Is In-Domain Data Really Needed? A Pilot Study on Cross-Domain Calibration for Network Quantization

Haichao Yu[1], Linjie Yang[2], Humphrey Shi[1]
[1]University of Illinois at Urbana-Champaign, [2]ByteDance Inc.
haichao3@illinois.edu, linjie.yang@bytedance.com, shihonghui3@gmail.com

Abstract

Post-training quantization methods use a set of calibration data to compute quantization ranges for network parameters and activations. The calibration data usually comes from the training dataset which could be inaccessible due to sensitivity of the data. In this work, we want to study such a problem: ***can we use out-of-domain data to calibrate the trained networks without knowledge of the original dataset?*** *Specifically, we go beyond the domain of natural images to include drastically different domains such as X-ray images, satellite images and ultrasound images. We find cross-domain calibration leads to surprisingly stable performance of quantized models on 10 tasks in different image domains with 13 different calibration datasets. We also find that the performance of quantized models is correlated with the similarity of the Gram matrices between the source and calibration domains, which can be used as a criterion to choose calibration set for better performance. We believe our research opens the door to borrow cross-domain knowledge for network quantization and compression.*

1. Introduction

With the increasing popularity of deploying neural networks on edge devices, neural network quantization has become a widely studied topic [6, 32, 27, 43, 20, 42, 39, 15, 2, 41]. By quantizing its weights and activations to low-bit integers, a neural network can be stored with smaller size and executed at faster speed with less memory footprint and computational resources.

Most existing network quantization methods can be roughly divided into two groups. One is quantization-aware training (QAT), the other is post-training quantization (PQ). By inserting differentiable simulated quantization operations into the network during training, QAT methods allow training losses to back-propagate through the quantization operations. Although QAT methods can achieve satisfactory performance in most cases, it is time-consuming and data-hungry, which is unacceptable in some scenarios, *e.g.*, when the time budget is restricted or training dataset is inaccessible. Compare with QAT, PQ has advantages on time and data efficiency. Given a pretrained full-precision model, PQ runs the model over a small set of calibration data to calculate value ranges of intermediate activations. These calibration data shares the same distribution as the training data on which the model is trained. After this, both network activations and weights can be quantized into low-precision integers for inference acceleration. Compared with QAT, this post-training quantization process is highly efficient (*e.g.*, in a few minutes).

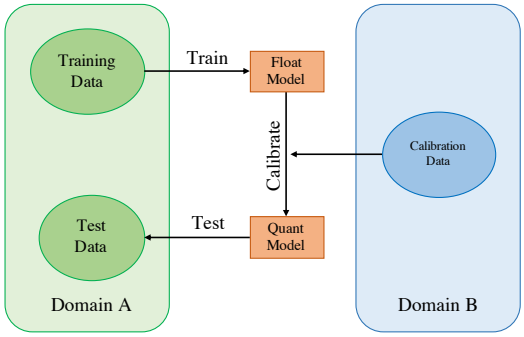


Figure 1: Pipeline of cross-domain calibration for post-training quantization without using in-domain data.

However, PQ methods still require in-domain calibration data, which may be inaccessible in some situations due to privacy or security reasons. For example, federated learning [16] trains models using distributed private user data that are not accessible by the developers. Without calibration data from the same domain as the training data, the traditional PQ methods will fail. Targeting at this problem, Cai *et al*. proposed a novel method termed ZeroQ that generates synthetic data for calibration [3]. Specifically, ZeroQ utilized BatchNorm [13] statistics from the pretrained model as supervision to optimize randomly initialized input such that the BatchNorm statistics of the input are similar to those from the pretrained model. ZeroQ showed its

effectiveness on various datasets including ImageNet [7] and MSCOCO [23]. Another method is to use Generative Adversarial Networks (GAN) to create fake calibration data [37]. The proposed framework named GDFQ is trained with BatchNorm statistics loss, Cross Entropy classification loss and knowledge distillation loss.

One drawback of these data-free quantization methods is their high time complexity. With common hardware setting (*e.g.*, on single GPU), ZeroQ requires hundreds of back-propagation iterations to synthesize a single batch and GDFQ needs to be trained for hundreds of epochs. On the other hand, due to the optimization nature of these methods, hyper-parameters also need to be tuned for satisfactory results. If in-domain training data is inaccessible and synthetic data generation is time-consuming, a straightforward question to ask is "Can we use real-world images from another domain for calibration?". As shown in Figure 1, we use calibration data from domain B to calibration model trained on domain A, resulting in a quantized model. The model is further evaluated on a test set in domain A.

To study this problem, we carry out a large-scale benchmark with 10 different tasks and 13 diversified image domains including nature images, low-resolution images, ultrasound images, satellite images, etc. We find that a simple treatment on the BatchNorm layers in the calibration procedure greatly improves performance of cross-domain calibration, almost bridging the gap between in-domain and out-of-domain calibration on a wide range of tasks. We also find that the performance of quantized models is correlated with the similarity of the Gram matrices between the source and calibration domains, which can be used as a criterion to choose similar image domains for better performance. Compared with synthetic data generation method ZeroQ, our approach achieves comparable or better performance on our large-scale benchmark. Although the study is mainly empirical, it reveals that cross-domain data can be used for post-training calibration just as effective as in-domain data, which could motivate the community to explore more in the direction of model quantization and compression with cross-domain knowledge.

2. Related Work

2.1. Quantization-Aware Training

Quantization-aware training (QAT) inserts simulated quantization operation into model forward and backward passes during training. First introduced by Courbariaux *et al*. [12], binary neural networks achieved seven times faster inference speed on GPU. Further in [32], Rastegari *et al*. proposed to use a binary tensor and a float scalar to approximate the weight or activation tensor. Parallel to binarization, multi-bit model quantization is also widely explored. DoReFa-Net [42] was proposed to quantize both network and training gradients. Scale-adjusted training [14] improve the model performance by scaling activation in the network to preserve proper scale of gradients. To bring flexibility to the model quantizer, some works also introduced learnable quantization schemes. In [39], LQ-Net was proposed to learn the quantization basis vectors. In [15], Jung *et al*. proposed a novel method to learn the quantization intervals.

2.2. Post-training Quantization

Different from QAT, post-training quantization (PQ) directly computes quantization parameters with a calibration set. In [17], Krishnamoorthi *et al*. introduced PQ with layer-wise and channel-wise quantization schemes. To improve quantization performance, Banner *et al*. proposed a novel per-channel PQ scheme that analytically computes clipping thresholds and bits allocated for each channel [2]. To mitigate the quantization error from the clipping operation, Zhao *et al*. introduced a technique named OCS to split the channels with extreme values [41].

Traditionally, PQ method requires a calibration dataset which is often sampled from training data to estimate the clipping ranges of model activations. However, in some scenarios, training data may be inaccessible due to data privacy issues. In this case, quantization could be conducted without a calibration set. Assuming features from a BatchNorm layer follow Gaussian distribution, Nagel *et al*. directly used BatchNorm statistics (*i.e.*, running means and variances) to estimate activation ranges [29]. The drawback of this method is the estimated ranges may be inaccurate since the distribution of intermediate layers may be different from Gaussian distribution. DFC and ZeroQ proposed to synthesize calibration data by gradient descent under the supervision of BatchNorm statistics [10, 3]. Generative Adversarial Networks (GAN) is also utilized to generate synthetic calibration data [37], in which the generator generates calibration data and the discriminator is the quantized model. One problem with these optimization-based methods is the high time complexity. Hundreds of iterations or even hundreds of epochs of back-propagation are commonly required. In addition, much human efforts are needed for hyper-parameter tuning such as learning rate and optimizer selection. In contrast to these methods, we find that out-of-domain image datasets can serve as calibration data effectively for a wide range of tasks.

2.3. Batch Normalization and Domain Knowledge

To adapt a deep neural network to a different domain, Li *et al*. proposed Adaptive BatchNorm to update running statistics in target domain [22]. The central assumption is that domain specific information is encoded in BatchNorm layers. Sharing a similar spirit with this, Li *et al*. showed that style transfer can be conducted by matching the BatchNorm statistics between two images [21]. Inspired by

the effect of BatchNorm statistics, we design a BatchNorm updating method on out-of-domain calibration dataset to estimate ranges of activations, which effectively improves the performance of cross-domain calibration even when the training and the calibration domains have huge appearance differences.

3. Cross-domain Calibration for Post-training Quantization

3.1. Motivation

Given a pretrained full-precision model, post-training quantization (PQ) is widely used for inference acceleration. Most existing PQ methods require a set of calibration data to calculate the quantization parameters for weights and activations. Training data is normally used for calibration. However, in some situations, the training data is not available due to privacy or security issues. To solve this problem, Cai *et al*. [3] proposed to synthesize calibration data using BatchNorm statistics as supervision. Since different types of real images are vastly available, we would like to explore another option: use out-of-domain real images for calibration. We conduct a large-scale empirical study with drastically different datasets and tasks to investigate this setting.

3.2. Quantization Schemes

In all our experiments, we employ layer-wise uniform quantization for both network weights and activations [17]. To quantize a tensor $\boldsymbol{x}$ by k-bit, the quantized tensor $\boldsymbol{x}_q$ is calculated as

$$\begin{aligned} \boldsymbol{x}_{int} &= \text{round}\left(\frac{\boldsymbol{x}}{s}\right) + z, \\ \boldsymbol{x}_q &= \text{clip}\left(\boldsymbol{x}_{int}, c_l, c_h\right), \end{aligned} \tag{1}$$

where s and z are scale and zero point parameters, c_l and c_h are lower and upper clipping thresholds. For network weight $\boldsymbol{w}$, we use a symmetric min-max quantization scheme, which is defined as

$$\begin{aligned} s_w &= \frac{\max(|w_{min}|, |w_{max}|)}{2^{k-1}}, \\ z_w &= 0, \\ c_l &= -2^{k-1}, \\ c_h &= 2^{k-1} - 1, \end{aligned} \tag{2}$$

where w_{min} and w_{max} are minimum and maximum of $\boldsymbol{w}$. To quantize activation $\boldsymbol{a}$, we employ an affine histogram quantization scheme, which is also used in PyTorch[1]. To determine the quantization parameters, we have

$$\begin{aligned} s_a &= \frac{a_h - a_l}{2^k - 1}, \\ z_a &= \frac{a_l(2^k - 1)}{a_l - a_h}, \\ c_l &= 0, \\ c_h &= 2^k - 1, \end{aligned} \tag{3}$$

where a_l and a_h are lower and upper clipping thresholds for $\boldsymbol{a}$. For each layer, a histogram is built to represent the activation distribution. We search for the optimal a_l and a_h values such that the quantization error is minimized with respect to $\boldsymbol{a}$. In all our experiments, we first fold BatchNorm layers to their preceding convolutional or linear layers before computing quantization parameters.

3.3. Datasets and Networks

Our experiments span across different image domains including natural images [19, 7, 8, 5, 23, 18], X-ray [36], ultrasound [1], and satellite images [4]. We also experiment with models in various computer vision tasks including image classification, semantic segmentation and object detection. The datasets and corresponding example images are illustrated in Figure 2. We train floating-point models on all the datasets from Figure 2a to Figure 2j, where each dataset has one or more associated models. These datasets and three additional datasets from Figure 2k to Figure 2m are used as calibration sets. When calibration and training data have different sizes, we resize the calibration data to the same size of the training data. One exception is when calibrating ResNet-18 on Cifar100, calibration samples are randomly cropped. In addition, when Agriculture-Vision data is used to calibrate models on other datasets, only RGB channels are used. When other datasets are used to calibrate Agriculture-Vision models, we use RGB channels to compute a grayscale channel to be used as NIR channel.

These models to be quantized are summarized in Table 1. The evaluation metrics are summarized as below:

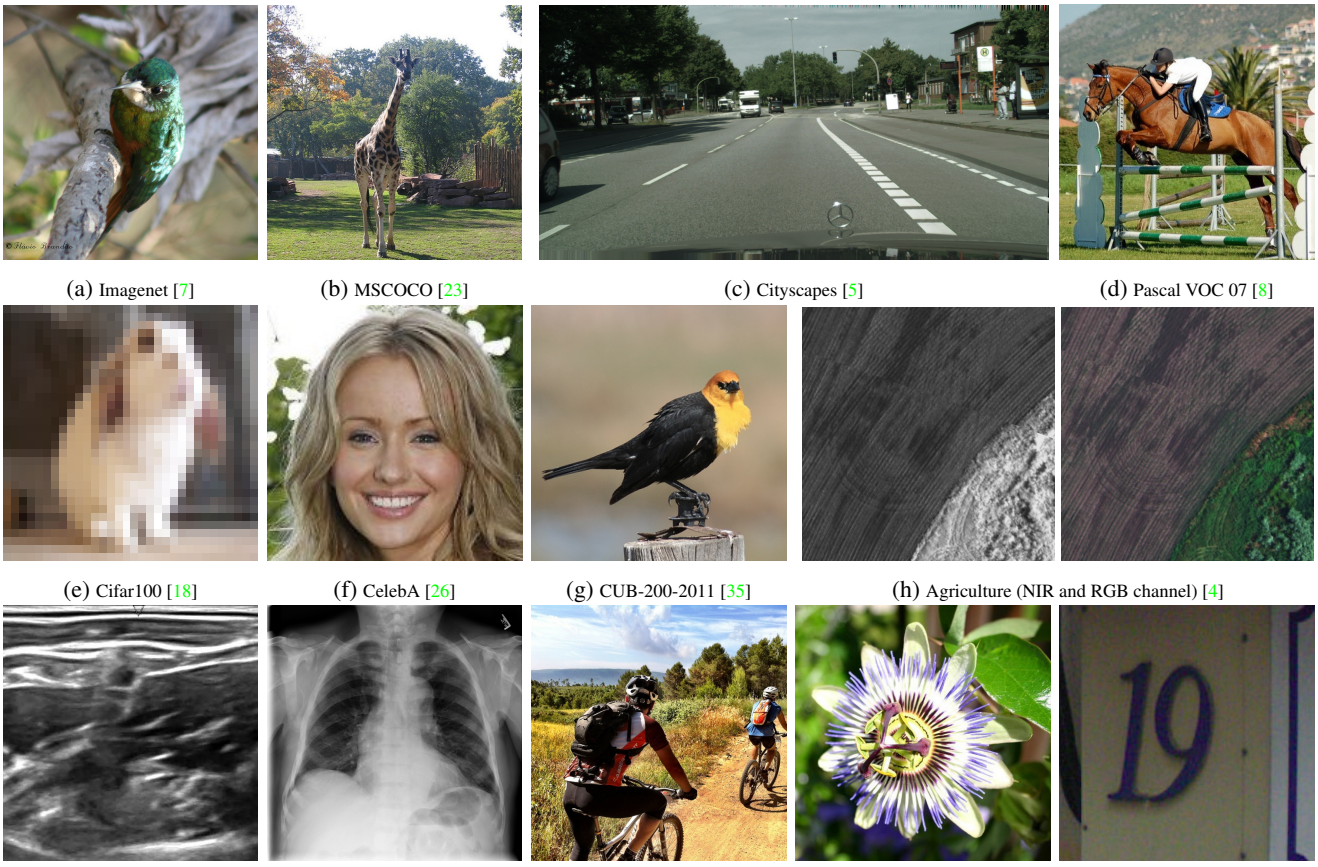
1. All classification tasks except CelebA: top-1 accuracy.
2. CelebA: average top-1 accuracy on 40 attributes.
3. Pascal VOC 2007: mAP.
4. MSCOCO, Cityscapes and Agriculture-Vision: mIoU.
5. Ultrasound: DICE.

Specifically, DICE is a widely used evaluation metric for binary segmentation, which is defined as

$$Dice = \frac{2|\boldsymbol{p} \cap \boldsymbol{g}|}{|\boldsymbol{p}| + |\boldsymbol{g}|}, \tag{4}$$

where $\boldsymbol{p}$ and $\boldsymbol{g}$ are binary prediction and ground truth tensors respectively.

[1] https://pytorch.org/docs/master/torch.quantization.html



(a) Imagenet [7] (b) MSCOCO [23] (c) Cityscapes [5] (d) Pascal VOC 07 [8]

(e) Cifar100 [18] (f) CelebA [26] (g) CUB-200-2011 [35] (h) Agriculture (NIR and RGB channel) [4]

(i) Kaggle Ultrasound [1] (j) NIH Chest X-ray [36] (k) Open Images [19] (l) Oxford Flowers [31] (m) SVHN [30]

Figure 2: Example images from different domains. We will use abbreviations in the following section: IN: ImageNet, CO: MSCOCO, CS: Cityscapes, VO: Pascal VOC 2007 CI: Cifar100, CE: CelebA, CB: CUB-200-2011, AG: Agriculture-Vision, US: Kaggle Ultrasound, NI: NIH Chest X-ray. OI: Open Images, OF: Oxford Flowers-101, SV: SVHN.

Datasets	Tasks	Models
Imagenet	C	ResNet-18, ResNet-50 [11] Inception-V3 [34]
MSCOCO	S	FCN [28]
Cityscapes	S	BiSeNet [38]
Pascal VOC 2007	D	MobileNetV2 SSD-Lite [33] [25]
Cifar100	C	ResNet-18
CelebA	C	ResNet-50
CUB-200-2011	C	MMAL-Net [40]
Agriculture-Vision	S	MSCG-Net-101 [24]
Ultrasound	S	U-net
NIH Chest X-ray	C	ResNet-50

Table 1: Training datasets and the corresponding pretrained models. C: Image Classification. S: Semantic Segmentation. D: Object Detection.

3.4. Cross-domain Calibration

A Naive Approach First we employ a naive cross-domain calibration approach, *i.e.*, directly using out-of-domain data to calibrate a model trained in another domain. The results of 8-bit quantization are plotted as blue bars in Figure 3. The in-domain calibration results are marked in dashed lines.

We can see that for all the datasets and models, a large number of out-of-domain calibration results are comparable to the in-domain baselines. Take ResNet-50 on ImageNet as an example, the gap of out-of-domain and in-domain calibration results is very small. This observation is surprising since most existing PQ methods assume in-domain calibration data is required to estimate activation ranges of intermediate layers. For some models such as MobileNetV2 SSD-Lite on Pascal VOC, ResNet-50 on CelebA, U-net on Ultrasound, and ResNet-50 on NIH Chest X-ray, there is still an accuracy gap between out-of-domain calibration results and the baseline in-domain results. Such discrepancy often occurs when the original and the calibration domain have a large gap, *e.g.* most natural image datasets are drastically different from ultrasound images. Motivated by this observation, we introduce an approach to reduce the representation gap between the original and the calibration domains for the target model, namely BatchNorm adjustment.

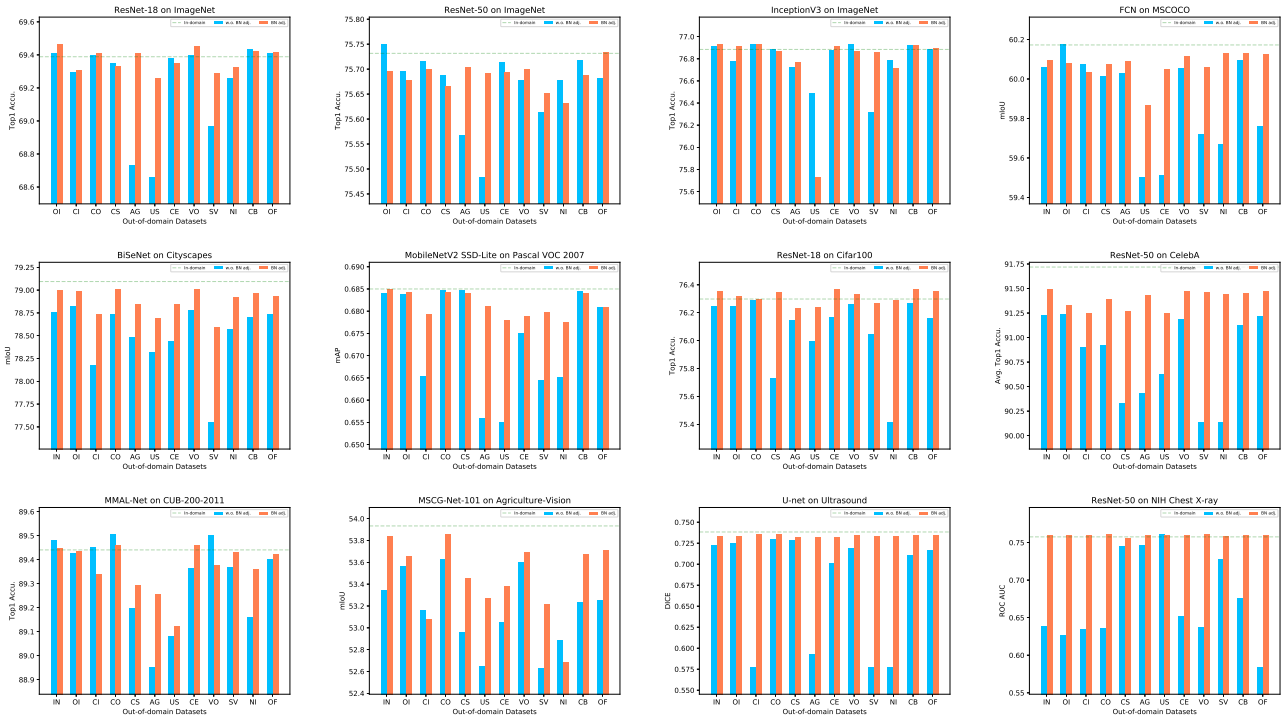


Figure 3: Cross-domain calibration with and without BatchNorm adjustment. Dashed green line denotes performance of in-domain calibration. The dataset abbreviations are defined in Figure 2. Zoom in for details.

BatchNorm Adjustment Since BatchNorm was proposed to reduce Internal Covariate Shift [13], it has been widely applied to deep neural networks. For one neural network, different domains will generate different BatchNorm parameters. One interesting application is that BatchNorm parameters can be updated on a new domain for domain adaptation [22, 21]. As stated by these work, BatchNorm layers encode information that is specific to the domain that the model is trained on. Inspired by this, we propose BatchNorm adjustment to adapt the models to the out-of-domain calibration datasets. We describe our method as below:

1. Given a model M to be quantized and an out-of-domain dataset D.
2. Reset running means and variances of all BatchNorm layers in M: $\mu \leftarrow 0$, $\sigma \leftarrow 1$.
3. Run M on D to accumulate new BatchNorm statistics.
4. Fold BatchNorm layers to their preceding convolutional or linear layers.
5. Calculate weight quantization parameters (s_w and z_w in Equation 2).
6. Run M on D again to calculate activation quantization parameters (s_a and z_a in Equation 3).

The results are plotted in orange bars in Figure 3. For most datasets and models, BatchNorm adjustment gets better performance than the baselines without BatchNorm adjustment. In most cases, the results are comparable to or even slightly better than the in-domain calibration results. On ResNet-18/50 on ImageNet, FCN on MSCOCO, ResNet-18 on Cifar100, U-net on Ultrasound and ResNet-50 on NIH Chest X-ray, all out-of-domain calibration results are within 0.3% performance gap from the in-domain calibration results. On some tasks such as MMAL-Net on CUB-200-2011 and MSCG-Net-101 on Agriculture-Vision, some out-of-domain data still performs relatively lower than others. In such cases, randomly chosen calibration datasets cannot guarantee performance similar to in-domain calibration. In order to fix this issue, we investigate how domain discrepancy affects calibration and how to improve the calibration results.

3.5. Influence of Domain Discrepancy on Calibration

In this section, we investigate the relationship between domain discrepancy and cross-domain calibration performance. Since Maximum Mean Discrepancy (MMD) was proposed to measure difference of sample mean in Reproducing Kernel Hilbert Space [9], it has been widely used as a domain discrepancy measure. As proved by Li *et al.* [21],

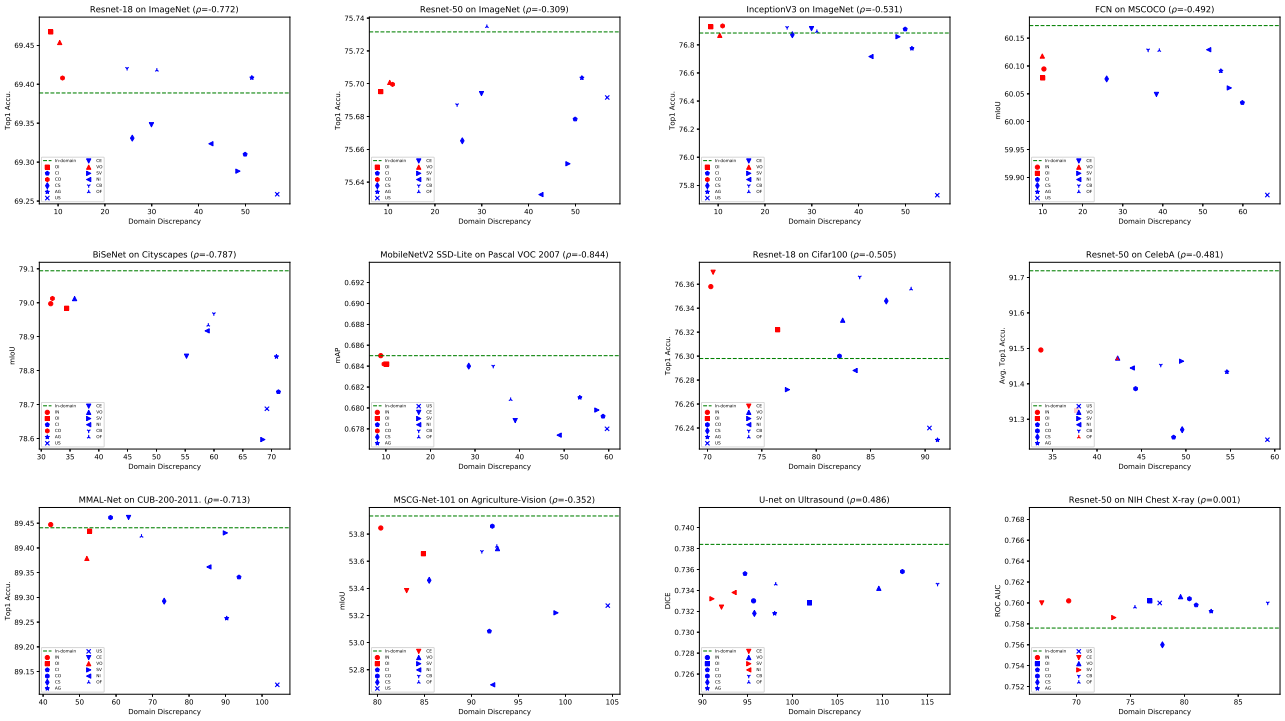


Figure 4: Visualization of correlation between calibration performance and domain discrepancy. Top-3 out-of-domain datasets with smallest domain discrepancy from the in-domain data are marked in red. Dashed green line denotes performance of in-domain calibration. The dataset abbreviations are defined in Figure 2. Zoom in for details.

matching MMD with the second order polynomial kernel is equivalent to matching Gram matrices of feature maps. Similar to [21], we employ mean L_2 distance between Gram matrices of feature maps to measure the discrepancy between two image domains. Formally, domain discrepancy D is defined as

$$D = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \|\boldsymbol{G}_{ij}^A - \boldsymbol{G}_{ij}^B\|_2^2, \tag{5}$$

where $\boldsymbol{G}^A \in \mathbb{R}^{N \times N}$ and $\boldsymbol{G}^B \in \mathbb{R}^{N \times N}$ are average Gram matrices of two domains A and B, which are defined as

$$\boldsymbol{G}^A = \frac{1}{|A|} \sum_{k=1}^{|A|} \boldsymbol{G}_k^A \quad \text{and} \quad \boldsymbol{G}^B = \frac{1}{|B|} \sum_{k=1}^{|B|} \boldsymbol{G}_k^B, \tag{6}$$

where $|A|$ and $|B|$ are number of samples in domain A and B respectively. Ignoring the domain superscript for simplicity, each element $\boldsymbol{G}_{k,ij}$ in $\boldsymbol{G}_k^A$ or $\boldsymbol{G}_k^B$ is

$$\boldsymbol{G}_{k,ij} = \sum_{m=1}^{M} \boldsymbol{F}_{k,im} \boldsymbol{F}_{k,jm}, \tag{7}$$

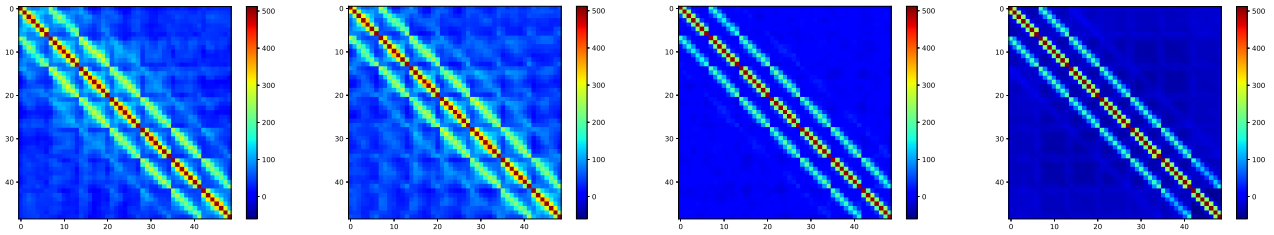
where $\boldsymbol{F}_k$ is the feature embedding of k-th sample, which is normalized by mean and standard deviation over all images in this domain, and M is the embedding dimension. In our experiments, we use the output of last conv layer in the fourth conv block `conv_4_3` of VGG-16 to extract the feature maps.

Next we study how domain discrepancy is related to calibration performance. In Figure 4, we plot model performance vs domain discrepancy with the source training data into scatter points. We also calculate correlation coefficient between model performance and domain discrepancy shown in the title of each sub-figure. First, on most of the datasets, calibration performance is negatively correlated to gram matrix distance. The out-of-domain datasets that have the smaller domain discrepancy with the in-domain data always achieve comparable performance with in-domain data. We marked the top-3 out-of-domain data with smallest domain discrepancy in red for better visualization. This observation can be useful for cross-domain calibration, when calibration/training data is sensitive and inaccessible but some high-level statistics such as Gram matrix are available. One can ask the data owner to provide a mean Gram matrix of the source dataset, which can be used to search in the pre-built candidate pool of cross-domain datasets for the out-of-domain calibration dataset with smallest domain discrepancy.

Using the above strategy, we compare our method with in-domain calibration and data synthesis approach Ze-

Calib Methods \ Datasets	IN			CO	CS	VO	CI	CE	CB	AG	US	NI
	R18	R50	IV3									
FP32	69.76	76.13	77.46	60.47	79.10	0.686	76.50	91.74	89.63	54.63	0.730	0.773
In-domain	69.39	75.73	76.88	60.17	79.09	0.685	76.30	91.72	89.36	53.93	0.738	0.758
ZeroQ	69.36	75.56	76.89	60.17	78.99	0.681	76.01	91.59	87.90	53.34	**0.739**	0.718
ZeroQ-real	69.29	75.57	76.89	**60.18**	78.87	0.681	75.87	**91.62**	**89.29**	53.29	0.738	0.675
Cross-domain	**69.47**	**75.70**	**76.93**	60.12	79.00	**0.685**	76.36	91.50	89.19	**53.85**	0.733	**0.761**
Cross-domain (MS)	**69.47**	**75.70**	**76.93**	60.08	**79.01**	0.684	**76.37**	91.50	89.19	53.38	0.732	**0.761**

Table 2: 8-bit quantization results with different calibration datasets. The dataset abbreviations are defined in Figure 2. R-18, R-50 and IV3 are Resnet-18, Resnet-50 and InceptionV3 respectively. Cross-domain (MS) uses an average of gram matrices from multiple layers. Best results without using in-domain data are emphasized in bold.



(a) ImageNet (b) Open Images (c) Ultrasound (d) ZeroQ

Figure 5: Visualization of Gram matrices of different datasets. The feature extraction model is VGG-16 with BatchNorm trained on ImageNet. In Column (b) and (c), BatchNorm adjustment is applied before calculating gram matrices. In Column (d), VGG-16 is used to synthesize the ZeroQ data. Best viewed in color.

roQ [3] in Table 2. In addition, we also compare with a variant of ZeroQ that we term as ZeroQ-real, short for ZeroQ with real images. Different from ZeroQ that initializes the synthesized data with random Gaussian values, ZeroQ-real uses real images for initialization. In all the experiments of ZeroQ-real, ImageNet data is used for initialization. We show performance of two variants of our method with one or multiple feature maps in VGG-16 to compute domain discrepancy, shown as Cross-domain and Cross-domain (MS) respectively. When multiple feature maps are used, each feature map will generate a Gram matrix and the domain discrepancy is defined as the average of L_2 distances of all Gram matrices. As shown in Table 2, our method achieves comparable or even better results than both ZeroQ and ZeroQ-real. Using one or multiple feature maps to compute domain discrepancy achieves similar performance across different tasks. Our method using selected cross-domain calibration data is on-par with in-domain data, proving the effectiveness of using cross-domain calibration data for post-training quantization.

In Figure 5, we show a visual comparison of gram matrices of different datasets. We show Gram matrices of ImageNet, and two datasets that have the smallest and largest domain discrepancy to ImageNet (*i.e.*, Open Images and Ultrasound). In addition, we also show Gram matrix of synthetic data from ZeroQ. We use VGG-16 with BatchNorm trained on ImageNet as feature extractor. BatchNorm adjustment is applied when using Open Images and Ultrasound datasets. As shown in Figure 5, Open Images shares a similar gram matrix with ImageNet. In contrast, Ultrasound dataset has a very different Gram matrix since it is not natural images. On the other hand, the Gram matrix of the synthetic ZeroQ data is also drastically different from the the two natural image datasets ImageNet and Open Images, showing less spatial correlation than the others. Our hypothesis is that with BatchNorm statistics as the only guidance to synthesize images is not sufficient to build spatial correlations as in real images. The lack of spatial correlation might be a key reason that ZeroQ does not achieve satisfactory performance on tasks such as NIH Chest X-ray classification.

3.6. Visualization of Activation Ranges

In this section, we visualize the activation ranges (clipping thresholds) calculated on different out-of-domain datasets. In Figure 6, we show three examples: Ultrasound to calibrate ResNet-18 on ImageNet, NIH Chest X-ray to calibrate FCN on MSCOCO, and ImageNet to calibrate ResNet-50 on NIH Chest X-ray. In each sub-figure, we show the lower and upper clipping thresholds for each layer. Compared with the naive cross-domain calibration method, the proposed BatchNorm adjustment makes a closer estimation of activation ranges to those from the in-domain data. Specifically, for ResNet-50 on NIH Chest X-ray, Batch-

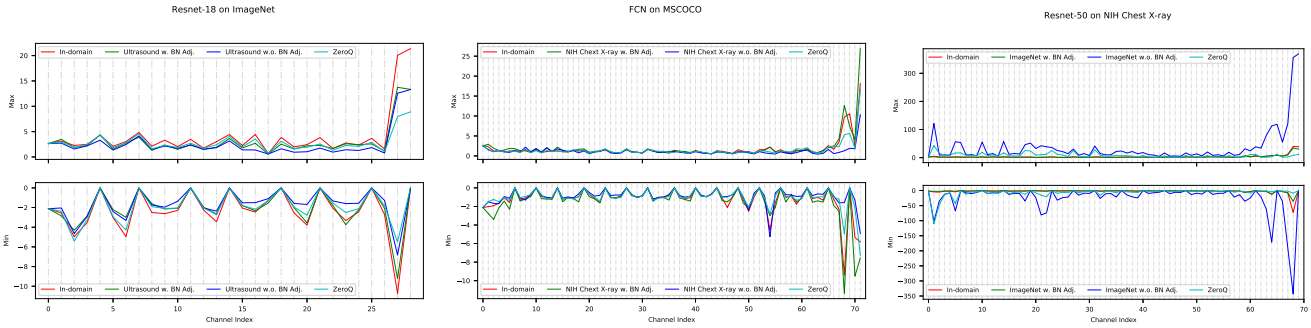


Figure 6: Visualization of calculated activation ranges of different methods. Three examples are shown from left to right: ResNet-18 on ImageNet, FCN on MSCOCO and ResNet-50 on NIH Chest X-ray. In each sub-figure, upper and lower clipping thresholds are plotted separately.

Calib Methods \ Datasets	IN			CO	CS	VO	CI	CE	CB	AG	US	NI
	R18	R50	IV3									
FP32	69.76	76.13	77.46	60.47	79.10	0.686	76.50	91.74	89.63	54.63	0.730	0.773
In-domain	62.19	66.80	65.81	45.97	52.57	0.612	74.86	91.00	87.01	49.11	0.739	0.711
ZeroQ	61.63	66.27	64.76	52.07	50.21	**0.614**	74.18	90.37	81.98	**48.73**	0.741	0.523
ZeroQ-real	61.80	66.22	63.87	**52.11**	49.78	0.611	74.43	**90.49**	86.54	47.82	**0.742**	0.502
Cross-domain	**62.18**	**66.82**	**65.52**	50.90	53.43	0.608	74.12	90.02	**87.06**	48.37	0.729	0.712
Cross-domain (MS)	**62.18**	**66.82**	**65.52**	50.22	**53.87**	0.608	**74.44**	90.02	**87.06**	48.15	0.729	**0.716**

Table 3: 6-bit quantization results with different calibration datasets. The dataset abbreviations are defined in Figure 2. Best results without using in-domain data are emphasized in bold.

Norm adjustment shows significant improvement over the baseline with original BatchNorm parameters. On the other hand, our estimation is also better than ZeroQ showing that calibration with cross-domain data is more robust than synthetic data in these cases.

3.7. Lower-bit Quantization

We also explore cross-domain calibration for 6-bit quantization. We use the same weight and activation quantization schemes as used in 8-bit experiments. Again, we show performance of two variants of our method with one or multiple feature maps to compute domain discrepancy. The results are summarized in Table 3. Best calibration results without using in-domain data are in bold. In most cases, our proposed cross-domain method achieves comparable or better performance than ZeroQ and its variant. Specifically, on Cityscapes and NIH Chest X-ray, our method outperforms ZeroQ by a large margin.

4. Discussion

In this work, we show the feasibility of using out-of-domain data for post-training quantization, which is different from the assumption of existing works that in-domain calibration dataset is necessary. Data synthesis methods are time-consuming and may not perform well in some cases as shown in our experiments. With stable and superior performance on a wide range of tasks, our method can be a new direction of post-training quantization when in-domain data are not available.

There are some interesting topics to explore in future works. First, cross-domain calibration can be improved by designing a better dataset pool consisting of a large amount of diverse domains. Second, better domain discrepancy measures can be explored to further improve the performance of quantized models.

5. Conclusion

In this work, we explored cross-domain calibration for post-training quantization. To study this problem, we conducted a large-scale study that spans across various tasks, datasets and neural networks. We find that a simple BatchNorm adjustment strategy can effectively improve the performance of quantized models by a large margin, almost bridging the gap between cross-domain calibration and in-domain calibration. In addition, we find that performance with cross-domain calibration is correlated with Gram matrix similarity between the source and the calibration domains. Therefore, Gram matrix similarity can be used as a criterion to select calibration dataset from a candidate pool to further improve performance. We believe our work will motivate future research on utilizing cross-domain knowledge for network quantization and compression.

References

[1] Ultrasound nerve segmentation. https://www.kaggle.com/c/ultrasound-nerve-segmentation/overview, 2016.

[2] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Advances in Neural Information Processing Systems*, pages 7950–7958, 2019.

[3] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020.

[4] Mang Tik Chiu, Xingqian Xu, Yunchao Wei, Zilong Huang, Alexander G Schwing, Robert Brunner, Hrant Khachatrian, Hovnatan Karapetyan, Ivan Dozier, Greg Rose, et al. Agriculture-vision: A large aerial image database for agricultural pattern analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2828–2838, 2020.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[6] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[9] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[10] Matan Haroush, Itay Hubara, Elad Hoffer, and Daniel Soudry. The knowledge within: Methods for data-free model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2020.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] 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.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[14] Qing Jin, Linjie Yang, and Zhenyu Liao. Towards efficient training for neural network quantization, 2019.

[15] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4350–4359, 2019.

[16] Jakub Konečný, Brendan McMahan, and Daniel Ramage. Federated optimization:distributed optimization beyond the datacenter, 2015.

[17] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

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

[21] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[22] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] Qinghui Liu, Michael C. Kampffmeyer, Robert Jenssen, and Arnt-Borre Salberg. Multi-view self-constructing graph convolutional networks with adaptive class weighting loss for semantic segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[27] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of*

the European conference on computer vision (ECCV), pages 722–737, 2018.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[29] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1325–1334, 2019.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[31] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[32] 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.

[33] 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.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

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

[36] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[37] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. *arXiv preprint arXiv:2003.03603*, 2020.

[38] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, pages 334–349. Springer, 2018.

[39] 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.

[40] Fan Zhang, Meng Li, Guisheng Zhai, and Yizhao Liu. Multi-branch and multi-scale attention learning for fine-grained visual categorization, 2020.

[41] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Christopher De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. *arXiv preprint arXiv:1901.09504*, 2019.

[42] 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.

[43] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.