Improved Techniques for Quantizing Deep Networks with Adaptive Bit-Widths

Ximeng Sun[1†], Rameswar Panda[2,3], Chun-Fu (Richard) Chen[2,3†], Naigang Wang[3], Bowen Pan[4], Aude Oliva[2,4], Rogerio Feris[2,3], Kate Saenko[1,2]
[1]Boston University, [2]MIT-IBM Watson AI Lab, [3]IBM Research, [4]MIT

Abstract

Quantizing deep networks with adaptive bit-widths is a promising technique for efficient inference across many devices and resource constraints. In contrast to static methods that repeat the quantization process and train different models for different constraints, adaptive quantization enables us to flexibly adjust the bit-widths of a single deep network during inference for instant adaptation in different scenarios. While existing research shows encouraging results on common image classification benchmarks, this paper investigates how to train such adaptive networks more effectively. Specifically, we present two novel techniques for quantizing deep neural networks with adaptive bit-widths of weights and activations. First, we propose a collaborative strategy to choose a high-precision "teacher" for transferring knowledge to the low-precision "student" while jointly optimizing the model with all bit-widths. Second, to effectively transfer knowledge, we develop a dynamic block swapping method by randomly replacing the blocks in the lower-precision student network with the corresponding blocks in the higher-precision teacher network. Extensive experiments on multiple image and video classification datasets, well demonstrate the efficacy of our approach over state-of-the-art methods.

1. Introduction

Low-precision deep neural networks [21, 57, 55, 11, 60, 50], which severely reduce computation and storage by quantizing weights and activations to low-bit representations, have attracted intense attention in recent years. Despite much progress in quantization for improving efficiency of deep networks, most of the existing methods repeat the quantization process and retrain the low-precision network from scratch, leading to different models for different resource constraints [57, 11, 60] (see Figure 1(a) for an illustrative example). This strategy leads to optimal efficiency for a given resource constraint. But designing specialized low precision models for every practical scenario is often not flexible and economical in terms of both memory cost, and energy consumption, either with human-based methods or automatic search-based methods.

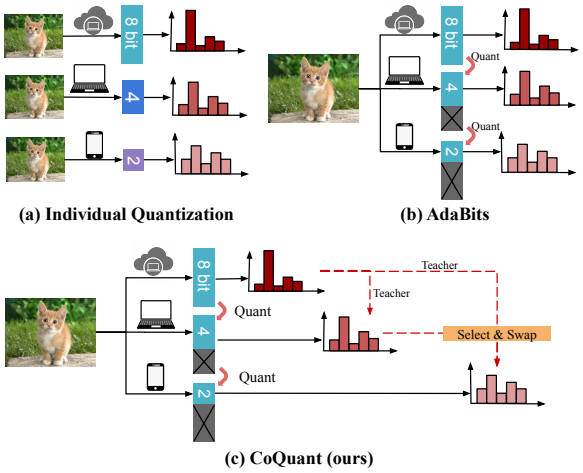

Figure 1: **A conceptual overview of our approach**. Consider a deployment scenario that requires three different bit-widths (e.g., 8, 4, and 2 bits) according to the resource constraints. (a) Conventional methods train individual quantized models with specific bit-widths for each case, which prohibits instant adaptation to different scenarios. (b) AdaBits [22] jointly learns a network by treating each bit-width equally, which fails to achieve optimal performance due to increasing optimization difficulty for lower precisions. (c) We propose a novel collaborative quantization (**`CoQuant`**) algorithm by *dynamically selecting and swapping blocks across different precisions* to transfer knowledge for effectively training a single model with all the bit-widths. Best viewed in color.

Motivated by this, recent works [22, 51, 5] focus on adaptive quantization which can flexibly choose the bit-width of a deep neural network during inference, to meet the dynamically changing demand. In particular, after training, we can freely quantize the weights and activations into various precision levels, without additional fine-tuning or calibration. A simple example of such network quantization is illustrated in Figure 1(b) where the network is jointly trained under different bit-widths with shared weights. While such an approach (*e.g.* AdaBits [22]) looks intuitive and handy at first glance, it fails to balance the optimization difficulty across different precisions (a major challenge in quantizing deep networks with adaptive bit-widths). As a result, the higher precision

[†]Work done at IBM.

tends to dominate the training, leading to sub-optimal performance across different quantization levels.

To address the above challenge, we present novel techniques for effective quantization of deep networks with adaptive bit-widths. It is well known that supervising the training of a small neural network by capturing and transferring the knowledge of a larger-parameter network can lead to a significant performance boost which is sometimes even better than that of the larger network [19, 10, 14, 33, 36]. However, unlike conventional knowledge transfer from a single teacher to a student network [19, 10, 33], a central problem while training adaptive networks for quantization is *from which high-precision teacher to transfer knowledge given multiple teachers of different capacities*. This is an especially important problem because a very low capacity student is often unable to mimic a very strong teacher, implying the necessity of intermediate teachers to bridge the gap among them [30]. Moreover, once the best high-precision teacher is identified, how to effectively transfer knowledge to the low-precision student is also crucial for training a single quantized neural network adaptive to different resource constraints.

To this end, we propose a collaborative quantization algorithm (**CoQuant**) for transferring knowledge from higher precision to lower precision while jointly optimizing the model with all the bit-widths (see Figure 1(c)). Specifically, we first develop a simple yet effective teacher selection strategy to choose the best teacher adaptively to the current input by balancing the confidence of prediction and the distance in the model space. We then adopt a dynamic block swapping method to transfer knowledge by randomly replacing the blocks in the lower precision network with the corresponding blocks in the higher precision network. Our dynamic block swapping not only utilizes strong feature transformation ability of the high-precision teacher at different locations of the network, but also makes the gradient back-propagate more easily for the low-precision student. For each input batch throughout the training, we dynamically select the best teacher for lower precisions, and apply our block swapping mechanism to transfer knowledge from the selected teacher.

Experiments on two standard image classification datasets (CIFAR10 [26] and ImageNet [37]) show that our proposed techniques greatly improve the training efficacy of deep networks with adaptive bit-widths and outperform the state-of-the-art methods, especially at the lower precisions, which is of significant practical value (e.g., about 1.7% and 3.0% improvement in 2-bit performance over AdaBits [22] on CIFAR10 and ImageNet, respectively). In addition to image classification, we extend our approach to quantizing video classification networks and observe that building low-precision networks for videos is more challenging with a larger performance drop at the lower-precisions compared to image classification. Through experiments on two video classification benchmark datasets (ActivityNet [13] and Mini-Kinetics [9]), we show that our approach achieves a significant improvement over the recent methods [22, 51, 5].

2. Related Work

Network Quantization. Several methods for quantizing deep networks have been studied, including binary [21, 34] or ternary networks [27], uniform quantization [55, 11, 57, 7, 32], or mixed precision quantization [44, 8, 46, 50, 51, 42]. Designing efficient strategies for training low bit-width models using distillation [60, 24] or auxiliary module [59] is also another popular trend in quantization. Most relevant to our approach are AdaBits [22] and Any Precision Networks [51] that use joint learning with separate quantization parameters for training a single network with adaptive bit-widths. Despite separate parameters, both approaches suffer from the performance degradation at low precision due to its interference with high precision networks which potentially disturbs the whole optimization of the network. Our approach on the other hand utilizes a collaborative mechanism for transferring knowledge across different precision networks, leading to higher performance at lower precision. Another distinctive feature of our approach is in swapping low precision blocks with the corresponding high precision blocks which helps in easily propagating gradient through the low-precision network. Bit-Mixer [5] trains mixed-precision networks that allow execution at variable bit-width quantization for every layer in the network. Our experiments quantitatively compare these three approaches [22, 51, 5], showing that **CoQuant** performs significantly better in adaptive quantization across multiple image and video classification datasets. A fully nested neural network that runs only once to build a nested set of compressed/quantized models is proposed in [12]. Our work is orthogonal to [12] as we propose two improved knowledge transfer techniques for effective training of adaptive quantization networks, which could also be adopted for training nested subnetworks.

Collaborative Learning. Collaborative learning for improving generalization ability of a network has been studied from multiple perspectives. Representative methods use knowledge distillation [19, 10, 33], force student networks to maintain their diversity via co-distillation [1], or jointly train multiple network branches to establish a strong teacher [58, 40]. Recent works in [14] and [48] focus on progressive module replacing for knowledge distillation and language model compression respectively unlike the problem considered in this paper. While our approach is inspired by these methods, in this paper, we focus on collaborative learning for network quantization, where our goal is to dynamically adjust the precision of a *single deep neural network* without requiring additional re-training. Note that we do not compress the network model by transferring knowledge between different models sizes [25, 14, 48] or employ a separate intermediate-sized network to bridge the gap between student and teacher [30].

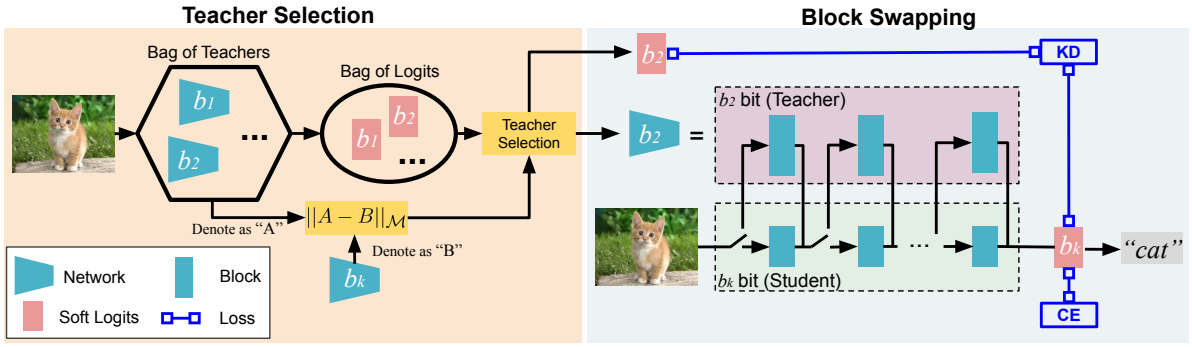

Figure 2: **Illustration of our proposed approach.** Our approach (**CoQuant**) trains deep networks jointly with all the given precisions using a single set of weights, and consists of two components. First, for bit-width b_k, we propose a dynamic teacher selection mechanism, based on the prediction confidence and the model distance, to select the best teacher for the current input from the existing higher precisions (with bit-width $b_1, b_2, \cdots b_{k-1}$). Second, to effectively transfer the teacher's knowledge, we introduce a dynamic block swapping strategy by randomly replacing the blocks in the lower precision network with the corresponding blocks in the higher precision network. We use task-specific supervision (*e.g.* Cross-Entropy (CE)) and a distillation loss across the selected teacher and student in training.

InstantNet [15] adopts cascade knowledge distillation for network quantization, but it always distills knowledge from the closest higher bit-width while we dynamically select the best teacher adaptive to current input and training progress. We show the superiority of our teacher selection mechanism over cascade distillation in the ablation study.

Adaptive Neural Networks. Many variants of adaptive neural networks have been recently proposed with the goal of achieving instant adaptation for different applications [2, 47, 16, 23]. While MSDNet [20] proposes a coarse to fine framework for makes early predictions to meet varying resource demands, MutualNet [49] trains a single network that is executable at dynamic resource constraints to achieve accuracy-efficiency trade-offs at runtime. Slimmable networks [53] train a model to support multiple width multipliers. OFA networks adjust width, depth and kernel sizes simultaneously for efficient deployment [6]. Despite recent progress, adaptive networks that are flexible to any numerical precision during inference still remains as a challenging and largely under-addressed problem.

3. Proposed Method

Given a set of n different bit-widths $\mathcal{B} = \{b_1, b_2, \cdots, b_n\}$ (assuming $b_1 > b_2 > \cdots > b_n$), the goal of adaptive network quantization is to seek a single set of weights, which can be executed with any bit-width $b \in \mathcal{B}$ during inference based on our demand (concerning computational resources, energy, storage, etc.), and achieves the best overall performance on $\mathcal{B}$. We denote a full precision model as $\mathcal{M}$ and its b-bit quantization as $\mathcal{M}_b$.

3.1. Preliminaries

In 2D Convolution, we denote the network weights and activations by two 4D tensors $\mathbf{W}$ and $\mathbf{A}$. Given a certain precision with bit-width b and a quantization function Q, we denote the quantization of $\mathbf{W}$ and $\mathbf{A}$ as $Q(\mathbf{W}, b) = \widehat{W}_b$ and $Q(\mathbf{A}, b) = \widehat{A}_b$. In this paper, we adopt DoReFa [57] for weight quantization and PACT [11], a learnable uniform quantization scheme, for activation quantization. Note that our approach is agnostic to the type of quantization scheme and hence can work on all categories of methods.

Weight Quantization. We first normalize $\mathbf{W}$ into $[-1, 1]$ and then round it to the nearest quantization levels, as in [57]:

$$\widehat{W}_b = 2 \times \text{quantize}_b\left(\frac{tanh(\mathbf{W})}{2 \max tanh(\mathbf{W})} + \frac{1}{2}\right) - 1, \quad (1)$$

$$\text{quantize}_b(x) = \frac{1}{2^b - 1} \times \lfloor (2^b - 1)x \rceil, \quad (2)$$

where $\lfloor . \rceil$ is the rounding operation.

Modified Weight Quantization. In DoReFa [57], all numerical precisions need to be quantized down from its 32-bit full-precision value. Even though highest precision is b_1 bit during inference, DoReFa requires the device to store the full-precision model $\mathcal{M}$ rather than $\widehat{M}_{b_1}$, which is memory inefficient. To alleviate this, we modify Eq. 2 to directly get $\widehat{W}_b$ from $\widehat{W}_{b_1}$ by discarding last $b_1 - b$ bits in $\widehat{W}_{b_1}$ instead of re-quantizing everytime from W as in DoReFa. We further align $\mathbb{E}[\widehat{W}_b]$ with $\mathbb{E}[\widehat{W}_{b_1}]$ to minimize mean discrepancy caused by the discarded bits. Our modified quantizer achieves very similar performance (e.g., 0.2% to 0.4% above or below on ActivityNet) compared to original DoReFa despite the fact that lower precisions are computed by simply dropping least significant bits.

Activation Quantization. For activation $\mathbf{A}$, we introduce a learnable clipping value α for each layer, whose value is originally in $[0, +\infty)$ [11]. The activation $\mathbf{A}$ is first clipped to α and then rounded to the nearest quantization levels:

$$\widehat{A}_b = \alpha \times \text{quantize}_b(\text{clip}(A, 0, \alpha)/\alpha).$$

Backward Propagation. The quantization function is non-differentiable and hence it is impossible to directly apply back-propagation to train the network. To circumvent this issue, we adopt the "Straight-Through Estimator" [18, 3, 57] to approximate gradient in the backward propagation:

$$\frac{\partial \mathcal{L}}{\partial r_i} = \frac{\partial \mathcal{L}}{\partial r_o}\frac{\partial r_o}{\partial r_i} = \frac{\partial \mathcal{L}}{\partial r_o},$$

where $\mathcal{L}$ is the loss function, r_i is an arbitrary input and r_o is the corresponding output of $\text{quantize}_b(r_i, b)$.

3.2. Approach Overview

Figure 2 illustrates an overview of our approach. Intuitively, the higher-precision network gains richer knowledge from the input and thus gives better performance due to its larger capacity, while a lower-precision network is more compact by sacrificing some performance. To take advantage of different precisions in the given $\mathcal{B}$, we propose two novel techniques for transferring knowledge from the existing higher precision "teacher" to the lower precision "student" while jointly optimizing a single network with all the bit-widths. Specifically, during the training, for the bit-width $b_k \in \mathcal{B}$, we first propose a mechanism to choose the best teacher from $\{b_1, b_2, \cdots, b_{k-1}\}$ adaptively to the current input by balancing the confidence of prediction and the distance in the model space (see Section 3.3). Second, we dynamically swap the blocks in the student with the corresponding blocks in the selected teacher network to better guide the training of the student (the current bit-width b_k) (see Section 3.4). We train the network using task-specific loss, such as cross-entropy loss, and distillation loss [19] between teacher and student (see Section 3.5). During inference, the trained network is directly set to different bit-widths, by truncating the least significant bits, to support instant adaptation for different deployment scenarios.

3.3. Dynamic Teacher Selection

Taking a higher precision network as a teacher benefits the performance of lower precision networks [24]. However, as shown in [10], a teacher with very high capacity might be too good, that the student is unable to mimic it, resulting in the devastation of the whole optimization. Thus, optimal selection of high-precision teacher for improving the performance of low-precision student is crucial while training a single deep network that can be quantized at different levels.

Given higher precision (e.g., bit-width $b_1, b_2, \cdots, b_{k-1}$) as teacher candidates for bit-width $b_k \in \mathcal{B}$, we introduce two types of criterion to select the best teacher adaptive to the current input and the training progress, namely the confidence of prediction and the distance in the model space. We adopt entropy $H(\cdot)$ of the output logits to evaluate the prediction confidence. While it is generally hard to measure the distance in the model space, it is specifically easy for adaptive networks due to shared weights. Since each precision model is quantized from the same set of weights, we simply define the distance of two quantized networks as:

$$||\mathcal{M}_{b_i} - \mathcal{M}_{b_j}|| = \sum_{l=1}^{L} D(\widehat{W}^l_{b_i}, \widehat{W}^l_{b_j}), \tag{3}$$

where L is number of layers and $D(\cdot, \cdot)$ is the average of entry-wise L_1 distance of two matrices. There exists a trade-off between prediction confidence and model distance. The network with higher precision usually has larger capacity and gives more confident prediction, while it is further from the lower precision network. Therefore, we choose the best teacher for bit-width b_k for the current batch via:

$$\underset{i \in \{b_1, b_2, \cdots, b_{k-1}\}}{\arg\min} \; H(y_i) + \lambda ||\mathcal{M}_i - \mathcal{M}_{b_k}|| \,, \tag{4}$$

where y_i is the soft logits (after Softmax) of the network with bit-width b_i and λ is a hyperparameter to balance the trade-off. When $\lambda \to 0$, the selection biased towards the highest precision. When λ is significantly large, it is biased towards choosing b_{k-1}. Notably, the preference for teacher also shifts during the training. Initially, the performance of higher-precision models improves faster and our mechanism favors the higher precision. As the training goes on, the difference among different precisions is mitigated and it favors the closer precision. Therefore, our proposed dynamic teacher selection strategy adapts better to the current input and the training progress than the manually fixed teacher. We provide visualizations on teacher selection later in Section 4.

3.4. Dynamic Block Swapping

Once the optimal teacher is selected, we transfer the teacher's knowledge by dynamically swapping low-precision blocks with high-precision to make better use of the information contained in teacher network. Specifically, execution in each precision merely requires the different quantization of the same set of parameters without changing the network architecture. Thus, we dynamically swap in the high-precision teacher blocks so that the low-precision student actively incorporates the teacher's intermediate knowledge [14, 48]. Moreover, higher-precision blocks experience less inaccurate gradient approximation [59]. So swapping in these blocks helps alleviate the difficulty of propagating gradient and results in the fast-start of convergence when training a lower-precision student. Note that we have a single network and we only switch the training precision of the same set (shared) of parameters instead of varying architectures.

For the l-th block or layer, let $\beta_l = \text{Bernoulli}(p_l)$ indicate whether the student block is executed ($\beta_l = 1$) or the teacher network is swapped in ($\beta_l = 0$):

$$A^{l+1} = \beta_l f(\widehat{A}^l_s, \widehat{W}^l_s) + (1 - \beta_l) f(\widehat{A}^l_t, \widehat{W}^l_t), \tag{5}$$

where $\widehat{A}_s^l$ and $\widehat{A}_t^l$ denotes the input activations of the student layer $f(\cdot, \widehat{W}_s^l)$ and the teacher layer $f(\cdot, \widehat{W}_t^l)$ correspondingly and A^{l+1} is the output activation.

Since top layers are less prone to the gradient issue and better taught by the soft logits than bottom layers, we prefer to train top layers more and then move to train the bottom layers. Therefore, we linearly increase the probability to execute the students with respect to the layer depth:

$$p_l = \min(1, (1 + l/L)p_1). \tag{6}$$

Following curriculum learning [4], we set the initial value for p_1 and gradually increase it to 1 during training, i.e. it auto-regresses towards training the network with all student blocks at the end. During inference, all the blocks are executed with the given precision without any swapping.

3.5. Optimization

During training, we gather losses of all the precisions and then update the network. Denote the total loss as $\mathcal{L}$, and the loss for the precision with bit-width b as $\mathcal{L}_b$, where $\mathcal{L} = \sum_{b \in \mathcal{B}} \mathcal{L}_b$. In classification tasks, $\mathcal{L}_b$ contains two parts: the cross-entropy loss ($\mathcal{L}_{ce}$) with respect to the true label y and a distillation loss computed by taking Kullback–Leibler ($\mathcal{L}_{kl}$) divergence between the output y_b and the soft logits y_t (after softmax) provided by the chosen teacher except for the highest precision which only has the first part:

$$\mathcal{L}_{kl}(y_t||y_b) = \sum_{i=1}^{m} (y_t)_i \log \frac{(y_t)_i}{(y_s)_i}, \tag{7}$$

$$\mathcal{L}_b = \mathcal{L}_{ce}(y_b, y) + \mathcal{L}_{kl}(y_t||y_b), \tag{8}$$

where m is the number of classes and $(\cdot)_i$ is the i-th element of the vector. Different from conventional knowledge distillation with a well-trained teacher in advance [60, 24], our model optimizes all the precisions jointly and collaboratively. Specifically, we learn all precisions with the same input batch and shared weights. Since all precisions are quantized from the same full-precision weights, the optimization processes for all precisions are intervened closely. We follow [22] and use a separate set of Batch Normalization layers and clipping level parameters for different precisions.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate the performance of our approach using CIFAR10 [26], ImageNet [37] for image classification and ActivityNet-v1.3 [13], Mini-Kinetics [9] for action recognition in videos. ActivityNet contains 10,024 videos for training and 4,926 videos for validation across 200 action categories. Mini-Kinetics-200 (assembled by [29]) is a subset of full Kinetics dataset [9] containing 121k videos for training and 10k videos for testing across 200 action classes.

Implementation Details. We use ResNet18, ResNet50 [35] and MobileNet V2 [39] to perform different experiments. For video classification, we adopt temporal segment network (TSN) [45] to aggregate the predictions over uniformly sampled 8 frames from the video. As shown in [57, 11], 8-bit model experiences no/little deficiency from full precision while 2-bit one always leads to a much worse performance. Thus, in our experiments, we set $\mathcal{B} = \{8, 6, 4, 2\}$. For ImageNet, ActivityNet and mini-Kinetics, we use standard architecture of ResNet18, ResNet50 and MobileNet V2, with the input size 224 x 224. For CIFAR10, we accommodate ResNet18 and MobileNet V2 to adapt for the input size 32x32 [56]. We switch BatchNorm (BN) and clipping values for each numerical precision. We use separate sets of learning parameters (learning rate, weight decay) for clipping values of each precision. We train our network with 160 epochs for CIFAR10 and 100 epochs for the other datasets. Following [57, 11], we do not quantize the input, first and last layers of the network. We follow [22] and initialize the network with full precision model except CIFAR10 since we found that low precision models in this setting do not converge. More details are included in supplementary material.

Baselines. We compare our approach with the following baselines. We first consider **Individual Quantization**, where we train a separate network for each precision. We also compare with **Direct Quantization** [22] methods (DQ) that directly quantize a higher precision network to a lower precision during inference without any extra training. We follow [52] to apply batch norm calibration (BN Calib.) to alleviate the discrepancy of the layer statistics when evaluated on another bit-width. We then compare with **Joint Training** baseline that trains a single network with all precisions simultaneously using the same BN layers and clipping values. **Switchable BN** considers the difference in layer statistics for different precisions and uses precision-specific BN layers. Finally, we compare our approach with state-of-the-art methods, **AdaBits** [22], **Any Precision** [51], and **Bit-Mixer** [5]. **Adabits** applies precision-specific clipping level parameters in addition to BN layers. **Any Precision** [51] distills knowledge from full precision on top of Switchable BN. **Bit-Mixer** first optimizes the network with activation quantization and then adopt quantization of both weights and activations.

Note that adaptive bit-width performance depends on two orthogonal components: (1) single-precision quantization scheme; (2) adaptive bit-width mechanism. AdaBits [22] uses Scale-adjusted quantizer, BitMixer [5] uses LSQ quantizer and Any precision [51] uses its own modified quantizer. Without unifying the quantizer, direct comparison between their original performance and ours is misleading when evaluating the proposed adaptive bit-width mechanisms. Thus, we adopt the same quantizer (modified DoReFa for weight quantization and PACT [11] for activation quantization) for all the baselines to enable fair comparison among SOTA

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	95.1	95.4	95.0	94.1	100
DQ (32 bit)	10.7	10.4	10.2	10.8	11.1
DQ (8 bit)	95.1	93.8	29.9	8.3	59.7
DQ (32 bit) + BN Calib	95.1	94.9	93.0	9.6	76.9
DQ (8 bit) + BN Calib	95.1	94.3	93.1	31.0	82.5
Joint Training	34.3	44.7	56.0	26.7	42.6
Switchable BN	94.6	94.5	94.5	92.3	99.2
AdaBits	94.4	94.2	94.2	92.4	99.0
Any Precision	94.2	94.2	94.2	92.3	98.8
Bit-Mixer	94.7	94.6	94.5	91.9	99.0
CoQuant (Ours)	95.2	95.4	95.1	94.1	**100.1**

Table 1: **ResNet18 on CIFAR10. CoQuant** achieves best overall performance $\Delta_{\mathcal{B}}$ (higher is better) among all baselines.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	94.2	93.8	93.6	89.0	100
DQ (32 bit)	10.0	. 9.9	10.0	10.1	10.8
DQ (8 bit)	94.2	93.0	80.3	10.9	74.3
DQ (32 bit) + BN Calib	91.4	91.3	89.8	38.2	83.3
DQ (8 bit) + BN Calib	94.2	94.1	93.7	77.6	96.9
Joint Training	14.2	15.3	29.2	46.1	28.6
Switchable BN	94.2	94.0	93.3	85.4	99.0
AdaBits	93.9	93.8	93.2	86.2	99.3
Any Precision	93.4	93.3	93.1	86.3	98.8
Bit-Mixer	94.1	94.0	92.9	86.7	99.3
CoQuant (Ours)	94.1	94.2	94.0	87.5	**99.8**

Table 2: **MobileNet V2 on CIFAR10. CoQuant** achieves best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	69.1	68.8	68.1	60.1	100
DQ (32 bit)	0.1	0.1	0.1	0.1	0.2
DQ (8 bit)	69.1	13.9	0.1	0.1	30.1
DQ (32 bit) + BN Calib	69.3	68.8	52.3	0.2	69.3
DQ (8 bit) + BN Calib	69.1	68.1	39.3	0.1	65.1
Joint Training	8.4	11.4	33.3	1.5	20.0
Switchable BN	67.9	67.7	66.5	54.0	96.0
AdaBits	67.9	67.7	66.5	54.1	96.1
Any Precision	67.4	67.3	66.7	54.0	95.8
Bit-Mixer	67.1	67.1	66.5	54.7	95.8
CoQuant (Ours)	67.9	67.6	66.6	57.1	**97.3**

Table 3: **ResNet18 on ImageNet. CoQuant** achieves the best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	66.2	66.5	66.6	62.4	100
DQ (32 bit)	0.4	0.5	0.5	0.5	0.7
DQ (8 bit)	66.2	62.1	1.9	0.5	49.3
DQ (32 bit) + BN Calib	65.9	65.8	54.4	0.5	70.3
DQ (8 bit) + BN Calib	66.2	65.4	54.5	0.6	70.3
Joint Training	0.5	0.5	0.5	0.7	0.9
Switchable BN	64.7	64.8	63.4	43.9	90.2
AdaBits	64.1	64.3	64.2	48.3	91.8
Any Precision	64.6	63.2	61.7	36.2	85.9
Bit-Mixer	65.8	66.1	64.7	53.4	**95.5**
CoQuant (Ours)	64.6	64.8	64.4	55.5	95.2

Table 4: **ResNet18 on Mini-Kinetics. CoQuant** achieves the second best overall performance $\Delta_{\mathcal{B}}$ among all baselines.

methods. We also adopt the scale-adjusted quantizer [22] to verify the effect of quantization scheme on adaptive bit-width performance of different methods (see Table 8).

Evaluation Metrics. We report top-1 accuracy for all precisions on CIFAR10 and ImageNet. For video classification, we compute top-1 clip accuracy and mean Average Precision (mAP) on Mini-Kinetics and ActivityNet respectively. Furthermore, we report a single relative performance $\Delta_{\mathcal{B}}$ with respect to Individual Quantization to show the overall performance of different baselines as:

$$\Delta_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \frac{M_i}{M_{PACT,i}} * 100\%,$$

where M_i and $M_{PACT,i}$ are the corresponding performance of models and individual quantization with bit-width b_i.

4.2. Results and Analysis

Table 1-6 show performance in six different pairs of datasets and backbones. We separate all methods to two groups, namely direct quantization methods and methods that are specific for training adaptive quantization networks.

Image Classification. We have the following key observations from Table 1-3. (1) On CIFAR-10, while Individual Quantization with 2 bits reaches similar performance to 8 bits using ResNet18, there is a large gap among 8-bit and 2-bit performance on MobileNet V2, which shows that MobileNet V2, a more compact architecture, is difficult to quantize at lower bits. Similarly, the performance difference on ImageNet indicates that 2-bit quantization is very challenging on this large scale dataset, as shown in [11, 55, 60]. (2) Direct Quantization from either 32-bit full precision or 8-bit precision experiences serious performance degradation when evaluated with a different precision far from the training one without BN Calibration [53]. It is due to the mismatch of layer statistics between different precisions. After re-calibrating BN layer parameters, Direct Quantization achieves good performance in 8 and 6 bits, especially quantized from 32-bit full precision model. However, it is still unable to recover the performance in low-precision regime, showing the necessity of joint learning approaches. (3) When trained under all precisions, Switchable BN, AdaBits [22] and Any Precision [51] largely improve the performance over Joint Training, which demonstrates the importance of switching BN layer parameters for different precisions in image classification. **CoQuant** consistently outperforms all of the methods on all three image classification tasks, significantly at the lower precisions, which is of great practical value. Notably, **CoQuant** improves 1.7% in 2-bit performance over AdaBits for ResNet18 on CIFAR10, 1.3% for MobileNet V2 on CIFAR10 and 3.0% for ResNet18 on ImageNet without sacrificing the higher-precision performance. This is due to our two novel components working in concert: dynamically selecting the best high-precision teacher and then swapping

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	65.3	65.5	64.3	59.9	100
DQ (32 bit)	0.7	0.7	0.7	0.7	1.1
DQ (8 bit)	65.3	61.2	4.0	0.7	50.2
DQ (32 bit) + BN Calib	67.6	67.5	57.4	0.7	74.3
DQ (8 bit) + BN Calib	65.3	64.7	56.3	0.9	71.9
Joint Training	0.8	0.9	0.7	0.7	1.2
Switchable BN	64.6	64.5	63.3	45.3	92.9
AdaBits	64.8	64.7	64.2	51.3	95.9
Any Precision	64.3	64.9	63.7	48.2	94.7
Bit-Mixer	63.2	63.0	62.4	55.2	95.5
CoQuant (Ours)	64.5	64.7	64.2	57.4	**98.3**

Table 5: **ResNet18 on ActivityNet.** **CoQuant** achieves the best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	70.0	69.0	68.9	64.8	100
DQ (32 bit)	0.7	0.7	0.7	0.7	1.1
DQ (8 bit)	70.0	39.1	0.7	0.7	39.7
DQ (32 bit) + BN Calib	71.5	71.0	45.8	0.8	68.2
DQ (8 bit) + BN Calib	70.0	66.1	6.2	0.8	51.5
Joint Training	0.7	0.7	0.8	0.7	1.1
Switchable BN	68.3	68.3	67.7	57.1	95.7
AdaBits	68.3	68.6	67.9	57.2	96.0
Any Precision	69.1	69.2	67.8	60.3	97.6
Bit-Mixer	67.2	66.8	66.4	59.3	95.3
CoQuant (Ours)	69.3	69.2	68.4	64.6	**99.6**

Table 6: **ResNet50 on ActivityNet.** **CoQuant** achieves the best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

low-precision blocks with the selected teacher to balance the optimization difficulty while training the adaptive quantization network with different precisions. (4) **CoQuant** outperforms Bit-Mixer [5] on all three image classification tasks showing that our proposed multi-teacher knowledge distillation method with joint training is better than two-phase optimizations especially for 2-bit performance. (5) On CIFAR 10 with ResNet 18, **CoQuant** yields a small standard deviation 0.08% over multiple runs for overall performance comparing with AdaBits (0.16%) , Any Precision (0.26%) and Bit-Mixer (0.18%). It shows **CoQuant** gives more stable performance than state-of-the-art methods. (6) Interestingly, on CIFAR10, **CoQuant** slightly improves the higher-precision performances over Individual Quantization (Table 1 and 2). We believe this is because the network easily gains information from other precisions similar to positive transfer in multi-task learning [31, 43, 28, 41].

Video Classification. Table 4-6 summarizes the results on video classification tasks. Video datasets introduce tougher classification tasks than image datasets due to the rich temporal information present in videos and results in more diverse optimization difficulty from high to low precisions, leading to significant drop in performance from 8 to 2 bits in Individual Quantization. Similar to image classification, Direct Quantization is unable to achieve plausible performance for 2-bit model. For 2-bit performance, **CoQuant** outperforms AdaBits by a larger margin than image classification: 7.2%, 6.1% and 7.4% for ResNet18 on Mini-Kinetics, ResNet18 on ActivityNet and ResNet50 on ActivityNet respectively. Comparing with Bit-Mixer, **CoQuant** achieves 2% to 5% improvement on 2-bit performance. To summarize, **CoQuant** improves the overall performance by about $2\% - 4\%$ over SOTA methods in most cases, showing its efficacy not only for image classification, but also for the challenging action classification on videos.

Visualizations on Dynamic Teacher Selection. We visualize the selection of high-precision teachers for the 2-bit low-precision student throughout the training process in two different scenarios: ResNet18 on CIFAR10, and ResNet18 on ActivityNet (Figure 3). We count the number of selected teachers for the input mini-batch in an epoch. We observe the shift of preference in all three cases. Specifically, our dynamic teacher selection strategy starts to favor the highest precision (8-bit) since it converges much faster at the beginning. However, as performances of 6-bit and 4-bit gradually improve during training, it is more likely to choose the lower bit-width as the teacher for 2-bit. It is helpful since 6-bit and 4-bit models are often easier for 2-bit one to mimic.

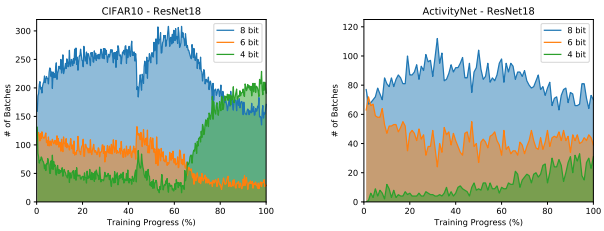

Figure 3: **Dynamic Teacher Selection.** Initially, our method prefers 8-bit teacher and the preference shifts to 6-bit and even 4-bit during the training. Best viewed in color.

Zero-Shot Testing. We show the robustness of **CoQuant** by evaluating the quantized model on missing bit-widths (7, 5, and 3 bits) when a network is trained with 8, 6, 4, and 2 bits. Specifically, after training with 8, 6, 4, and 2 bits, we calibrate the model to obtain BN layer parameters of the remaining 3, 5, 7 bit-widths so that the model can be executed at different precisions from 2 to 8 bits. Table 7 shows that **CoQuant** achieves the best performance in all the 3 remaining precisions in both cases. Not surprisingly, All the three methods suffer from performance drop to some extent when executed with 3 bit compared to their 4-bit performance. Notably, on ActivityNet-ResNet50, while the 3-bit performance of both Switchable BN and AdaBits are lower than their 2-bit performance, **CoQuant** yields 66.8% when evaluated with 3 bit, which is 2.2% better than its 2-bit performance.

Methods	7-bit	5-bit	3-bit
CIFAR10 – ResNet18			
Switchable BN	94.4	94.4	92.4
AdaBits	94.3	94.5	93.4
CoQuant (Ours)	**95.1**	**95.3**	**94.6**
ActivityNet – ResNet50			
Switchable BN	68.8	68.5	56.4
AdaBits	68.7	68.5	57.0
CoQuant (Ours)	**70.0**	**70.0**	**66.8**

Table 7: Zero Shot Testing.

Comparison using Different Quantizer. We use the modified scale-adjusted quantizer (SAT) for training {5, 4, 3} bits jointly on CIFAR10 (i.e., without 2-bit as primarily done in [22]) and **CoQuant** significantly outperforms AdaBits including Any Precision and Bit-Mixer (see Table 8). These results show effectiveness of our adaptive bit-width mechanism irrespective of the single-precision quantization scheme used for quantizing deep networks.

Methods	5-bit	4-bit	3-bit
AdaBits	89.7	89.8	88.6
Any Precision	91.0	91.1	90.7
Bit-Mixer	93.6	92.9	91.5
CoQuant	93.5	93.2	92.1

Table 8: SAT Quantization with ResNet18 on CIFAR10.

4.3. Ablation Studies

Dynamic Block Swapping. We test **CoQuant** without dynamic block swapping where the selected teacher only provides soft-logits to compute distillation loss. Better performance of our full model over "**CoQuant** w/o Swap" shows that dynamic swapping is helpful to effectively transfer knowledge and ease the optimization difficulty while training adaptive networks (1 vs 14). We also find that reversing the swapping schedule, i.e., earlier layers get swap less, leads to an overall performance of 98.7% which is 1.4% lower than swapping earlier layers more with the teachers (2 vs 14). This demonstrates that earlier layers need to swap more for better optimization of the network with adaptive bit-widths. We also replace linear scheduler with a cosine scheduler to increase p_l from initial value to 1 but did not observe any noticeable difference in performance (3 vs 14). So, we use linear scheduler for experiments in this paper.

Distillation Loss. We remove the distillation loss from **CoQuant** and transfer knowledge only with dynamic block swapping. The performance of "**CoQuant** w/o DL" in Table 9 (4 vs 14) shows that it is important to use the teacher's soft targets in addition to dynamic swapping for better knowledge transfer across the low and high precisions.

Dynamic Block Swapping vs Feature Distillation. We also compare with intermediate feature distillation (IFD) [36, 54] by minimizing the L1 difference of intermediate activations and observe that feature distillation ("**CoQuant** w/ IFD") does not give competitive results with swapping (5 vs 14). This conforms that dynamic block swapping not only transfers intermediate knowledge but also helps in propagating gradient through low-precision networks.

Effectiveness of Dynamic Teacher Selection. We compare with three teacher selection variants by using the highest capacity (8-bit) network as the teacher for all lower precisions (6), using the nearest superior bit-width as the teacher for the current precision (7) and finally with random selection that randomly selects a higher precision as the teacher for each mini-batch (8). Table 9 shows that **CoQuant** outperforms all three methods (14 vs {6, 7, 8}), which demonstrates that both the predication confidence and model distance play an important role in optimal selection of teacher for transferring knowledge across different precisions.

	Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_B \uparrow$
(1)	**CoQuant** w/o Swap	94.8	94.8	94.6	93.2	99.4
(2)	Reverse Swap	94.5	94.2	93.8	92.0	98.7
(3)	Swap (Cosine)	95.2	95.3	95.2	94.1	100.1
(4)	**CoQuant** w/o DL	94.6	94.5	94.6	92.7	99.2
(5)	**CoQuant** w/ IFD	94.7	94.7	94.7	93.3	99.5
(6)	Highest Capacity	94.0	94.1	93.9	92.0	98.6
(7)	Recursive	93.2	93.3	93.2	91.2	97.7
(8)	Random	94.8	94.8	94.6	93.2	99.4
(9)	Average	94.6	94.5	94.5	92.9	99.2
(10)	Learnable Weighted	94.2	94.2	94.1	93.0	99.0
(11)	MinLogit	94.3	94.2	93.9	93.2	98.9
(12)	Progressive (Descend)	91.8	91.9	92.2	92.9	97.1
(13)	Progressive (Ascend)	94.7	94.5	94.2	87.0	97.6
(14)	**CoQuant** (Ours)	95.2	95.4	95.1	94.1	**100.1**

Table 9: **Ablation Study using ResNet18 on CIFAR10.** Our full model claims the best performance over all the ablated variants.

Comparison with Multi-Teacher Distillation Methods. We also compare our method with three different multi-teacher distillation methods, namely Average Ensemble (9), Learnable Weighted Ensemble (10) and MinLogit (11). In Average Ensemble, we simply use the average of the soft logits of multiple teacher candidates, while in Learnable Ensemble, we use a learnable linear combination of soft logits to compute the distillation loss. MinLogit selects minimum element of each row of the soft logits matrix constructed by taking difference between logit values for each of the teacher [17]. As seen from Table 9, **CoQuant** claims better performance over all three multi-teacher distillation methods (14 vs {9, 10, 11}), which shows that our proposed dynamic select and swap strategy makes better use of the knowledge provided by multiple teacher candidates.

Comparison with Progressive Training. Following [38], we progressively train a quantized model with multiple bit-widths in a descending order (*i.e.*, from 8 bits to 2 bits by sequentially finetuning). Table 9 shows that it fails to preserve the performance of higher precisions and **CoQuant** outperforms it by 3.0% in overall performance (12 vs 14). We also progressively train another model with multiple bit-widths in a ascending order (from 2 bits to 8 bits) and observe that joint training under all precisions using **CoQuant** is still more effective than the ascending strategy (13 vs 14), especially at low precision ($+7.1\%$ for 2-bit).

5. Conclusion

In this paper, we present two improved techniques for quantizing a single deep network with adaptive bit-widths of weights and activations. Given a low-precision student, we choose the best high-precision teacher adaptively to the current input while randomly replacing low-precision student blocks with the corresponding higher-precision blocks to alleviate the difficulty of propagating gradients. We show effectiveness of our approach on four image and video datasets, outperforming several competing methods.

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

[2] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[5] Adrian Bulat and Georgios Tzimiropoulos. Bit-mixer: Mixed-precision networks with runtime bit-width selection. In *Proceedings of the IEEE international conference on computer vision*, 2021.

[6] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[7] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5918–5926, 2017.

[8] 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.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[10] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4794–4802, 2019.

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

[12] 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 *IJCAI*, pages 2080–2087, 2020.

[13] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[14] Shipeng Fu, Zhen Li, Jun Xu, Ming-Ming Cheng, Gwanggil Jeon, and Xiaomin Yang. Interactive knowledge distillation. *arXiv preprint arXiv:2007.01476*, 2020.

[15] Yonggan Fu, Zhongzhi Yu, Yongan Zhang, Yifan Jiang, Chaojian Li, Yongyuan Liang, Mingchao Jiang, Zhangyang Wang, and Yingyan Lin. Instantnet: Automated generation and deployment of instantaneously switchable-precision networks. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 757–762. IEEE, 2021.

[16] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6926–6935, 2018.

[17] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

[18] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[21] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

[22] 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*, pages 2146–2156, 2020.

[23] Eunwoo Kim, Chanho Ahn, and Songhwai Oh. Nestednet: Learning nested sparse structures in deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8669–8678, 2018.

[24] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

[25] Animesh Koratana, Daniel Kang, Peter Bailis, and Matei Zaharia. LIT: Learned intermediate representation training for model compression. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3509–3518. PMLR, 09–15 Jun 2019.

[26] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[27] Fengfu Li, Bo Zhang, and Bin Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

[28] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and

Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. 2020.

[30] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

[31] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[32] Eunhyeok Park, Junwhan Ahn, and Sungjoo Yoo. Weighted-entropy-based quantization for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5456–5464, 2017.

[33] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[34] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[38] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[39] 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*, pages 4510–4520, 2018.

[40] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1832–1841, 2018.

[41] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[42] 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*, 2020.

[43] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019.

[44] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8612–8620, 2019.

[45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

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

[47] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018.

[48] Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*, 2020.

[49] Taojiannan Yang, Sijie Zhu, Chen Chen, Shen Yan, Mi Zhang, and Andrew Willis. Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In *European Conference on Computer Vision (ECCV)*, 2020.

[50] Haibao Yu, Qi Han, Jianbo Li, Jianping Shi, Guangliang Cheng, and Bin Fan. Search what you want: Barrier panelty nas for mixed precision quantization. *arXiv preprint arXiv:2007.10026*, 2020.

[51] Haichao Yu, Haoxiang Li, Honghui Shi, Thomas S Huang, Gang Hua, et al. Any-precision deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[52] Jiahui Yu and Thomas Huang. Network slimming by slimmable networks: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 3, 2019.

[53] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

[54] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

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

[56] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pages 9597–9608, 2019.

[57] 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 arXiv:1606.06160*, 2016.

[58] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *Advances in neural information processing systems*, pages 7517–7527, 2018.

[59] Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian Reid. Training quantized neural networks with a full-precision auxiliary module. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1488–1497, 2020.

[60] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7920–7928, 2018.