Data-Free Quantization via Pseudo-label Filtering

Chunxiao Fan[1,2], Ziqi Wang[1], Dan Guo[1,2*], Meng Wang[1,2]

[1]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230009, Anhui, China

[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, 230088, Anhui, China

fanchunxiao@hfut.edu.cn, zackiewang29@gmail.com, guodan@hfut.edu.cn, eric.mengwang@gmail.com

Abstract

Quantization for model compression can efficiently reduce the network complexity and storage requirement, but the original training data is necessary to remedy the performance loss caused by quantization. The Data-Free Quantization (DFQ) methods have been proposed to handle the absence of original training data with synthetic data. However, there are differences between the synthetic and original training data, which affects the performance of the quantized network, but none of the existing methods considers the differences. In this paper, we propose an efficient data-free quantization via pseudo-label filtering, which is the first to evaluate the synthetic data before quantization. We design a new metric for evaluating synthetic data using self-entropy, which indicates the reliability of synthetic data. The synthetic data can be categorized with the metric into high- and low-reliable datasets for the following training process. Besides, the multiple pseudo-labels are designed to label the synthetic data with different reliability, which can provide valuable supervision information and avoid misleading training by low-reliable samples. Extensive experiments are implemented on several datasets, including CIFAR-10, CIFAR-100, and ImageNet with various models. The experimental results show that our method can perform excellently and outperform existing methods in accuracy.

1. Introduction

Deep Neural Networks (DNNs) [26] have shown tremendous potential in a number of fields, but their high computing costs and storage requirements make the implementation complex, especially on embedded systems and edge devices [9, 36]. In recent years, many model compression methods [7] have been proposed to decrease the computational and memory requirements while keeping the performance. The existing methods can be divided into network pruning [41, 45], model quantization [2, 21], knowledge distillation [18, 31], and neural architecture search (NAS) [33, 39]. Among these techniques, quantization uses finite approximations to represent the full-precision values in the pre-trained network, which needs to be quantized. It can efficiently reduce the network complexity for acceleration and storage, but the approximate operation inevitably affects the network performance, resulting in accuracy drops after quantization.

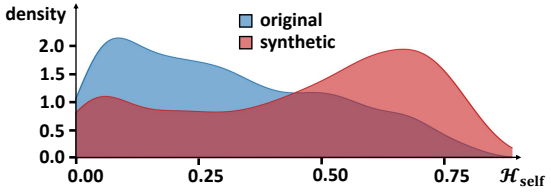

Figure 1. The self-entropy of the prediction results on the original and synthetic data using pre-trained network. The prediction results on the synthetic dataset always have a higher self-entropy than that on the original dataset, owing to different reliability.

To reduce the performance loss caused by quantization, many methods propose to optimize the quantizer [1, 2, 10, 11, 35] or retrain the pre-trained network with quantization constraint [4, 15, 27]. In these methods, the original training data is very helpful for maintaining model performance, but it may not be feasible due to data privacy concerns in specific scenarios. The Data-Free Quantization (DFQ) methods [3, 5, 28, 30, 42, 47, 48] have been proposed to deal with the absence of original training data. In principle, the batch normalization (BN) [20] layers in the pre-trained model contain the statistical information of the original training data, *i.e.* mean and variance, which can be regarded as prior information for data synthesis [46]. Thus, the synthetic data can be generated from random initial data by guiding its distribution closer to the original data with the full-precision pre-trained model. Generally, exsiting DFQ methods can be divided into Generator-Based Approach (GBA) [5, 22, 37, 42] and Distill-Based Approach (DBA) [3, 28, 30, 47].

The GBA trains a specific generative network (*e.g.*, GAN [12] or VAE [24]) to generate the synthetic data, which re-

*Corresponding author.

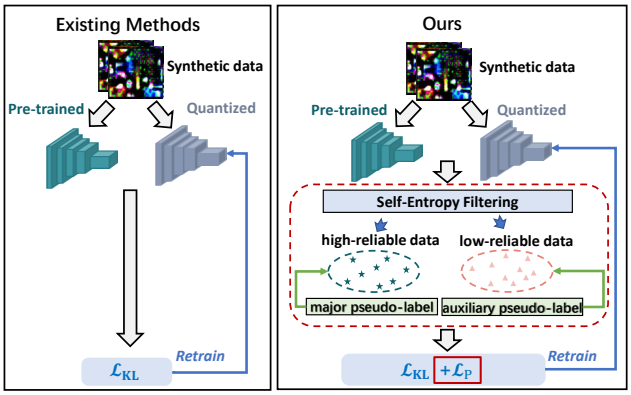

Figure 2. The difference between existing data-free methods and our method. The existing methods optimize the quantizer or retrain the pre-trained network with quantization constraint directly without evaluating the synthetic data. We propose to evaluate the synthetic data with self-entropy and divide the synthetic data into high- and low-reliable datasets before training the quantized network for better performance.

quires a complex training process for the generative network. The DBA regards the synthetic data as trainable, and iterates them to fit the original data distribution using the back-propagation of the pre-trained model, avoiding the training process for the generative network. However, although many exquisite technologies are designed in these existing methods, a noticeable difference still exists between the synthetic and original training data, which affects the performance of the quantized network. Thus, it is necessary to evaluate the synthetic training data before quantization.

In this paper, we propose to evaluate the synthetic data before quantization as shown in Figure 2. To complete this goal, we aim to deal with three problems: 1) *How to evaluate the synthetic data?* In existing frameworks, the synthetic training data is generated with the information in the pre-trained model, and it lacks clear evaluation indicators for the synthetic data. 2) *How to label the evaluated synthetic data?* After evaluating the synthetic data, it is necessary to assign suitable labels to the synthetic data according to different evaluation results, which aims to further improve the reliability of synthetic data and avoid misleading caused by inappropriate labels. 3) *How to design the training process with evaluated synthetic data?* The training process needs to be able to learn supervision information from the synthetic data with different labels, and more importantly, avoid misleading caused by the data with low evaluation results.

To overcome the problems above, we propose an efficient Data-Free Quantization via Pseudo-label Filtering as follows: **1)** The self-entropy [23] is used as a metric for synthetic data to evaluate the reliability of the pre-trained network on it. We notice that the pre-trained network has a relatively higher self-entropy on the synthetic dataset than the original dataset, as shown in Figure.1, owing to different reliability performance, so the self-entropy can be used as a metric to evaluate the reliability of synthetic data before quantization. **2)** The synthetic data is labeled using multiple pseudo-labels, which include major and auxiliary pseudo-labels, to reflect its reliability. Major pseudo-labels are assigned to high-reliable samples, providing valuable supervision information. In contrast, low-reliable samples are labeled with auxiliary pseudo-labels to give a soften supervised learning for enhancing the robustness of the quantized network and avoiding misled by low-reliable samples. **3)** The pseudo-label training is designed to integrate the supervision information provided by major and auxiliary pseudo-labels. The knowledge distillation is selected as the basic framework to train the quantized model, which has similar performance and intermediate features as the pre-trained network using the proposed synthetic data evaluation.

The main contributions of this work can be summarized as follows:

- We propose an efficient data-free quantization via pseudo-label filtering, which is the first to evaluate the synthetic data before quantization, so that the samples with different reliability can provide different information and improve the performance.
- The self-entropy is used as the metric for the synthetic data evaluation, which represents the reliability of synthetic data under a specific label. With the metric, the synthetic data can be categorized into high- and low-reliable samples for the following training process.
- The multiple pseudo-labels are designed in our method to label the synthetic data with different reliability. It incorporates major and auxiliary pseudo-labels for the high- and low-reliable samples, providing valuable supervision information and avoiding misleading quantization by low-reliable samples.
- Extensive experiments are conducted on the CIFAR-10, CIFAR-100, and ImageNet in various models. Our method can achieve superior performances compared with existing DFQ methods.

2. Related Work

Quantization. Quantization is widely used in the model compression for neural networks to represent the full-precision model with low-bit approximations. To alleviate the performance loss caused by approximate operations, many methods have been proposed, which can be grouped into post-training quantization (PTQ) [1, 13, 34, 35] and quantization-aware training (QAT) [2, 4, 10, 11]. The PTQ methods take calibration data to optimize the parameters in the quantizer, and the QAT methods retrain the pre-trained network with a quantization constraint. However, the original training data is necessary for these methods, it may not be feasible in specific scenarios. To deal with the challenge

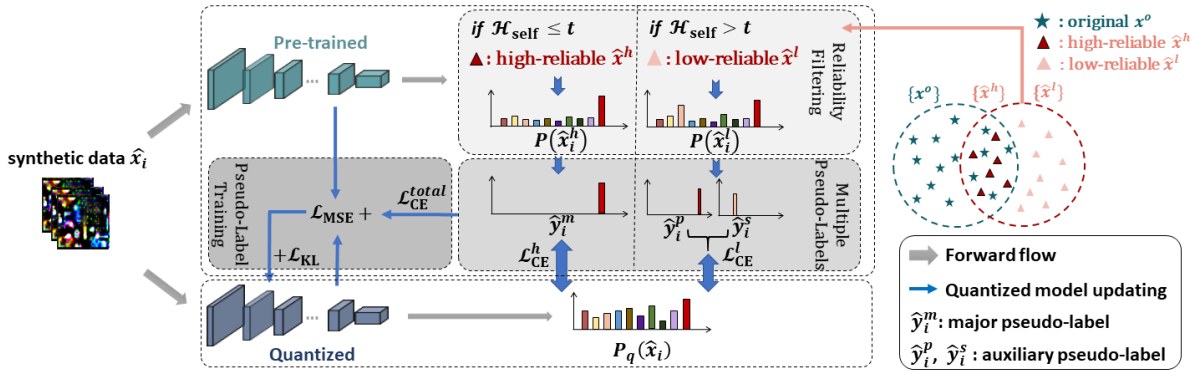

Figure 3. The framework of the proposed method. In the proposed method, synthetic data is evaluated with self-entropy of the pre-trained network on the synthetic data, which is divided into high- and low-reliable data. Then, the multiple pseudo-labels are used to label the evaluated synthetic data according to its reliability. The high-reliable samples are assigned with major pseudo-labels, and the low-reliable samples use auxiliary pseudo-labels. With these multiple pseudo-labels, the pseudo-labels cross-entropy loss can be obtained and combined with the MSE loss function to guide the quantized network has similar prediction ability with pre-trained network.

without original training data, the Data-Free Quantization (DFQ) is proposed.

Data-free Quantization. Many works [14, 34, 43] attempt to optimize the quantized network solely relying on the information inherent in the pre-trained network itself without the demand for training data. D-FQ [34] proposes the method of weight equalization and bias correction to make the network more suitable to quantize. SQuant [14] adopts a new rounding metric based on a diagonal Hessian approximation to improve the performance of the quantized network. However, owing to the absence of training data to adjust the pre-trained network, these methods can hardly achieve a significant performance improvement.

Many methods propose to utilize generated data, and can be divided into Generator-Based Approach (GBA) [5, 32, 37, 38, 42, 44, 48] and Distill-Based Approach (DBA) [3, 30, 30, 47]. ***(1) GBA*** proposes to use a generator for synthesizing training data. GDFQ [42] constructs informative data from the pre-trained model and generates data approximating the original dataset. Qimera [5] proposes to use superposed latent embeddings to generate higher-quality samples. These methods always demand much time and resources to generate high-quality data. ***(2) DBA*** utilizes the pre-trained model to directly optimize the generated data, thereby eliminating the need for the generator. ZeroQ [3] matches the statistics of batch normalization to optimize for a distilled dataset. IntraQ [47] attempts to synthesize images with intra-class heterogeneity. HAST [28] generates more hard samples to enhance model training effectiveness.

However, even with many ingenious designs, there is a difference between the generated synthetic data and the original training data, but none of the existing methods takes this into account, and the synthetic data is directly used for the training of the quantized network. In the proposed method, we propose to evaluate the generated synthetic data and label different samples according to the evaluation results to enhance the performance of the quantized network.

3. Proposed Method

The framework of our proposed method is illustrated in Figure 3, including Reliability Filtering, Multiple Pseudo-Labels, and Pseudo-Label Training.

3.1. Reliability Filtering

In principle, the mean and variance in BN layers of the pre-trained model are affected by the original training data. Thus, the synthetic data can be generated from random initial samples by adapting the output distribution of BN layers the same as that stored in the pre-trained model with:

$$\mathcal{L}_{\text{BN}} = \sum_{l=1}^{L} \left(||\boldsymbol{\mu}_l^p - \boldsymbol{\mu}_l^s||_2^2 + ||\boldsymbol{\sigma}_l^p - \boldsymbol{\sigma}_l^s||_2^2 \right), \tag{1}$$

where L denotes the layer number of the model. $\boldsymbol{\mu}_l^p$ and $\boldsymbol{\sigma}_l^p$ are the mean and variance stored in l-th BN layer of the pre-trained model, and $\boldsymbol{\mu}_l^s$ and $\boldsymbol{\sigma}_l^s$ are the mean and variance of synthetic data batch in l-th layer. The synthetic data is generated by minimize $\mathcal{L}_{\text{DATA}}$ as,

$$\mathcal{L}_{\text{DATA}} = \mathcal{L}_{\text{BN}} + \gamma \cdot \mathcal{L}_{\text{IL}}, \tag{2}$$

where $\mathcal{L}_{\text{IL}} = \sum_{i=1}^{N} \text{CE}\left(P(\hat{\boldsymbol{x}}_i), \hat{y}_i\right)$ is the loss to improve the predicted probability of pre-trained network for the assigned label [16]. N denotes the number of samples for the synthetic data, and $\text{CE}(\cdot)$ represents the cross-entropy loss. $\hat{\boldsymbol{x}}_i$ denotes the generated synthetic sample and $P\left(\hat{\boldsymbol{x}}_i\right)$ represents its predicted probability after the softmax layer. $\hat{y}_i$ denotes the assigned label. γ is a hyper-parameters to balance two losses of $\mathcal{L}_{\text{BN}}$ and $\mathcal{L}_{\text{IL}}$.

However, the synthetic data can hardly have precisely the same features as the original data. As discussed above, we notice that the synthetic dataset always has a higher self-entropy compared with the original dataset as shown in Figure 1. The reason is that the pre-trained network is trained on the original dataset, which can have high reliability on most samples. Owing to the difference between the original

and synthetic data, it is impossible for the pre-trained network to have high reliability on the whole synthetic data, and the low-reliable data may lead to incorrect guidance during training and seriously influence the quantization performance.

Thus, we apply **self-entropy** as a metric to evaluate the reliability of the pre-trained network on the synthetic data as in Eq. 3:

$$\mathcal{H}_{\text{self}}(\hat{\boldsymbol{x}}_i) = -\frac{1}{\log N_c}\sum_{c=1}^{N_c}\left(\mathbb{P}(\hat{\boldsymbol{x}}_i, c)\cdot\log\left(\mathbb{P}(\hat{\boldsymbol{x}}_i, c)\right)\right), \quad (3)$$

where N_c refers to the number of prediction classes, and $\mathbb{P}\left(\hat{\boldsymbol{x}}_i, c\right)$ denotes the predicted probability of the class c obtained by the pre-trained network.

In principle, $\mathcal{H}_{\text{self}}$ indicates the uncertainty, whereas a low self-entropy for the predicted results can refer to a reliable prediction with high confidence [23]. In other words, low $\mathcal{H}_{\text{self}}$ means the pre-trained network can have a high possibility for the specific label on the synthetic data, which means it has high reliability for the samples, so it can provide adequate supervision information for training the quantized network to fit the performance of the pre-trained network.

Based on the reliability evaluation, the samples in the synthetic dataset $\hat{X}$ can be divided into high-reliable dataset $\hat{X}^h$ and low-reliable dataset $\hat{X}^l$ with a reliable threshold t as,

$$\begin{aligned}\hat{X}^h &= \left\{\hat{\boldsymbol{x}}^h \middle| \hat{\boldsymbol{x}}^h \in \hat{X},\ \mathcal{H}_{\text{self}}\left(\hat{\boldsymbol{x}}^h\right) \leqslant t\right\},\\ \hat{X}^l &= \left\{\hat{\boldsymbol{x}}^l \middle| \hat{\boldsymbol{x}}^l \in \hat{X},\ \mathcal{H}_{\text{self}}\left(\hat{\boldsymbol{x}}^l\right) > t\right\},\end{aligned} \quad (4)$$

where t is a dynamic threshold parameter for fast convergence and learning more supervision information. To converge fast, the requirement for high-reliable samples can be relaxed at the beginning of training to provide more samples and quickly improve network performance; As the training progresses, the network performance gradually improves, which raises the standard for high-reliable samples. It is necessary to use high-reliable samples to provide better supervision information and avoid the impact of low-reliable samples on network performance. At the same time, using more low-reliable samples can also improve the robustness of the network. Thus, t continuously decrease as the training epoch increases:

$$t = T_u - f_t(epoch)(T_u - T_l), \quad (5)$$

where T_l and T_u are the lower and upper boundaries, and $t \in [T_l,\ T_u]$, which continuously decrease as the training epoch increase. $f_t(epoch) = \frac{epoch}{E}$, $epoch$ and E denote the current and whole epoch for training, respectively.

3.2. Multiple Pseudo-Labels

With the reliability filtering, the samples in $\hat{X}^h$ have high reliability and can provide more supervision information for the quantized model. In the proposed method, **major pseudo-labels** are designed for these samples, and the pseudo-label $\hat{y}_i^m$ for the high-reliable sample $\hat{\boldsymbol{x}}_i^h$ is assigned as,

$$\hat{y}_i^m = \arg\max_{c\in\mathbb{C}}\ \mathbb{P}(\hat{\boldsymbol{x}}_i^h, c), \quad (6)$$

where $\mathbb{C} = \{1, 2, ..., N_c\}$ denotes the set of all the possible prediction classes. We define the loss of $\mathcal{L}_{\text{CE}}^h$ for supervised learning with high-reliable dataset:

$$\mathcal{L}_{\text{CE}}^h = \frac{1}{N_h}\sum_{i=1}^{N_h}\text{CE}\left(P_q(\hat{\boldsymbol{x}}_i^h), \hat{y}_i^m\right), \quad (7)$$

where N_h refers to the number of high-reliable samples, and $P_q\left(\hat{\boldsymbol{x}}_i^h\right)$ denotes the predicted probability for all classes with the quantized network on high-reliable samples $\hat{\boldsymbol{x}}_i^h$.

Generally, the high-reliable samples account for a small portion of the total samples [23]. In quantization, we aim to train the quantized network, which can fit the performance of the pre-trained network. The low-reliable synthetic data can also reflect the features of the pre-trained network and provide some information for robustness. But owing to the low predicted probability for the labels with low reliability, they may mislead the training if applied directly in training.

To deal with it, **auxiliary pseudo-labels** are designed to give a soften supervised learning with the low-reliable samples, which consists of the primary label $\hat{y}_i^p$ and secondary label $\hat{y}_i^s$. The primary label $\hat{y}_i^p$ represents the label with the highest predicted probability, and the secondary label $\hat{y}_i^s$ represents that with the second highest probability. In the prediction with high self-entropy, the labels excluding $\hat{y}_i^p$ can also have a decent probability, especially the secondary label $\hat{y}_i^s$. Thus, the pseudo-labels from $\hat{y}_i^p$ and $\hat{y}_i^s$ can be leveraged to enhance the training and mitigate the risk of misleading for these samples [29]. The primary label $\hat{y}_i^p$ and secondary label $\hat{y}_i^s$ can be obtained as,

$$\begin{aligned}\hat{y}_i^p &= \arg\max_{c\in\mathbb{C}}\ \mathbb{P}(\hat{\boldsymbol{x}}_i^l, c),\\ \hat{y}_i^s &= \arg\max_{c\in\hat{\mathbb{C}}}\ \mathbb{P}(\hat{\boldsymbol{x}}_i^l, c),\end{aligned} \quad (8)$$

where $\hat{\mathbb{C}}$ denotes the set of classes removing the class with the highest predicted probability.

The auxiliary cross-entropy $\mathcal{L}_{\text{CE}}^l$ loss are defined with the primary and secondary labels $\hat{y}_i^p$, $\hat{y}_i^s$ as,

$$\begin{aligned}\mathcal{L}_{\text{CE}}^l = \frac{1}{N_l}\sum_{i=1}^{N_l}\Big(&\lambda_i^p\cdot\text{CE}\left(P_q(\hat{\boldsymbol{x}}_i^l), \hat{y}_i^p\right)\\ &+\lambda_i^s\cdot\text{CE}\left(P_q(\hat{\boldsymbol{x}}_i^l), \hat{y}_i^s\right)\Big),\end{aligned} \quad (9)$$

where N_l refers to the number of low-reliable samples, λ_i^p and λ_i^s represent the weights of primary and secondary labels to balance their importance, and can be obtained as,

$$\begin{aligned}\lambda_i^p &= \frac{\mathbb{P}(\hat{\boldsymbol{x}}_i^l, p)}{\mathbb{P}(\hat{\boldsymbol{x}}_i^l, p) + \mathbb{P}(\hat{\boldsymbol{x}}_i^l, s)},\\ \lambda_i^s &= \frac{\mathbb{P}(\hat{\boldsymbol{x}}_i^l, s)}{\mathbb{P}(\hat{\boldsymbol{x}}_i^l, p) + \mathbb{P}(\hat{\boldsymbol{x}}_i^l, s)},\end{aligned} \quad (10)$$

where $\mathbb{P}(\hat{\boldsymbol{x}}_i^l, p)$ and $\mathbb{P}(\hat{\boldsymbol{x}}_i^l, s)$ denote the predicted probabilities of the primary and secondary labels with the pre-trained model, respectively.

With the designed major and auxiliary pseudo-labels, the high- and low-reliable samples can be simultaneously used to provide the supervision information for model training:

$$\mathcal{L}_{\mathrm{CE}}^{total} = \mathcal{L}_{\mathrm{CE}}^{h} + \beta \cdot \mathcal{L}_{\mathrm{CE}}^{l}, \tag{11}$$

where β is the hyper-parameter to balance the importance of samples with different reliability, which is less than 1.

3.3. Pseudo-Label Training

To train a high-performance quantized model, we design a pseudo-label training based on the knowledge distillation framework [18], whereby the pre-trained network is employed as the teacher for the quantized network. In order to learn the supervision information from the designed major and auxiliary pseudo-labels, the designed cross-entropy loss $\mathcal{L}_{\mathrm{CE}}^{total}$ is used in training and combined with $\mathcal{L}_{\mathrm{MSE}}$ to form the prediction similarity loss $\mathcal{L}_{\mathrm{P}}$, which aims to guide the quantized network has similar prediction ability with pre-trained network as,

$$\mathcal{L}_{\mathrm{P}} = \mathcal{L}_{\mathrm{CE}}^{total} + \mu \cdot \mathcal{L}_{\mathrm{MSE}}, \tag{12}$$

where μ is the hyperparameter to balance the two losses. $\mathcal{L}_{\mathrm{MSE}}$ denotes the Mean Squared Error (MSE) between the intermediate feature layers of the two networks. Owing to the reduction of bits number for weights and activations in the quantized network, the features extracted by intermediate layers are significantly affected, resulting in performance degradation. $\mathcal{L}_{\mathrm{MSE}}$ is utilized to minimize the difference between the intermediate feature layers of the two networks:

$$\mathcal{L}_{\mathrm{MSE}} = \sum_{k=1}^{L} \left(\frac{1}{N} \sum_{i=1}^{N} \left(f_k\left(\hat{\boldsymbol{x}}_i\right) - f_k^q\left(\hat{\boldsymbol{x}}_i\right) \right)^2 \right), \tag{13}$$

where $f_k\left(\hat{\boldsymbol{x}}_i\right)$ and $f_k^q\left(\hat{\boldsymbol{x}}_i\right)$ represent the outputs in the k-th layer of the full-precision model and the quantized model, respectively.

In common, the Kullback-Leibler (KL) divergence $\mathcal{L}_{\mathrm{KL}}$ [3] between the outputs of the pre-trained and quantized networks can be used to minimize the discrepancy between the outputs of the two networks, thereby ensuring the performance of the quantized network.

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{N} \sum_{i=1}^{N} \left(P(\hat{\boldsymbol{x}}_i) \cdot log \frac{P(\hat{\boldsymbol{x}}_i)}{P_q(\hat{\boldsymbol{x}}_i)} \right), \tag{14}$$

where $P\left(\hat{\boldsymbol{x}}_i\right)$ and $P_q\left(\hat{\boldsymbol{x}}_i\right)$ denotes predicted probability using the pre-trained and quantized networks, respectively.

Thus, the total loss function for the entire training process can be obtained as,

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{KL}} + \tau \mathcal{L}_{\mathrm{P}}, \tag{15}$$

where hyperparameter τ is used to balance the weights of the two losses.

4. Experiment

4.1. Experiment Setup

Datasets and Networks. Data-Free Quantization is typically evaluated on CIFAR-10/100 [25] and ImageNet (ILSVRC2012) [8] datasets. The proposed method is implemented and examined with ResNet [17] and MobileNet [19], *i.e*. ResNet-20 on CIFAR-10/100; ResNet-18/50, and MobileNet-V1 on ImageNet (ILSVRC2012).

Implementation Details. For a fair comparison, we synthesize 5,120 images as the synthetic data, which is the same as the settings of IntarQ [47] and HAST [28], and these synthetic samples are optimized with 1000 iterations. The hyper-parameters γ are set as 10 for CIFAR-10/100 and 0.1 for ImageNet. We adopt the SGD optimizer with a weight decay of 10^{-4} and momentum of 0.9 for model training. The initial learning rate is set as 0.001 for CIFAR-10/100 and 10^{-5} for ImageNet. The hyper-parameters T_l, T_u, β, μ and τ are respectively set to 0.2, 0.5, 0.3, 100 and 1 for CIFAR-10/100 and 0.1, 0.4, 0.5, 4000 and 1 for ImageNet. All full-precision pre-trained models are provided by pytorchcv [42]. All layers are quantized, including the first and last layers of the model. Our implementation is conducted with PyTorch on a GPU Nvidia GTX 3090 Ti workstation, CUDA 11.4, and Ubuntu 18.04.

4.2. Comparisons with State-of-the-Art Methods

The proposed method is compared with state-of-the-art data-free quantization methods on CIFAR-10/100 and ImageNet as listed in Tables 1∼4. W-bit/A-bit represent the bits number of weights and activations after quantization, respectively. Among these methods, GDFQ [42], DSG [44], ZAQ [32], Qimera [5], ARC [6], ARC+AIT [48], AdaSG [38] and AdaDFQ [37] belong to GBA. ZeroQ [3], IntraQ [47] and HAST [28] belong to DBA.

CIFAR-10/100. For ResNet-20 on CIFAR-10/100, the model is quantized into 4-bit and 3-bit. As shown in Table 1, the proposed method achieves the superior performances at 92.47% (4-bit), 88.04% (3-bit) on CIFAR-10 and 66.94% (4-bit), 57.03% (3-bit) on CIFAR-100. So our method can get the higher performance compared with most of exsiting methods, *i.e*. +0.98% over IntraQ, +0.37% over AdaSG, +0.16% over AdaDFQ, and +0.11% over HAST on CIFAR-10 (4-bit). The performance improvement is higher on CIFAR-100 than CIFAR-10, *i.e*. +1.96% (4-bit) and +8.78% (3-bit) over IntraQ, +0.52% (4-bit) and +4.27% (3-bit) over AdaSG, +0.13% (4-bit) and +4.29% (3-bit) over AdaDFQ, and +0.26% (4-bit) and +1.36% (3-bit) over HAST. Generally, the proposed method can get the best performance except in 3-bit on CIFAR-10 dataset (0.30% lower

Table 1. Top-1 accuracy (%) comparison with the state-of-the-art methods on CIFAR-10, CIFAR-100 for 3/4-bit **ResNet-20**. * represents the results reimplemented in paper [37].

Dataset	Method	Venue	W4A4	W3A3
CIFAR-10 For ResNet-20 (FP:94.03)	ZeroQ[3]	CVPR'20	84.68	29.32
	GDFQ[42]	ECCV'20	90.25	71.10
	DSG[44]	CVPR'21	88.74	32.90
	Qimera[5]	NeurIPS'21	91.26	74.43*
	ARC[6]	IJCAI'21	88.55	-
	ARC+AIT[48]	CVPR'22	90.49	-
	IntraQ[47]	CVPR'22	91.49	77.07
	AdaSG[38]	AAAI'23	92.10	84.14
	AdaDFQ[37]	CVPR'23	92.31	84.89
	HAST[28]	ICCV'23	92.36	**88.34**
	Ours	-	**92.47**	88.04
CIFAR-100 For ResNet-20 (FP:70.33)	ZeroQ[3]	CVPR'20	58.42	15.38
	GDFQ[42]	ECCV'20	63.58	43.87
	DSG[44]	CVPR'21	62.36	25.48
	Qimera[5]	NeurIPS'21	65.10	46.13*
	ARC[6]	IJCAI'21	62.76	40.15*
	ARC+AIT[48]	CVPR'22	61.05	41.34*
	IntraQ[47]	CVPR'22	64.98	48.25
	AdaSG[38]	AAAI'23	66.42	52.76
	AdaDFQ[37]	CVPR'23	66.81	52.74
	HAST[28]	ICCV'23	66.68	55.67
	Ours	-	**66.94**	**57.03**

Table 2. Top-1 accuracy (%) comparison with the state-of-the-art methods on ImageNet for 4/5-bit **ResNet-18**.

Dataset	Method	Venue	W5A5	W4A4
ImageNet For ResNet-18 (FP:71.47)	ZeroQ[3]	CVPR'20	69.65	60.68
	GDFQ[42]	ECCV'20	66.82	60.60
	DSG[44]	CVPR'21	69.53	60.12
	ZAQ[32]	CVPR'21	64.54	52.64
	Qimera[5]	NeurIPS'21	69.29	63.84
	ARC[6]	IJCAI'21	68.88	61.32
	ARC+AIT[48]	CVPR'22	70.28	65.73
	IntraQ[47]	CVPR'22	69.94	66.47
	AdaSG[38]	AAAI'23	70.29	66.50
	AdaDFQ[37]	CVPR'23	70.29	66.53
	HAST[28]	ICCV'23	-	66.91
	Ours	-	**70.35**	**67.02**

than HAST). By analysis, HSAT aims to improve the quality of synthetic data by increasing the proportion of hard samples, resulting in more high quality synthetic data, but our method emphasises on the synthetic data evaluation for better utilization of synthetic data.

ImageNet. To further verify the effectiveness of our method on the large-scale dataset, we compare the performance on ImageNet using different networks with state-of-the-art methods in Tables 2∼4. **(1)** The **ResNet-18** is implemented with the proposed method, and the results are listed in Table 2. It can be observed that the proposed method achieves the best performance among these methods, which are 70.35% (5-bit) and 67.02% (4-bit). Especially in the case of 4-bit, compared with GDFQ (60.60%), Qimera (63.84%), AdaDFQ (66.53%), and HAST (66.91%), our method can get much better performance. **(2)** The **MobileNet-V1** is selected for the evaluation of the proposed method on the light-weighted network as shown in Table 3. Generally, light-weighted networks always have a heavy accuracy drop after quantization, but the proposed method can also outperform existing methods, which get 0.92% and 1.81% higher accuracy than the advanced method HAST for 5-bit and 4-bit, respectively. **(3)** The **ResNet-50** is used to evaluate the performance of the proposed method on the network with complex and deeper structure as Table 4. The proposed method achieves superior performances at 76.08% (5-bit) and 68.97% (4-bit), especially compared with the latest methods AdaSG (68.58%) and AdaDFQ (68.38%) at 4-bit setup. Thus, the proposed method can efficiently improve performance on various network structures and datasets, proving the effectiveness of our proposed evaluation for synthetic data.

Table 3. Top-1 accuracy (%) comparison with the state-of-the-art methods on ImageNet for 4/5-bit **MobileNet-V1**. 'IL' denotes using the inception loss [16].

Dataset	Method	Venue	W5A5	W4A4
ImageNet For MobileNet-V1 (FP:73.39)	ZeroQ[3]+IL[16]	CVPR'20	67.11	25.43
	GDFQ[42]	ECCV'20	59.76	28.64
	DSG[44]+IL[16]	CVPR'21	66.61	42.19
	SQuant[13]	ICLR'22	64.20	10.32
	IntraQ[47]	CVPR'22	68.17	51.36
	HAST[28]	ICCV'23	68.52	57.70
	Ours	-	**69.44**	**59.51**

Table 4. Top-1 accuracy (%) comparison with the state-of-the-art methods on ImageNet for 4/5-bit **ResNet-50**.

Dataset	Method	Venue	W5A5	W4A4
ImageNet For ResNet-50 (FP:77.73)	GDFQ[42]	ECCV'20	71.63	54.16
	ZAQ[32]	CVPR'21	73.38	53.02
	Qimera[5]	NeurIPS'21	75.32	66.25
	ARC[6]	IJCAI'21	74.13	64.37
	ARC+AIT[48]	CVPR'22	76.00	68.27
	AdaSG[38]	AAAI'23	76.03	68.58
	AdaDFQ[37]	CVPR'23	76.03	68.38
	Ours	-	**76.08**	**68.97**

4.3. Ablation Study

Effect of Different Components in $\mathcal{L}_{\mathrm{P}}$. To evaluate the effect of each component in the proposed method, we test each item in the designed prediction similarity loss $\mathcal{L}_{\mathrm{P}}$. Table 5 lists different networks ($\mathcal{N}_1 \sim \mathcal{N}_7$) trained with different loss combinations (the common $\mathcal{L}_{\mathrm{KL}}$ is used in all the networks to provide the basic performance of quantized network). Symbol ✓ means the component is used for quantization training, and symbol × indicates that the component is removed for training. $\mathcal{L}_{\mathrm{CE}}^{h}$ and $\mathcal{L}_{\mathrm{CE}}^{l}$ denote the cross-

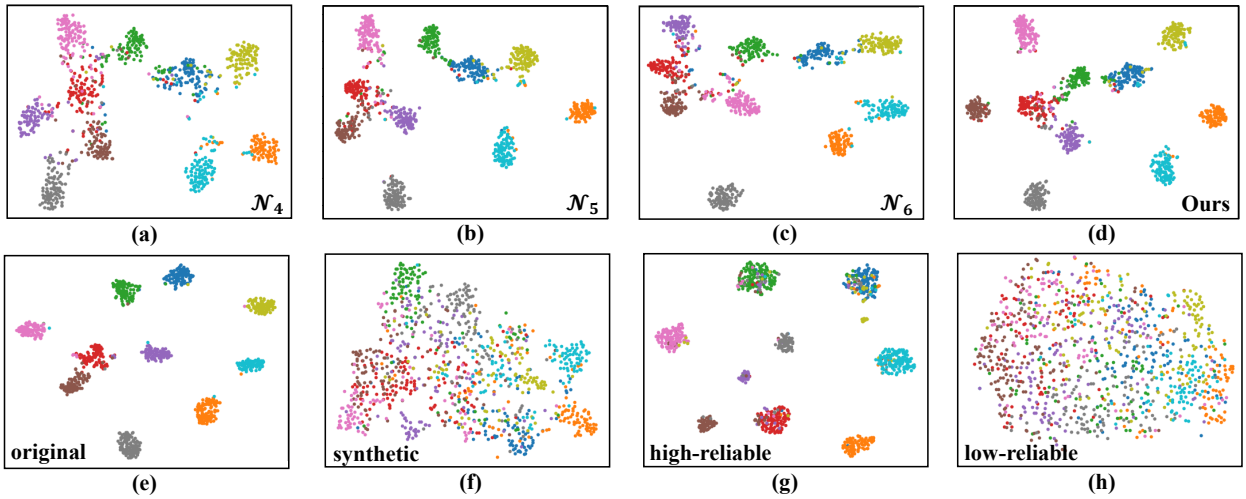

Figure 4. Feature visualization using t-SNE. Figure 4 (a∼d) show the feature distributions of the network $\mathcal{N}_4$ (w/o $\mathcal{L}_{\text{CE}}^h$ and $\mathcal{L}_{\text{CE}}^l$), $\mathcal{N}_5$ (w/o $\mathcal{L}_{\text{CE}}^l$), $\mathcal{N}_6$ (w/o $\mathcal{L}_{\text{CE}}^h$) and $\mathcal{N}_7$ (Ours) on the original test dataset. Figure 4 (e ∼ h) show the feature distributions of the pre-trained network on the original dataset, the synthetic dataset, the evaluated high-reliable dataset $\hat{X}^h$ and low-reliable dataset $\hat{X}^l$, respectively.

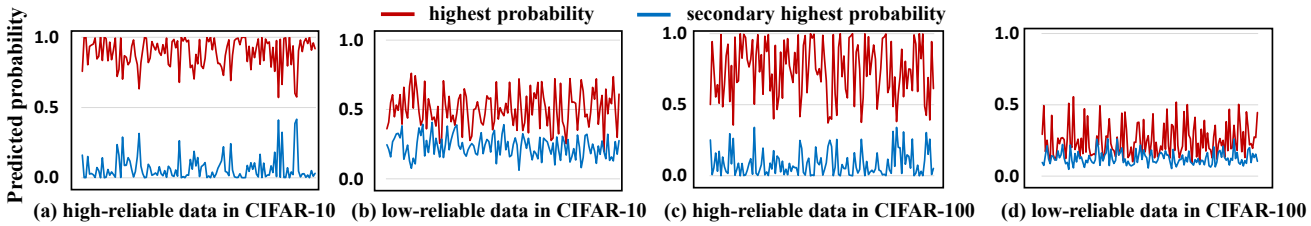

Figure 5. The highest and secondary highest predicted probabilities of pre-trained network on the high- and low-reliable datasets. The highest predicted probabilities on the high-reliable dataset are far higher than the secondary highest predicted probabilities, since the pre-trained network has high performance on these data. While on the low-reliable dataset, the pre-trained network cannot perform well, which can be reflected by the lower highest predicted probabilities and higher secondary highest predicted probabilities.

Table 5. Ablation studies of losses with **3/4-bit ResNet-20** on the CIFAR-100.

-	$\mathcal{L}_{\text{CE}}^h$	$\mathcal{L}_{\text{CE}}^l$	$\mathcal{L}_{\text{MSE}}$	W4/A4	W3/A3
$\mathcal{N}_1$	×	×	×	65.81	50.87
$\mathcal{N}_2$	✓	×	×	66.30	53.22
$\mathcal{N}_3$	✓	✓	×	66.52	54.14
$\mathcal{N}_4$	×	×	✓	66.32	54.59
$\mathcal{N}_5$	✓	×	✓	66.68	56.43
$\mathcal{N}_6$	×	✓	✓	66.57	55.86
$\mathcal{N}_7$	✓	✓	✓	**66.94**	**57.03**

entropy loss functions using high-reliable and low-reliable samples, respectively. $\mathcal{L}_{\text{MSE}}$ represents the usage of the MSE loss function. There are two observations. *(1) Multiple Pseudo-labels* can provide supervision information for model training, and improve the performance of the quantized network. In the comparison for $\mathcal{L}_{\text{CE}}^h$ and $\mathcal{L}_{\text{CE}}^l$ (*i.e.* comparing $\mathcal{N}_1$ with $\mathcal{N}_3$, and comparing $\mathcal{N}_4$ with $\mathcal{N}_7$), it is obvious that the designed $\mathcal{L}_{\text{CE}}^h$ and $\mathcal{L}_{\text{CE}}^l$ can improve the quantized network performance, which can prove the efficiency of proposed reliability filtering. By comparing the network with different combinations of $\mathcal{L}_{\text{CE}}^h$ and $\mathcal{L}_{\text{CE}}^l$

Table 6. Top-1 accuracy (%) of the combination of GDFQ [42], IntraQ [47] and our training method on CIFAR-100 for 4-bit **ResNet-20** and ImageNet for 4-bit **ResNet-18**. The combination with our multiple pseudo-labels is denoted as $+\mathcal{L}_{\text{CE}}^{total}$.

Dataset	Method	Acc	Acc Up
Cifar100 (FP:70.33)	GDFQ[42]	63.58	-
	GDFQ[42]+$\mathcal{L}_{\text{CE}}^{total}$	**64.01**	**0.43** ↑
	IntraQ[47]	64.98	-
	IntraQ[47]+$\mathcal{L}_{\text{CE}}^{total}$	**65.49**	**0.51** ↑
	Ours	**66.94**	-
ImageNet (FP:73.09)	GDFQ[42]	60.60	-
	GDFQ[42]+$\mathcal{L}_{\text{CE}}^{total}$	**61.36**	**0.76** ↑
	IntraQ[47]	66.47	-
	IntraQ[47]+$\mathcal{L}_{\text{CE}}^{total}$	**66.79**	**0.32** ↑
	Ours	**67.02**	-

(*i.e.* comparing $\mathcal{N}_2$ with $\mathcal{N}_3$, comparing $\mathcal{N}_5$ with $\mathcal{N}_7$, and comparing $\mathcal{N}_6$ with $\mathcal{N}_7$), the network trained using multiple pseudo-labels can have better performance than using only one of them, which shows the designed multiple pseudo-labels works and can provide more supervision information

for improving the performance of the quantized network. Especially, the performance of $\mathcal{N}_5$ is higher than that of $\mathcal{N}_6$, which can show better training efficiency using high-reliable data than low-reliable data. *(2) Mean Squared Error (MSE)* can make significant performance gain for quantized network. In the comparison with and w/o $\mathcal{L}_{\text{MSE}}$ (*i.e.* comparing $\mathcal{N}_1$ with $\mathcal{N}_4$, comparing $\mathcal{N}_2$ with $\mathcal{N}_5$, and comparing $\mathcal{N}_3$ with $\mathcal{N}_7$), the performance can have a stable improvement with $\mathcal{L}_{\text{MSE}}$, which can prove the efficiency for minimizing the difference between the intermediate feature layers of the two networks.

To give visualizations for the improvement of network performance with the designed loss function, the feature distributions of the network $\mathcal{N}_4$ (w/o $\mathcal{L}^h_{\text{CE}}$ and $\mathcal{L}^l_{\text{CE}}$), $\mathcal{N}_5$ (w/o $\mathcal{L}^l_{\text{CE}}$), $\mathcal{N}_6$ (w/o $\mathcal{L}^h_{\text{CE}}$) and $\mathcal{N}_7$ (Ours) on the original test dataset are shown in Figure 4 (a$\sim$d) with t-SNE [40], which can display the distribution for classification by transferring data from high dimension into the two-dimensional space. It is evident that the designed $\mathcal{L}^h_{\text{CE}}$ and $\mathcal{L}^l_{\text{CE}}$ can improve the classification performance of the trained quantized network, which can be observed from the aggregation effect of different classes (comparing $\mathcal{N}_4$ with $\mathcal{N}_5$ in Figure 4 (a) and (b), and comparing $\mathcal{N}_4$ with $\mathcal{N}_6$ in Figure 4 (a) and (c)), so the designed reliability filtering can improve the training of quantized network. In addition, $\mathcal{N}_7$ in Figure 4 (d) has the best aggregation effect among these four networks, which can prove the efficiency of the designed multiple pseudo-labels in the proposed method.

Effect of Reliability Filtering. To show the efficiency of proposed self-entropy metric for evaluating the reliability of samples in the synthetic data, we present the feature distributions of the pre-trained network on the original training dataset, the synthetic dataset, the evaluated high-reliable dataset $\hat{X}^h$ and low-reliable dataset $\hat{X}^l$ as shown in Figure 4 (e $\sim$ h). The feature distributions of the pre-trained network on the original training dataset are shown in Figure 4 (e), which means the pre-trained network can have a good classification performance on the original training dataset. However, the feature distributions in Figure 4 (f) show the pre-trained network can hardly efficiently classify the synthetic data. From the results in Figure 4 (g) and (h), it is evident that the pre-trained network can have an excellent classification performance on $\hat{X}^h$. This means the application of designed reliability filtering can efficiently filter the high-reliable samples, which is suitable to provide supervision information and train quantized network.

Effect of Multiple Pseudo-Labels. The highest and secondary highest predicted probabilities of pre-trained network on the high- and low-reliable datasets are plotted in Figure 5. It can be observed that the highest predicted probabilities on the high-reliable dataset are far higher than the secondary highest predicted probabilities, since the pre-trained network has high reliability on these data. While on the low-reliable dataset, the pre-trained network cannot perform well, which can be reflected by the lower highest predicted probabilities and higher secondary highest predicted probabilities. Therefore, the designed secondary label $\hat{y}^s_i$ can also provide information for fitting the performance of the pre-trained network, which can enhance the training and mitigate the risk of misleading for the high-reliable samples.

Effect of Pseudo-Label Training. Our designed multiple pseudo-labels $\mathcal{L}^{total}_{\text{CE}}$ can also combined with other training framework. To evaluate the proposed training framework, two classic DFQ methods (GDFQ and IntraQ) are selected as baselines, and we combine the designed $\mathcal{L}^{total}_{\text{CE}}$ with these two baselines. We examine the performance on CIFAR-100 for 4-bit ResNet-20 and ImageNet for 4-bit ResNet-18 as listed in Table 6. From the experimental results, the combination with our designed $\mathcal{L}^{total}_{\text{CE}}$ can efficiently improve the performance of the quantized network for the existing methods, which can prove the effectiveness of our designed multiple pseudo-labels. In addition, it is also observed that the proposed training framework can still have the best accuracy among these methods, which can prove that the proposed pseudo-label training can guide the quantized network to have a similar prediction ability with the pre-trained network and learn supervision information from the synthetic data with different labels.

5. Conclusion

This work proposes an efficient data-free quantization method via pseudo-label filtering, which is the first to evaluate the synthetic data before training the quantized network. In the proposed method, self-entropy is selected as an evaluation metric to divide the synthetic data into high-reliable and low-reliable data. The multiple pseudo-labels are designed to label the evaluated samples, which can further improve the reliability and avoid misleading caused by low-reliable data. The pseudo-label training is designed to integrate the supervision information provided by multiple pseudo-labels. Extensive experiments are implemented and evaluated on CIFAR-10/100 and ImageNet datasets, demonstrating that the proposed framework performs better than existing methods.

Acknowledgement. This work is supported in part by the National Key R&D Program of China (No. 2022ZD0118201), the National Natural Science Foundation of China (No.61802105, 62272144, 72188101, 62020106007, and U20A20183), the University Synergy Innovation Program of Anhui Province (No. GXXT-2021-005 and GXXT-2022–033), the Fundamental Research Funds for the Central Universities (No. JZ2022HGTB0250 and PA2023IISL0096), and the Major Project of Anhui Province (202203a05020011).

References

[1] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Aciq: Analytical clipping for integer quantization of neural networks. *International Conference on Learning Representations*, 2019. 1, 2

[2] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. 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*, pages 696–697, 2020. 1, 2

[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. 1, 3, 5, 6

[4] Ting-An Chen, De-Nian Yang, and Ming-Syan Chen. Alignq: Alignment quantization with admm-based correlation preservation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2022. 1, 2

[5] Kanghyun Choi, Deokki Hong, Noseong Park, Youngsok Kim, and Jinho Lee. Qimera: Data-free quantization with synthetic boundary supporting samples. *Advances in Neural Information Processing Systems*, 34:14835–14847, 2021. 1, 3, 5, 6

[6] Kanghyun Choi, Hye Yoon Lee, Deokki Hong, Joonsang Yu, Noseong Park, Youngsok Kim, and Jinho Lee. It's all in the teacher: Zero-shot quantization brought closer to the teacher. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8311–8321, 2022. 5, 6

[7] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53:5113–5155, 2020. 1

[8] 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. 5

[9] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020. 1

[10] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020. 1, 2

[11] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4852–4861, 2019. 1, 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[13] Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. Squant: On-the-fly data-free quantization via diagonal hessian approximation. In *International Conference on Learning Representations*, 2021. 2, 6

[14] Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. Squant: On-the-fly data-free quantization via diagonal hessian approximation. *arXiv preprint arXiv:2202.07471*, 2022. 3

[15] Tiantian Han, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Improving low-precision network quantization via bin regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5261–5270, 2021. 1

[16] 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. 3, 6

[17] 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. 5

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 5

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. pmlr, 2015. 1

[21] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 1

[22] Yongkweon Jeon, Chungman Lee, and Ho-young Kim. Genie: Show me the data for quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12064–12073, 2023. 1

[23] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, 2(6):508–518, 2021. 2, 4

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[27] Jung Hyun Lee, Jihun Yun, Sung Ju Hwang, and Eunho Yang. Cluster-promoting quantization with bit-drop for minimizing network quantization loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5370–5379, 2021. 1

[28] Huantong Li, Xiangmiao Wu, Fanbing Lv, Daihai Liao, Thomas H Li, Yonggang Zhang, Bo Han, and Mingkui Tan. Hard sample matters a lot in zero-shot quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24417–24426, 2023. 1, 3, 5, 6

[29] Xinhao Li, Jingjing Li, Lei Zhu, Guoqing Wang, and Zi Huang. Imbalanced source-free domain adaptation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3330–3339, 2021. 4

[30] Yuhang Li, Feng Zhu, Ruihao Gong, Mingzhu Shen, Xin Dong, Fengwei Yu, Shaoqing Lu, and Shi Gu. Mixmix: All you need for data-free compression are feature and data mixing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4410–4419, 2021. 1, 3

[31] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1504–1512, 2023. 1

[32] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1512–1521, 2021. 3, 5, 6

[33] Vasco Lopes, Fabio Maria Carlucci, Pedro M Esperança, Marco Singh, Antoine Yang, Victor Gabillon, Hang Xu, Zewei Chen, and Jun Wang. Manas: Multi-agent neural architecture search. *Machine Learning*, pages 1–24, 2023. 1

[34] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2019. 2, 3

[35] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206, 2020. 1, 2

[36] Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim, Jeremy Fowers, Karin Strauss, and Eric S Chung. Accelerating deep convolutional neural networks using specialized hardware. *Microsoft Research Whitepaper*, 2(11):1–4, 2015. 1

[37] Biao Qian, Yang Wang, Richang Hong, and Meng Wang. Adaptive data-free quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7960–7968, 2023. 1, 3, 5, 6

[38] Biao Qian, Yang Wang, Richang Hong, and Meng Wang. Rethinking data-free quantization as a zero-sum game. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 3, 5, 6

[39] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021. 1

[40] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[41] Wenxiao Wang, Minghao Chen, Shuai Zhao, Long Chen, Jinming Hu, Haifeng Liu, Deng Cai, Xiaofei He, and Wei Liu. Accelerate cnns from three dimensions: A comprehensive pruning framework. In *International Conference on Machine Learning*, pages 10717–10726. PMLR, 2021. 1

[42] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 1–17. Springer, 2020. 1, 3, 5, 6, 7

[43] Edouard Yvinec, Arnaud Dapogny, Matthieu Cord, and Kevin Bailly. Spiq: Data-free per-channel static input quantization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3869–3878, 2023. 3

[44] Xiangguo Zhang, Haotong Qin, Yifu Ding, Ruihao Gong, Qinghua Yan, Renshuai Tao, Yuhang Li, Fengwei Yu, and Xianglong Liu. Diversifying sample generation for accurate data-free quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15658–15667, 2021. 3, 5, 6

[45] Yuyao Zhang and Nikolaos M Freris. Adaptive filter pruning via sensitivity feedback. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1

[46] Yunshan Zhong, Mingbao Lin, Mengzhao Chen, Ke Li, Yunhang Shen, Fei Chao, Yongjian Wu, and Rongrong Ji. Fine-grained data distribution alignment for post-training quantization. In *European Conference on Computer Vision*, pages 70–86. Springer, 2022. 1

[47] Yunshan Zhong, Mingbao Lin, Gongrui Nan, Jianzhuang Liu, Baochang Zhang, Yonghong Tian, and Rongrong Ji. Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12339–12348, 2022. 1, 3, 5, 6, 7

[48] Baozhou Zhu, Peter Hofstee, Johan Peltenburg, Jinho Lee, and Zaid Alars. Autorecon: Neural architecture search-based reconstruction for data-free compression. In *30th International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 3470–3476. International Joint Conferences on Artificial Intelligence, 2021. 1, 3, 5, 6