. Compared to RobustQuant and CoQuant, our method has almost 10% accuracy improvement, mainly due to the LRH Co-Training strategy that improves the adaptation of model weights to low bit-width scenarios. It is worth noting that MultiQuant model with weights and activations quantization of 3 MP can achieve similar accuracy to the baseline on ResNet50. In addition, by comparing with different quantization methods, AnyPrecision performs well compared to RobustQuant and CoQuant methods in low bit-width scenarios but also makes lossless quantization of the model impossible for high bit-widths (e.g.,6-bit to 8-bit). On the contrary, RobustQuant ensures no loss of accuracy in high bit-width but cause a significant loss of accuracy in 2-bit. The

Network	Benchmark	Uniform Quantization	Mixed Quantization	Search cost (GPU hours)	Training cost (GPU hours)	Total (GPU hours)
ResNet18	LSQ	✓	×	——	120N	120N
	EdMIPS	×	✓	20N	120N	140N
	AnyPrecision	✓	×	——	152	152
	CoQuant	✓	×	——	——	——
	RobustQuant	✓	×	——	427	427
	MultiQuant	✓	✓	20	268	288
ResNet50	LSQ	✓	×	——	480N	480N
	HAQ	×	✓	190N	384N	574N
	MultiQuant	✓	✓	96	1200	1296
MobileNetV2	LSQ	✓	×	——	240N	240N
	HAQ	×	✓	96N	192N	288N
	MultiQuant	✓	✓	48	620	668

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

Method	QAT	PTQ	MultiQuant		
			L_{HL}	L_{LS}	L_{OAL}
2 bit	89.83%	21.30%	88.61%	88.20%	88.77%
3 bit	91.36%	83.91%	91.07%	90.57%	91.07%
4 bit	91.87%	90.48%	91.69%	91.81%	92.08%
5 bit	91.81%	91.64%	91.87%	92.21%	92.17%
6 bit	92.19%	92.05%	91.89%	92.18%	92.21%
7 bit	92.16%	92.33%	91.96%	92.17%	92.23%
8 bit	92.24%	92.19%	91.84%	92.23%	92.31%

Table 2: Comparison of MultiQuant results with QAT and PTQ under different loss function and bit-width of ResNet-20 on CIFAR-10.

online adaptive soft label strategy proposed in this work ensures that the quantization results of the supernet can maintain the desired accuracy at different bit widths, and even the quantization results of different bit widths on ResNet50 are close to LSQ. Moreover the Pareto front curve (blue solid line in Figure 1) based on mixed precision quantization outperforms the uniform quantization front (red solid line in Figure 1) for different models, especially in the low-bit condition.

4.3 Ablation Studies

The Efficiency of MultiQuant Framework. DNNs take up tremendous amounts of energy, leaving a large carbon footprint. Quantization can improve energy efficiency of neural networks on both commodity GPUs and specialized accelerators. MultiQuant takes another step and create one model that can be deployed across many different inference chips avoiding the need to re-train it before deployment (i.e., reducing CO_2 emissions associated with re-training). As shown in Table 1, MultiQuant is much more efficient than uniform and mixed precision quantization method when handling multiple deployment scenarios, since the cost of MultiQuant is constant while LSQ/HAQ/EdMIPS are linear to the number of deployment scenarios (N). Compared to the all-in-once method, MultiQuant adapts to more quantization scenarios.

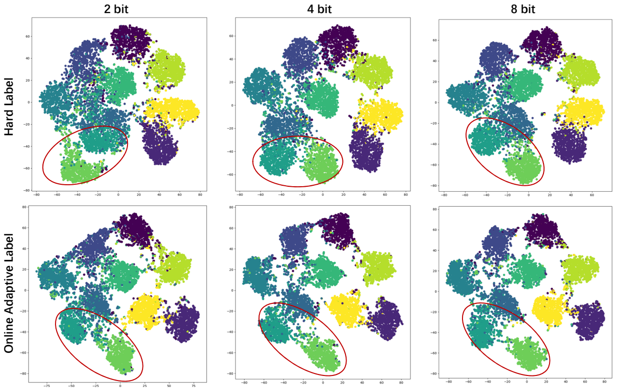
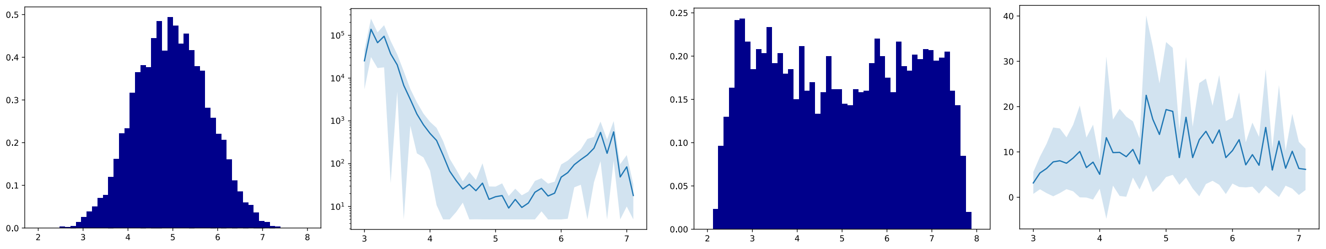


Figure 2: Visualization of the penultimate layer representations of ResNet-20 on CIFAR-10 under different bit widths.

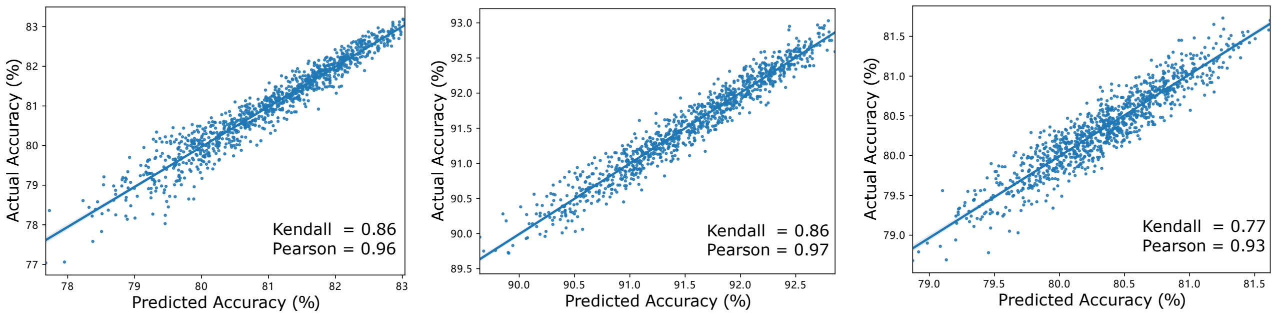
The Effectiveness of Online Adaptive Label. According to Table 2, we first focus on the quantization results in the two columns of PTQ [Nahshan *et al.*, 2019] and L_{HL}. L_{HL} represents the one-hot cross-entropy loss. The quantization result of 2 to 4 bits can be significantly improved based on the LRH Co-Training, but 6 to 8 bits is lower than the result of PTQ. It can be proved that the co-training process will affect the accuracy of high-bit quantization due to over-optimization of low-bit. Secondly, by comparing with three columns of L_{HL}, L_{LS} [Szegedy *et al.*, 2016], and L_{OAL} in the table, we can find that online adaptive label can effectively alleviate the problem of high-bit quantization accuracy degradation. Compared with the QAT [Esser *et al.*, 2020] result, only the 2-bit quantization accuracy drop 1%, and the other accuracy is close to QAT, or even better. To give a more intuitive explanation, we utilize t-SNE [Maaten and Hinton, 2008] to visualize the penultimate layer representations of ResNet-20 on CIFAR-10 trained with L_{HL} and L_{OAL}, respectively. Figure 2 shows that online adaptive label provides a more recognizable difference between representations of different



(a) Uniform Sampling.

(b) Monte Carlo Sampling.

Figure 3: An illustration of the the effectiveness of Monte Carlo Sampling. Figure 3(a) shows the distribution of the average bit-width in the accuracy dataset with 5K samples, and the number of trails to sample one model under constraints by uniform sampling(each constraint is tested 20 times) in ResNet18. Figure 3(b) shows the corresponding result by Monte Carlo Sampling.



(a) ResNet18.

(b) ResNet50.

(c) MobileNetV2.

Figure 4: Rank correlation between actual accuracy and predicted accuracy on split validation set of ImageNet.

classes and tighter intra-class representations. Its category distribution under different quantization bit widths is more stable than the hard label, as shown by the red ellipses.

The Effectiveness of Monte Carlo Sampling. The Monte Carlo Sampling can help to generate a relatively uniform accuracy sampling pool, as well as shorten the sampling time in the search process when searching the low or high average bit-width model. We visualize the distribution of the average bit-width in the accuracy dataset of ResNet18 and the number of trails to sample one model under constraints in Figure 3. It is obvious from Figure 3(a) that most of the models in the sample space are around 5 average bit-width, and the 3 or 7 average bit-width models are very few, and the need to sample a lot of times to get one sample under target constraint in the search procedure, e.g., average 25370 times under 3 average bit-width. As shown in 3(b), by Monte Carlo Sampling, we not only generate more balanced accuracy dataset, which can improve the performance of the accuracy predictor in low and high bit-width condition, but also reduce the number of trails tremendously, e.g., average 5 times under 3 average bit-width.

Rank Preservation Analysis of Accuracy Predictor. In our search process, we adopt accuracy predictor to estimate the candidate performance, and it is important to maintain the rank correlation between the prediction of predictor and the actual performance. We estimate the candidate performance by the accuracy on 10K images sampled from the original training set. The illustration correlation graph and coefficient of rank correlation of three all-in-once models are shown in Figure 4. It can be seen that the Kendall coefficient τ can reach 0.86 on ResNet18 and ResNet50. The parameters of MobileNetV2 have a long tail distribution, which is sensitive on accuracy to the bit width setting. The Kendall coefficient is 0.77 lower than the ResNet model. The Pearson coefficient is above 0.9 for all three models, which confirms strong correlation between predicted accuracy and actual accuracy.

5 Conclusion

This work aims at maintaining a good performance under a variety of quantization bit-widths scenarios. We analyzed how quantization at different precisions influences the compute cost-quality Pareto curves on different models including ResNet18, ResNet50 and MobileNetV2. We found the problem of vicious competition between high bit-width and low bit-width in the Lowest-Random-Highest bit-width co-training phase, and designed an online adaptive label to alleviate this problem. Additionally, this work also supports mixed precision search based on genetic algorithms. By analyzing the quantization results, we demonstrate that all-in-once quantization training model achieved accuracy comparable to QAT. Noted that this work implements a mixture of training in the 2 to 8 bit precisions space of the ResNet structure, while the MobileNetV2 model only implements training in the 3 to 8 bit due to the compact structure design. Therefore, how to solve the low bit-width training with compact structure (e.g. MobileNetV2/V3) could be a future research point.

Acknowledgments

This work was supported by National Key R&D Program of China under Grant 2018AAA0100100, Beijing Natural Science Foundation under Grant 4202063, National Key Research and Development Program of China under Grant 2019YFB2204200.

References

[Alizadeh *et al.*, 2020] Milad Alizadeh, Arash Behboodi, Mart van Baalen, Christos Louizos, Tijmen Blankevoort, and Max Welling. Gradient ℓ_1 regularization for quantization robustness. In *Proc. of ICLR*, 2020.

[Cai *et al.*, 2020] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *Proc. of ICLR*, 2020.

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

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*, 2009.

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

[Gholami *et al.*, 2021] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *ArXiv preprint*, 2021.

[Gong *et al.*, 2019] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proc. of ICCV*, 2019.

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

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

[Li *et al.*, 2021] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: pushing the limit of post-training quantization by block reconstruction. In *Proc. of ICLR*, 2021.

[Maaten and Hinton, 2008] L. V. D. Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.

[Migacz, 2017] Szymon Migacz. 8-bit inference with tensorrt. In *GPU technology conference*, 2017.

[Nagel *et al.*, 2020] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proc. of ICML*, 2020.

[Nahshan *et al.*, 2019] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alexander M. Bronstein, and Avi Mendelson. Loss aware post-training quantization. *ArXiv preprint*, 2019.

[Qin *et al.*, 2020] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 2020.

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

[Shen *et al.*, 2021] Mingzhu Shen, Feng Liang, Ruihao Gong, Yuhang Li, Chuming Li, Chen Lin, Fengwei Yu, Junjie Yan, and Wanli Ouyang. Once quantization-aware training: High performance extremely low-bit architecture search. In *Proc. of ICCV)*, 2021.

[Shkolnik *et al.*, 2020] 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.

[Sun *et al.*, 2021] 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.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. of CVPR*, 2016.

[Wang *et al.*, 2020a] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentivenas: Improving neural architecture search via attentive sampling. *ArXiv preprint*, 2020.

[Wang *et al.*, 2020b] Tianzhe Wang, Kuan Wang, Han Cai, Ji Lin, Zhijian Liu, Hanrui Wang, Yujun Lin, and Song Han. APQ: joint search for network architecture, pruning and quantization policy. In *Proc. of CVPR*, 2020.

[Whitley, 1994] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 1994.

[Yu *et al.*, 2020] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas S. Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *Proc. of ECCV*, 2020.

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

[Zhou *et al.*, 2016] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *ArXiv preprint*, 2016.