Quantized Feature Distillation for Network Quantization

Ke Zhu, Yin-Yin He, Jianxin Wu*

State Key Laboratory for Novel Software Technology, Nanjing University, China
zhuk@lamda.nju.edu.cn, heyy@lamda.nju.edu.cn, wujx2001@nju.edu.cn

Abstract

Neural network quantization aims to accelerate and trim full-precision neural network models by using low bit approximations. Methods adopting the quantization aware training (QAT) paradigm have recently seen a rapid growth, but are often conceptually complicated. This paper proposes a novel and highly effective QAT method, quantized feature distillation (QFD). QFD first trains a quantized (or binarized) representation as the teacher, then quantize the network using knowledge distillation (KD). Quantitative results show that QFD is more flexible and effective (i.e., quantization friendly) than previous quantization methods. QFD surpasses existing methods by a noticeable margin on not only image classification but also object detection, albeit being much simpler. Furthermore, QFD quantizes ViT and Swin-Transformer on MS-COCO detection and segmentation, which verifies its potential in real world deployment. To the best of our knowledge, this is the first time that vision transformers have been quantized in object detection and image segmentation tasks.

Introduction

Network quantization transfers a full precision (FP) network's weights and activations to their fixed point approximations without obvious accuracy drop. Recently, various approaches (Lee, Kim, and Ham 2021; Liu et al. 2022; Lin et al. 2022) have been proposed and quantization aware training (QAT) becomes a mature paradigm for its ability to recover network accuracy even in extreme low bit settings.

Modern QAT methods are based on a general principle: optimizing quantization interval (parameters) with task loss (Jung et al. 2019). Many variants have been put forward, such as non-uniform quantization (Yamamoto 2021; Liu et al. 2022), complex gradient approximation (Gong et al. 2019; Lee, Kim, and Ham 2021) or manually designed regularization (Zhuang et al. 2020; Lee et al. 2021). These methods are complex and *simplicity* is often sacrificed. Another line of QAT methods introduce knowledge distillation (Hinton et al. 2015) into the quantization phase. While the idea of quantization KD is straightforward (i.e., a full precision teacher helps recover the accuracy of a quantized student network), the implementation includes heuristic stochastic precision (Boo et al. 2021), manually defined auxiliary modules (Zhuang et al. 2020) or several stages (Kim et al. 2019). All these methods resort to logit distillation, which are then difficult to be applied in *other computer vision tasks* (e.g., object detection).

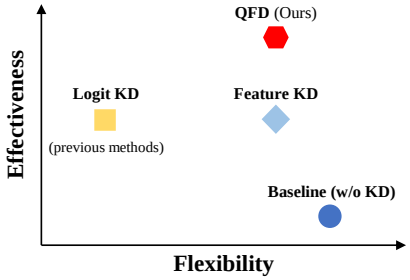
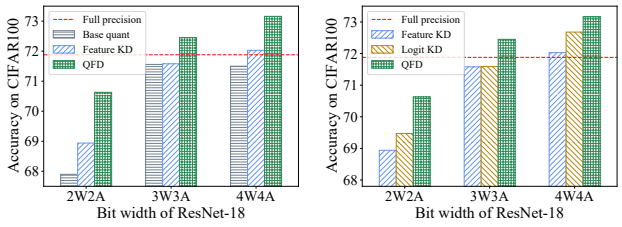


Figure 1: Illustration of different quantization KD methods. 'Flexibility' stands for adaptability to different vision tasks, and 'Effectiveness' indicates whether or not a method is friendly for network quantization (i.e., the ability to achieve high accuracy by the quantized network).

In this paper, we propose a novel, simple, and effective distillation method targeting network quantization, *quantized feature distillation* (QFD). Our motivation came from an important result in Wu and Luo (2018): an *FP model with only its output features binarized* can achieve similar or better accuracy compared with the full FP model. Then, it must be advantageous to use these quantized features as a teacher signal to help quantize the rest of the network (i.e., the student). On one hand, *feature distillation is more flexible in comparison with logit distillation*, especially in tasks like object detection (Yang et al. 2022); on the other hand, it is natural to conjecture that *it will be easier for the fully quantized student to mimic a fixed point feature representation rather than directly mimicking the floating-point logits or features*. In other words, the proposed QFD will be both more flexible and more effective (quantization friendly). This conjecture (and motivation for our QFD) is illustrated in Figure 1, which is firmly supported by the experimental results in Figure 2 and Table 1.

These experiments that verified our motivation were carried out on CIFAR100 with ResNet-18 using 4 different quantization methods: the quantization baseline (a uniform

*J. Wu is the corresponding author.



(a) compare with baseline (b) compare with KD

Figure 2: The test accuracy of ResNet-18 models on CIFAR100 under 2-bit, 3-bit and 4-bit quantization settings. The left figure shows the results of the proposed QFD (with its teacher's feature binarized), float feature distillation and the baseline quantization, while the right compare three different KD methods (feature KD, logit KD and QFD). This figure is best viewed in color.

Method	Teacher bit width	1/1	2/2	3/3	4/4
Baseline		57.38	67.90	71.56	71.50
QFD	1 (71.14)	**62.62**	**70.63**	**72.45**	**73.16**
	4 (71.30)	61.60	69.63	71.60	72.18
	8 (71.53)	61.45	69.98	71.43	72.28
	32 (71.88)	59.66	68.94	71.58	72.03

Table 1: ResNet-18 test accuracy on CIFAR-100. "W/A" means the weight and activation bit width in quantization. 'Baseline' is the quantization without distillation. During QFD, the teacher's feature is quantized to 1, 4 and 8 bits with the teacher accuracy inside the brackets '()'. The 32-bit feature refers to full precision (normal feature distillation).

quantizer without KD), logit distillation (logits as teacher signal), feature distillation (FP features as teacher signal) and our QFD (quantized features as teacher signal). All the 3 KD methods adopt the same quantizer as the baseline, and the teacher's feature is binarized (1-bit) in QFD. As shown in Figure 2, QFD is not only superior than float feature distillation, but also surpasses logit distillation in all settings. The improvement of QFD over baseline is also consistent and large enough to recover the full precision model's accuracy (3-bit and 4-bit). We further quantize the teacher's feature to 4 bit and 8 bit using QFD. As shown in Table 1, all feature distillation methods show improvement over the baseline. As teacher's bit width decreases, the accuracy of the teacher model drops, but the final distillation results have been consistently improved. These results show that our motivation is valid: *quantized features are better teachers for network quantization!*

We tested QFD in not only classification but also detection and segmentation, and achieved state-of-the-art accuracy on diverse network structures (ResNet, MobileNet and ViT). Our contributions can be summarized as follows:

- A novel quantization aware training KD method that is easy to implement.
- Remarkable accuracy advantage on classification, detection and segmentation benchmarks over previous quantization aware training methods.
- A first trial of quantizing the vision transformer structure on common object detection and segmentation tasks.

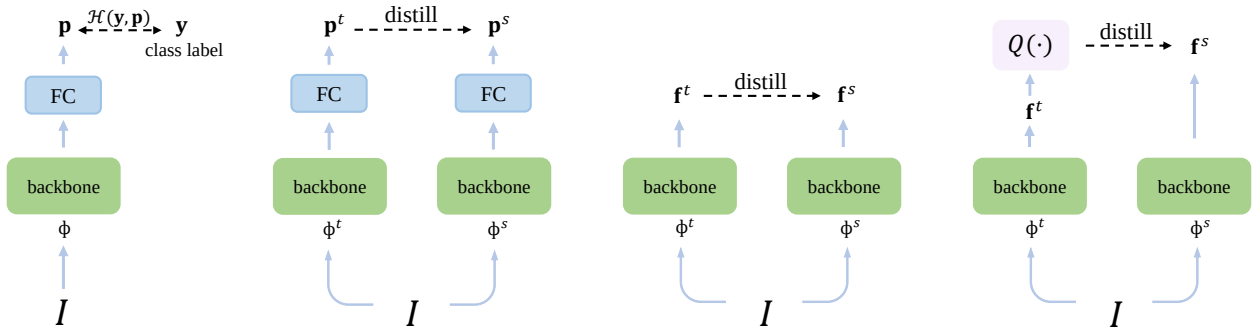
Related Work

Neural network quantization can be categorized into two paradigms: quantization aware training (QAT) and post training quantization (PTQ). We adopt QAT in this paper. In this section, we will describe the basics, knowledge distillation, and vision transformers in QAT.

Quantization aware training. QAT (Nagel et al. 2021) is a powerful paradigm to tackle low bit (e.g., 3- or 4-bit) quantization without significant accuracy drop. Integrating quantization operations into the computation graph is the key in QAT such that the weights and quantization parameters can be learned simultaneously through back-propagation. Early methods in this family focused on how to binarize models (Courbariaux, Bengio, and David 2015; Rastegari et al. 2016), fit quantizers with statistics (Zhou et al. 2016; Cai et al. 2017; Choi et al. 2018), or minimize local error (Zhang et al. 2018), but they suffer from the incomplete or sub-optimal issue. Modern QAT methods adopt the principle of optimizing quantization interval with task loss (Esser et al. 2020; Jung et al. 2019) and resort to much more complicated techniques, including non-uniform quantizer (Yamamoto 2021; Liu et al. 2022), gradient approximation (Kim, Lee, and Ham 2021; Lee, Kim, and Ham 2021; Gong et al. 2019) or extra regularization (Han et al. 2021; Lee et al. 2021; Zhuang et al. 2020). Their simplicity and effectiveness, however, remain as big challenges.

Knowledge distillation in quantization. KD (Hinton et al. 2015) is popular in various computer vision tasks (He, Wu, and Wei 2021; Guerra et al. 2020) and has gradually emerged in quantization aware training (Mishra and Marr 2018; Polino, Pascanu, and Alistarh 2018; Boo et al. 2021; Zhuang et al. 2020). The point of quantization KD is neat: utilizing a full-precision teacher to recover the accuracy of the quantized student network. However, recent methods are short of simplicity in that they involve complex stages (Kim et al. 2019), auxiliary modules (Zhuang et al. 2020) and dedicated mixed precision adjustment (Boo et al. 2021). Moreover, these methods all adopt logit distillation, which is inflexible when KD is applied to object detection quantization (Guo et al. 2021). Instead, we propose our quantized feature KD, which is both quantization friendly in terms of accuracy and flexible in terms of pipeline design.

Quantizing vision transformers. Vision Transformers (ViT) have boosted numerous vision tasks (Dosovitskiy et al. 2021; Touvron et al. 2021) and there is an urgent need to precisely quantize them so as to facilitate their practical usage (Sun et al. 2022). Recent methods (Yuan et al. 2021; Liu et al. 2021b; Lin et al. 2022) tried post training quantization (Nagel et al. 2021) techniques to quantize ViT to 6- or 8-bit for only image classification. Low bit (3- or 4-bit) quantization and its applicability to detection and segmentation of ViT and variants remain unexplored. For the first time, we will answer both questions by exploring its quantization performance in all these settings and tasks.



(a) Baseline (b) Logit distillation (c) Feature distillation (d) QFD (ours)

Figure 3: The baseline quantization method and 3 different knowledge distillation quantization methods (logit distillation, feature distillation and our QFD). I is the input image, ϕ, $\mathbf{p}$ and $\mathbf{f}$ mean backbone, logits and feature, respectively. t and s denote the teacher and the student, respectively; $Q(\cdot)$ is the process of quantizing the teacher's features. The common cross entropy loss $\mathcal{H}(\mathbf{y}, \mathbf{p})$ is calculated using the true label $\mathbf{y}$ and logits $\mathbf{p}$ as shown in (a). For clarity, we do not show the cross entropy loss $\mathcal{H}(\mathbf{y}, \mathbf{p}^s)$ of all 3 distillation methods in (b), (c) and (d). This figure is best viewed in color.

The Proposed Method

We have revealed the motivation of the proposed QFD method in the introduction, and the results in Figure 2 and Table 1 not only supported this motivation but also initially verified the effectiveness of QFD. After introducing the preliminaries of neural network quantization (the QAT baseline we use), we will then move on to a detailed description of our proposed method.

Preliminaries

We adopt the method in Lee, Kim, and Ham (2021) as our baseline method, which is a uniform quantizer composed of normalization, quantization and de-quantization steps.

For any given full-precision data v (a certain layer's weight or activation in the neural network), we define the quantization parameter l and u, which represent the lower bound and upper bound of the quantization interval, respectively. The normalization step is as follow:

$$\hat{v} = clip\left(\frac{v-l}{u-l}, 0, 1\right), \tag{1}$$

where $clip(\cdot, min, max)$ clips data that lies outside the min-max range. Then, a quantization function is used,

$$\widetilde{v} = \frac{\lfloor (2^b-1)\hat{v} \rceil}{2^b-1}, \tag{2}$$

in which $\lfloor \cdot \rceil$ is the rounding function and b stands for the bit width of quantization. The operation $\lfloor (2^b-1)\hat{v} \rceil$ maps $\hat{v}$ from the range $[0, 1]$ to a discrete number in $\{0, 1, ..., 2^b - 1\}$. Finally, a de-quantization step is applied to output the quantized weight $\overline{v}_W$ or activation $\overline{v}_A$:

$$\overline{v}_W = 2(\widetilde{v} - 0.5), \text{ or,} \tag{3}$$

$$\overline{v}_A = \widetilde{v}, \tag{4}$$

where the quantized weight $\overline{v}_W$ is roughly symmetric about zero and the quantized activation $\overline{v}_A$ is positive considering the ReLU activation. Similar to Lee, Kim, and Ham (2021), we use a trainable scale parameter α that is multiplied by the output quantized activation.

During training, we adopt the straight through estimator (STE) (Bengio, Léonard, and Courville 2013) to approximate the gradient of the rounding operator as 1:

$$\frac{\partial \lfloor x \rceil}{x} = 1. \tag{5}$$

The model's weights and these quantization parameters are learned simultaneously through back propagation.

Quantized feature distillation

We first define the basic notation in quantization aware training, then introduce our quantized feature distillation method, which is illustrated in Figure 3. For a given image I, it is first sent to a feature extractor $\phi(\cdot, \theta, \theta_q)$ (the backbone, e.g., a CNN or ViT) to get a full precision feature vector $\mathbf{f} \in \mathbb{R}^D$ (which is often obtained by a global average pooling),

$$\mathbf{f} = \phi(I, \theta, \theta_q), \tag{6}$$

in which D is the feature dimensionality, θ and θ_q represent the weight parameters and the quantization parameters of the model, respectively. $\mathbf{f}$ is passed through a classifier to get the final logit $\mathbf{p} \in \mathbb{R}^C$ with C classes, which produces the cross entropy loss $\mathcal{H}(\mathbf{y}, \mathbf{p})$ along with the true class label $\mathbf{y} \in \mathbb{R}^C$. The parameters θ and θ_q are learned through back propagation.

For our proposed QFD method, an image I is separately sent to the teacher network $\phi^t(\cdot, \theta^t)$ and the student network $\phi^s(\cdot, \theta^s, \theta^s_q)$ to get features $\mathbf{f}^t$ and $\mathbf{f}^s$, respectively. The teacher's full precision feature $\mathbf{f}^t$ will be quantized into lower bit (e.g., 1-bit or 4-bit) representation:

$$\mathbf{f}^t \xrightarrow{Q(\cdot)} \overline{\mathbf{f}}^t, \tag{7}$$

where the quantizer $Q(\cdot)$ is defined in Equations (1)–(4). The feature quantizer $Q(\cdot)$ follows the activation quantization process described in the preliminaries.

The teacher's quantized feature then acts as the supervision signal to guide quantization of the student network by the mean squared loss $\mathcal{L}(\cdot,\cdot)$, and the student still produces its usual cross entropy loss $\mathcal{H}(\cdot,\cdot)$ with the true label $\mathbf{y}$. The overall optimization objective is:

$$\arg\min_{\theta^s,\theta_q^s} \lambda\mathcal{L}(\mathbf{f}^s,\overline{\mathbf{f}^t}) + (1-\lambda)\mathcal{H}(\mathbf{y},\mathbf{p}^s)\,. \tag{8}$$

Here λ is used to weigh the importance of the distillation loss with respect to the cross entropy loss. For simplicity, we set $\lambda = 0.5$ except in ablation studies.

Experimental Results

In this section, we will first describe the general experimental settings, then present the results of our QFD on classification, object detection and segmentation benchmarks.

Experimental settings

During training, we first take a few epochs (roughly 1/10 of the total number of training epochs) to quantize the teacher's feature to fixed low bit (e.g., 2-bit) before starting our QFD training. Following previous QAT works (Zhuang et al. 2020; Lee, Kim, and Ham 2021), we conduct our experiments on the CIFAR, ImageNet, CUB and MS-COCO datasets. All experiments use PyTorch (Paszke et al. 2019) with 8 GeForce RTX 3090. The evaluation metrics for classification and detection (segmentation) are top-1 (top-5 is also used on ImageNet) accuracy and AP (average precision), respectively.

Classification settings. We experiment with ResNet-20 on CIFAR10 and ResNet-18/32 on CIFAR100. On both CIFAR datasets (Krizhevsky 2009), we use SGD with learning rate of 0.004, weight decay of 0.0005 and train 200 epochs in total. The input resolution is 32×32, and random flip and random crop are used as data augmentation. On ImageNet (Russakovsky et al. 2015), we train ResNet-18, ResNet-34 and MobileNet-v2 for 100 epochs. The initial learning rate and the momentum is 0.01 and 0.9, respectively. The weight decay is set to 1e-4, 5e-5 and 2.5e-5 for 4-bit, 3-bit and 2-bit, respectively, following Han et al. (2021); Esser et al. (2020). We adopt random resized crop and random flip as data augmentation and set input resolution as 224×224. On CUB200 (Wah et al. 2011), resolution and augmentation are the same as those on ImageNet. We train ViT Small, ViT Base, Deit Small, Deit Base, Deit Tiny for 90 epochs with batch size 64, following a cosine scheduler. The learning rate and weight decay are 5e-3 and 5e-4, respectively. We take 3 runs for CIFAR and CUB since these results have larger variance.

Object detection and segmentation settings. We train RetinaNet (Lin et al. 2017b) detectors with ResNet as backbones, and explore vision transformer detection and segmentation quantization using ViT and Swin Transformer (Liu et al. 2021a) pretrained with the newly proposed self-supervised method MAE (He et al. 2022). The object detector structure follows Li et al. (2022). For RetinaNet, we train 90k iteration with a base learning rate 0.001. Following Yamamoto (2021), we quantize all the layers (backbone, FPN and detection head) except the input and the output of the whole network, and utilize BN after FPN and detection head. To implement our QFD method, we quantize the teacher's FPN layer output to 8-bit and then use this 'quantized feature' for distillation, following the concept of previous object detection distillation works (Guo et al. 2021). For ViT and Swin Transformer, we quantize all the linear layer in their backbone and evaluate them on detection and segmentation tasks. All these models take 2 runs on MS-COCO and are implemented with Detectron2 (Wu et al. 2019).

Model	Methods	32/32	2/2	3/3	4/4
R20	LQ-Net	92.1	90.20	91.60	-
	SPEQ	92.1	91.40	-	-
	APRT*	91.6	88.60	91.80	92.20
	Baseline	92.1	90.49	91.74	92.09
	Feature KD		90.84	91.98	91.12
	Logit KD		90.61	91.78	92.02
	QFD		**91.41**	**92.64**	**93.07**

Table 2: ResNet-20 top-1 accuracy on CIFAR10. "W/A" represent the bit-width of weight and activation. SPEQ (Boo et al. 2021) and APRT* (Mishra and Marr 2018) are two logit distillation methods.

Model	Methods	32/32	1/1	2/2	3/3	4/4
R18	Auxi	-	-	67.90	-	-
	Baseline	71.9	57.38	67.90	71.56	71.50
	Feature KD		59.66	68.94	71.58	72.03
	Logit KD		62.60	69.47	71.59	72.68
	QFD		**62.62**	**70.63**	**72.45**	**73.16**
R32	APRT*	70.8	-	63.50	70.30	71.50
	Baseline	70.5	56.09	67.38	69.24	70.13
	Feature KD		54.60	67.08	69.43	70.23
	Logit KD		55.46	67.72	70.25	71.14
	QFD		**56.84**	**68.30**	**70.65**	**71.52**

Table 3: ResNet-18/32 on CIFAR-100, "W/A" represents the bit width of weight and activation. Both Auxi (Zhuang et al. 2020) (a variant of KD) and APRT* utilize logit distillation.

Classification results

CIFAR10/100. We first validate our proposed QFD method using ResNet models on CIFAR10 and CIFAR100, each containing 50,000 training images and 10,000 validation images, while the latter serves as a finer categorization (100 classes) than the former (10 classes).

For ResNet-20 models on CIFAR10 (results shown in Table 2), we run the baseline, feature knowledge distillation ('Feature KD'), logit distillation ('Logit KD') and our proposed quantized feature distillation method ('QFD'). We quantize ResNet-20 to 2-bit, 3-bit and 4-bit for both weights and activations ("W/A"). Following previous work (Zhang et al. 2018; Boo et al. 2021), we quantize all the layers except the input to the backbone and the last fully connected layer (i.e., the classifier). SPEQ (Boo et al. 2021) and APRT* both utilize the logit distillation method, and LQ-Net (Zhang et al. 2018) is a quantization aware training method. As can be seen in Table 2, our QFD surpasses previous knowledge

Architecture	Methods	32/32	2/2	3/3	4/4
ResNet-18	Auxi	-	66.7/87.0		
	SPEQ (Boo et al. 2021)	70.3	67.4		
	QIL (Jung et al. 2019)	70.2	65.7	69.2	70.1
	LSQ + BR(Han et al. 2021)	70.5	67.2/87.3	69.9/89.1	70.8/89.6
	EWGS (Lee, Kim, and Ham 2021)	69.9	67.0	69.7	70.6
	DAQ (Kim, Lee, and Ham 2021)	69.9	66.9	69.6	70.5
	LSQ+ (Bhalgat et al. 2020)	70.1	66.7	69.4	70.7
	QFD	70.5	**67.6/87.8**	**70.3/89.4**	**71.1/89.8**
ResNet-34	Auxi (Zhuang et al. 2020)+DoReFa	-	71.2/89.8		
	SPEQ (Boo et al. 2021)	73.6	71.5		
	QKD (Kim et al. 2019)(teacher R50)	73.5	71.6/90.3	73.9/91.4	74.6/92.1
	QIL (Jung et al. 2019)	73.7	70.6	73.1	73.7
	DAQ (Kim, Lee, and Ham 2021)	73.3	71.0	73.1	73.7
	EWGS (Lee, Kim, and Ham 2021)	73.3	71.4	73.3	73.9
	QFD	73.3	**71.7/90.4**	**73.9/91.7**	**74.7/92.3**
MobileNetV2	QKD (Kim et al. 2019)	71.8	45.7/68.1	62.6/84.0	67.4/87.0
	LSQ+KURE (Han et al. 2021)		37.0/62.0	65.9/86.8	69.7/89.2
	LSQ* (Esser et al. 2020)		46.7/71.4	65.3/86.3	69.5/89.2
	QFD		**52.8/77.1**	**66.4/87.0**	**70.5/89.5**

Table 4: Comparing with state-of-the-art methods on ImageNet. SPEQ, QKD and Auxi all adopt KD in their quantization training. "W/A" in the first row represents the bit width of weights and activations. For the proposed QFD method, we report the top-1 (%) and top-5 (%) accuracy of each result item with '/' to separate them.

distillation quantization methods SPEQ and APRT* by a large margin, and is better than both Feature KD and Logit KD. Note that our QFD has achieved even higher accuracy than the full-precision model under 3-bit (92.64%) and 4-bit (93.07%) settings.

We also validate our QFD on CIFAR100 using ResNet-18 and ResNet-32. Similar to the experiments on CIFAR10, we reproduce the baseline, feature distillation and logit distillation methods. As shown in Table 3, Feature KD and Logit KD are generally better than the Baseline, showing the power of knowledge distillation. Our QFD is better than all of them, especially in extreme low bit scenarios (1-bit and 2-bit). Our method can almost recover the accuracy of a full precision model in 2, 3, 4 bit for ResNet-18 and ResNet-32. Especially for 1-bit ResNet-32, only our QFD shows improvement over the Baseline (56.84% vs 56.09%).

ImageNet results. We compare the proposed QFD with other QAT methods on the ImageNet1k dataset. The results can be found in Table 4. The proposed quantization feature distillation surpasses previous methods (including other knowledge distillation methods SPEQ, QKD and Auxi) with ResNet-18, ResNet-34 and MobileNetV2 models under different bit settings. Note that Auxi uses a manually designed auxiliary module, SPEQ needs empirical exploration of stochastic precision, and QKD involves larger teacher models (e.g., ResNet-50 to distill ResNet-34). In comparison, our method is both *simpler* in concept and *more effective* in accuracy, especially for MobileNetV2, where our QFD surpasses QKD by a large margin (increase by 7.1%, 3.8% and 3.1% under 2-bit, 3-bit and 4-bit quantization settings, respectively). For ResNet series models, our QFD perfectly recovers the full precision's top-1 accuracy under 3- and 4-bit quantization (4-bit ResNet-34 with 74.7% top-1 even surpasses its full precision counterpart by 1.3%). Meanwhile, the accuracy of MobileNetV2 is relatively more difficult to recover under low bit, possibly due to its large channel variance, as pointed out by Nagel et al. (2021). But our QFD is still better than other methods.

CUB200 with ViT. We also quantize vision transformers on the image classification benchmark CUB200 (Wah et al. 2011), which contains 200 categories of birds, with 5,994 and 5,794 images for training and testing, respectively. Specifically, we quantize the linear layer in multi-layer-perceptron (MLP) and multi-head-attention (MHA) to 3- or 4-bit, using different structures of ViT (Dosovitskiy et al. 2021) and Deit (Touvron et al. 2021), including ViT Small, ViT Base, Deit Tiny, Deit Small and Deit Base. We also list the accuracy of the teacher network with quantized features (a preprocessing step of our QFD method). As Table 5 shows, although quantizing only the feature brings a slight accuracy drop to the original FP model, the improvement of QFD method over Baseline is significant and consistent. But, there is still a gap between 4-bit and the FP models. Quantizing transformer still remains a challenging task.

Object detection results

RetinaNet. RetinaNet (Lin et al. 2017b) is a one-stage object detector composed of the backbone, FPN (Lin et al. 2017a) and detection head. On the MS-COCO dataset (Lin et al. 2014), we quantize all its layers to 4-bit and 3-bit using the proposed QFD method (including the convolution operation in the skip connection) except the input to the backbone and the output in the detection head. Following previous work (Yamamoto 2021; Zhuang et al. 2020), our quantization training is finetuned using the full precision model ('FP' in Table 6).

For the teacher network, we first quantize its output feature at the $p3$ level to 8-bit since it contains the most gradient flow in the FPN graph (Lin et al. 2017b), then use it as the

Methods	Bit (W/A)	Vit Small	ViT Base	Deit Tiny	Deit Small	Deit Base
Full precision	32/32	82.64	89.44	73.58	81.29	84.14
Feature 4-bit	32/32	82.19	88.89	72.04	79.84	83.95
Feature 8-bit	32/32	82.21	88.71	72.28	80.00	83.70
Baseline	3/3	74.40	83.15	61.11	64.77	69.24
QFD	3/3	**77.01**	**84.28**	**62.24**	**67.05**	**70.68**
Baseline	4/4	78.44	86.74	69.36	74.56	78.68
QFD	4/4	**81.15**	**87.42**	**69.68**	**76.44**	**81.67**

Table 5: Vision transformers on CUB200. "Feature k-bit" means the performance of full precision (FP) teacher with its feature quantized to k-bit. Here we utilize the teacher with 4-bit feature ("Feature 4-bit") to distill the student network.

Arch	Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
R18	FP	33.4	52.3	35.7	19.3	36.2	44.0
	Auxi	31.9	50.4	33.7	16.5	34.6	42.3
	LCQ	32.7	51.7	34.2	18.6	35.2	42.3
	QFD	**33.7**	**52.4**	**35.6**	**19.7**	**36.3**	**44.5**
R34	FP	37.1	56.7	39.6	22.7	41.0	47.6
	Auxi	34.7	53.7	36.9	19.3	38.0	45.9
	LCQ	36.4	55.9	38.7	21.2	40.0	46.6
	Ours	**37.0**	**56.4**	**39.4**	**22.8**	**40.5**	**48.1**
R50	FP	38.6	58.3	41.5	24.1	42.2	49.7
	Auxi	36.1	55.8	38.9	21.2	39.9	46.3
	LCQ	37.1	57.0	39.6	21.2	40.8	47.1
	QFD	**38.2**	**57.5**	**40.9**	**23.1**	**41.4**	**49.3**

Table 6: RetinaNet detector with ResNet-18, 34 and 50 backbones under 4-bit setting on MS-COCO. Note that LCQ (Yamamoto 2021) adopt a more complex quantizer (non-uniform quantizer). 'FP' stands for full precision.

quantized feature to distill a student RetinaNet. Empirically, we find that utilizing the quantized feature of all the FPN level (including $p3$, $p4$, $p5$, $p6$, $p7$), the common approach in object detection distillation (Guo et al. 2021), achieves similar accuracy but is unstable. For simplicity, we only use $p3$ for feature distillation and do *not* involve any complex operation like distinguishing foreground and background features (Yang et al. 2022; Guo et al. 2021). The quantized feature distillation loss occupy about 1/5 of the total detection loss, and the RetinaNet structure strictly follows previous quantization work (Yamamoto 2021).

Table 6 shows the result of quantizing RetinaNet to 4-bit. Our QFD (ResNet18/34/50 as backbone) surpasses previous methods by a large margin. Especially for ResNet-18, our QFD even surpasses its full precision counterpart (improvement of 0.3%, 0.4% and 0.5% for AP, AP_S and AP_L, respectively). Our accuracy drop from the full precision ones with ResNet-34 is negligible as well, with a slight decrease of 0.1% on AP and 0.2% on AP_{75}.

Table 7 shows the result of quantizing RetinaNet to 3-bit. Unlike 4-bit quantization, 3-bit is more challenging and difficult to optimize due to its limited representation power. Empirically we find ResNet-34 often face unstable training issue, such that we elongate its warmup iterations while keeping the total training iterations fixed. Overall, our QFD outperforms previous state-of-the-art method by a relatively large margin, especially for ResNet-18 where the improvement of AP_M and AP_L over LCQ (Yamamoto 2021) is

Arch	Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
R18	FP	33.4	52.3	35.7	19.3	36.2	44.0
	APoT	31.2	50.1	32.8	18.0	33.5	40.6
	LCQ	31.3	50.2	33.1	17.6	33.8	40.4
	QFD	**32.0**	**50.3**	**33.9**	**18.0**	**34.5**	**42.6**
R34	FP	37.1	56.7	39.6	22.7	41.0	47.6
	APoT	35.2	54.9	37.1	19.7	**39.1**	45.3
	QFD	**35.6**	**54.9**	**37.9**	**21.6**	39.0	**45.6**
R50	FP	38.6	58.3	41.5	24.1	42.2	49.7
	LCQ	36.1	**56.2**	38.4	21.7	**39.9**	46.1
	QFD	**36.5**	56.1	**39.0**	**22.1**	39.5	**47.2**

Table 7: RetinaNet detector with ResNet-18, 34 and 50 backbones under 3-bit setting on MS-COCO. APoT (Li, Dong, and Wang 2020) and LCQ both use non-uniform quantizers while our QFD uses the simple uniform one.

0.7% and 2.2%, respectively.

The ViT structure. Lastly, we explore quantizing ViT. To the best of our knowledge, this is the first time that ViT has been quantized in detection and segmentation tasks. We tried ViT (Dosovitskiy et al. 2021) and Swin Transformer (Liu et al. 2021a) pretrained on ImageNet1k and ImageNet21k, respectively, using the self-supervised learning methods MAE (He et al. 2022). The detection pipeline follows the newly published ViTDet (Li et al. 2022). Since most parameters are in the linear layer of MLP and MHA in the backbone transformer blocks, we quantize only the linear layers in the backbone and run the baseline quantization (without QFD distillation) under 8, 6, 4 bit settings.

As shown in Table 8, 8-bit or 6-bit quantization for linear layer of ViT and Swin Transformer ('SwinB' means the base structure of it) is roughly enough to recover its detection and segmentation accuracy, demonstrating the potential to deploy vision transformers on real-world hardware devices (Li et al. 2021). On the contrary, quantizing vision transformers to 4-bit leads to noticeable performance drop, possibly due to its limited representation capacity. We further analyze the impact of MHA and MLP by alternatively quantizing each only, and the results in Table 8 convey an interesting observation: quantizing vision transformer in detection and segmentation is *not at all sensitive to the attention layer, but to the linear layers in the MLP*. Note that the performance of $4/4^a$ in ViT and SwinB even surpassed its 8/8 counterpart. It is possible because the MLP layer is seriously affected by the inter-channel variation in LayerNorm inputs (Lin et al. 2022; Ba, Kiros, and Hinton 2016), while the MHA layer

Arch	Bit	Detection			Segmentation		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
ViTDet	FP	50.8	71.6	56.3	45.3	69.2	49.3
	8/8	50.1	70.9	54.9	44.5	68.2	48.3
	6/6	49.8	70.7	54.7	44.3	68.0	48.2
	4/4	46.7	67.5	51.0	41.4	64.4	44.4
	$4/4^a$	**50.3**	**71.0**	**55.2**	**44.5**	**68.3**	**48.1**
	$4/4^m$	47.6	68.4	52.1	42.3	65.6	45.5
SwinB	FP	53.6	72.6	58.4	46.1	70.1	50.2
	8/8	52.9	71.8	57.9	45.5	69.0	49.6
	6/6	52.7	71.7	57.3	45.4	68.9	49.5
	4/4	51.6	70.5	56.2	44.4	67.8	47.9
	$4/4^a$	**53.2**	**72.1**	**58.1**	**45.7**	**69.5**	**49.6**
	$4/4^m$	52.3	71.2	57.1	45.0	68.4	49.1

Table 8: Results of ViT and Swin Transformer quantization on MS-COCO detection and segmentation tasks. $4/4^a$ means quantizing multi-head-attention (MHA) layer only, while $4/4^m$ quantizes MLP layer only. All the other bit settings quantize both MHA and MLP.

λ	bit	top-1 acc	top-5 acc
Full-precision	32/32	71.8	90.2
Baseline	2/2	47.1	72.1
0.1	2/2	50.8	75.4
0.3	2/2	**52.8**	**77.1**
0.5	2/2	49.1	74.5

Table 9: Results of different distillation hyper-parameter λ of 2-bit MobileNetV2 on ImageNet. 'acc' means accuracy.

contains additional operations which might mitigate this effect.

Ablation studies

Effect of λ & Quantizing feature. We verify the only hyper-parameters λ defined in Eq. 8 on ImageNet and CUB under different bit settings (2-bit, 3-bit and 4-bit). The result can be found in Tables 9 to 11. For both CNNs and vision transformers on various datasets, all the λ values lead to improvements over the baseline quantization method and our method is not sensitive to the value of λ. Interestingly, in Table 11 (4-bit ResNet-34 quantization on ImageNet), our QFD further increases the accuracy by 1.1% top-1 accuracy even when the baseline quantization method has already surpassed its full precision counterpart. Hence, we can choose $\lambda = 0.5$ by default for simplicity.

Effect of quantizing the teacher's feature. Meanwhile, we show the accuracy of the teacher network (with its feature quantized). As shown in Table 12, the feature-quantized teacher networks almost have no difference with the original in terms of accuracy. Hence, the accuracy improvement is brought by QFD, not because the teacher's accuracy is higher than that of the baseline.

λ	bit	best acc	last acc
Full-precision	32/32	82.64	82.38
Baseline	3/3	74.40	74.40
0.1	3/3	75.66	74.78
0.5	3/3	**77.01**	**76.77**
0.9	3/3	76.39	76.08

Table 10: Results of different distillation hyper-parameter λ with 3-bit ViT Small on CUB200. 'acc' stands for accuracy.

λ	bit	top-1	top-5
Full-precision	32/32	73.3	91.4
Baseline	4/4	73.6	91.5
0.1	4/4	73.9	91.6
0.5	4/4	**74.7**	**92.3**
0.7	4/4	74.2	91.9

Table 11: Results of different distillation hyper-parameter λ of 4-bit ResNet34 on ImageNet. 'acc' stands for accuracy.

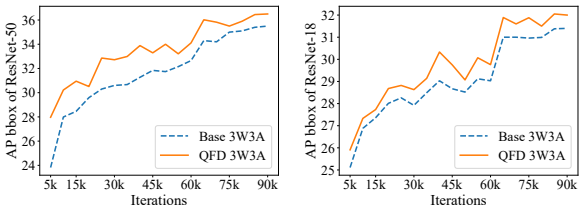
	Full precision acc	Teacher acc
ResNet-18	70.5	70.9
ResNet-34	73.3	73.3
MobileNetV2	71.8	71.1

Table 12: The teacher's accuracy with its feature quantized to 4-bit on ImageNet classification. Quantizing feature to low bit has similar accuracy as that of the original FP model.

Consistent improvement of detection. Finally in this section, we plot the convergence curve of RetinaNet using either the baseline quantization or our QFD distillation on MS-COCO detection tasks. The results using ResNet-18 and ResNet-50 backbones under 3-bit quantization can be found in Figure 4. There is no doubt that QFD makes consistent improvement over the baseline throughout the whole training process, demonstrating the generality of our methods: it is not only suitable for classification, but also boosts object detection performance as well.

Conclusions

In this paper, we proposed a novel and easy to implement feature distillation method QFD in quantization aware training. We first qualitatively illustrated QFD's advantages: simple, quantization friendly, and flexible. Extensive experiments on image classification, object detection and segmentation benchmarks with both convolutional networks (ResNet and MobileNetV2) and vision transformers were consistently better than previous state-of-the-art quantization aware training methods.



(a) ResNet-50 backbone (b) ResNet-18 backbone

Figure 4: Comparison of QFD with baseline using 3 bit RetinaNet (ResNet backbone) on MS-COCO detection tasks.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China under Grant 62276123 and Grant 61921006.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. arXiv:1607.06450.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv:1308.3432.

Bhalgat, Y.; Lee, J.; Nagel, M.; Blankevoort, T.; and Kwak, N. 2020. LSQ+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 696–697.

Boo, Y.; Shin, S.; Choi, J.; and Sung, W. 2021. Stochastic precision ensemble: Self-knowledge distillation for quantized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6794–6802.

Cai, Z.; He, X.; Sun, J.; and Vasconcelos, N. 2017. Deep learning with low precision by half-wave gaussian quantization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5918–5926.

Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P. I.-J.; Srinivasan, V.; and Gopalakrishnan, K. 2018. PACT: Parameterized clipping activation for quantized neural networks. arXiv:1805.06085.

Courbariaux, M.; Bengio, Y.; and David, J.-P. 2015. BinaryConnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, 3123–3131.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*.

Esser, S. K.; McKinstry, J. L.; Bablani, D.; Appuswamy, R.; and Modha, D. S. 2020. Learned step size quantization. In *Proceedings of the International Conference on Learning Representations*.

Gong, R.; Liu, X.; Jiang, S.; Li, T.; Hu, P.; Lin, J.; Yu, F.; and Yan, J. 2019. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4852–4861.

Guerra, L.; Zhuang, B.; Reid, I.; and Drummond, T. 2020. Switchable precision neural networks. arXiv:2002.02815.

Guo, J.; Han, K.; Wang, Y.; Wu, H.; Chen, X.; Xu, C.; and Xu, C. 2021. Distilling object detectors via decoupled features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2154–2164.

Han, T.; Li, D.; Liu, J.; Tian, L.; and Shan, Y. 2021. Improving low-precision network quantization via bin regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5261–5270.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

He, Y.-Y.; Wu, J.; and Wei, X.-S. 2021. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 235–244.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531.

Jung, S.; Son, C.; Lee, S.; Son, J.; Han, J.-J.; Kwak, Y.; Hwang, S. J.; and Choi, C. 2019. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4350–4359.

Kim, D.; Lee, J.; and Ham, B. 2021. Distance-aware quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5271–5280.

Kim, J.; Bhalgat, Y.; Lee, J.; Patel, C.; and Kwak, N. 2019. QKD: Quantization-aware knowledge distillation. arXiv:1911.12491.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Lee, J.; Kim, D.; and Ham, B. 2021. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6448–6457.

Lee, J. H.; Yun, J.; Hwang, S. J.; and Yang, E. 2021. Cluster-promoting quantization with bit-drop for minimizing network quantization loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5370–5379.

Li, Y.; Dong, X.; and Wang, W. 2020. Additive Powers-of-Two Quantization: An Efficient Non-uniform Discretization for Neural Networks. In *Proceedings of the International Conference on Learning Representations*.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. arXiv:2203.16527.

Li, Y.; Shen, M.; Ma, J.; Ren, Y.; Zhao, M.; Zhang, Q.; Gong, R.; Yu, F.; and Yan, J. 2021. MQBench: Towards Reproducible and Deployable Model Quantization Benchmark. In *In Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, volume 8693 of *Lecture Notes in Computer Science*, 740–755. Springer.

Lin, Y.; Zhang, T.; Sun, P.; Li, Z.; and Zhou, S. 2022. FQ-ViT: Post-Training Quantization for Fully Quantized Vision Transformer. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1173–1179.

Liu, Z.; Cheng, K.-T.; Huang, D.; Xing, E. P.; and Shen, Z. 2022. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4942–4952.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021b. Post-training quantization for vision transformer. In *Advances in Neural Information Processing Systems*, 28092–28103.

Mishra, A.; and Marr, D. 2018. Apprentice: Using Knowledge Distillation Techniques To Improve Low-Precision Network Accuracy. In *Proceedings of the International Conference on Learning Representations*.

Nagel, M.; Fournarakis, M.; Amjad, R. A.; Bondarenko, Y.; van Baalen, M.; and Blankevoort, T. 2021. A white paper on neural network quantization. arXiv:2106.08295.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 8024–8035.

Polino, A.; Pascanu, R.; and Alistarh, D. 2018. Model compression via distillation and quantization. In *Proceedings of the International Conference on Learning Representations*.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, volume 9908 of *Lecture Notes in Computer Science*, 525–542. Springer.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.; and Fei-Fei, L. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Sun, M.; Ma, H.; Kang, G.; Jiang, Y.; Chen, T.; Ma, X.; Wang, Z.; and Wang, Y. 2022. VAQF: Fully automatic software-hardware co-design framework for low-bit vision transformer. arXiv:2201.06618.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wu, J.; and Luo, J.-H. 2018. Learning effective binary visual representations with deep networks. arXiv:1803.03004.

Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; and Girshick, R. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Yamamoto, K. 2021. Learnable companding quantization for accurate low-bit neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5029–5038.

Yang, Z.; Li, Z.; Jiang, X.; Gong, Y.; Yuan, Z.; Zhao, D.; and Yuan, C. 2022. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4643–4652.

Yuan, Z.; Xue, C.; Chen, Y.; Wu, Q.; and Sun, G. 2021. PTQ4ViT: Post-training quantization framework for vision transformers. arXiv:2111.12293.

Zhang, D.; Yang, J.; Ye, D.; and Hua, G. 2018. LQ-Nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European Conference on Computer Vision*, volume 11212 of *Lecture Notes in Computer Science*, 365–382. Springer.

Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; and Zou, Y. 2016. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv:1606.06160.

Zhuang, B.; Liu, L.; Tan, M.; Shen, C.; and Reid, I. 2020. Training quantized neural networks with a full-precision auxiliary module. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1488–1497.