Generative Data Free Model Quantization With Knowledge Matching for Classification

Shoukai Xu, Shuhai Zhang, Jing Liu, Bohan Zhuang, Yaowei Wang, and Mingkui Tan, *Member, IEEE*

***Abstract*— Neural network quantization aims to reduce the model size, computational complexity, and memory consumption by mapping weights and activations from full-precision to low-precision. However, many existing quantization methods, either post-training with calibration or quantization-aware training with fine-tuning, require original data for better performance, which may not be available due to confidentiality or privacy constraints. This lack of data can lead to a significant decline in performance. In this paper, we propose a universal and effective method called Generative Data Free Model Quantization with Knowledge Matching for Classification (KMDFQ) that removes the dependence on data for neural network quantization. To achieve this, we propose a knowledge matching generator that produces meaningful fake data based on the latent knowledge in the pre-trained model, including classification boundary knowledge and data distribution information. Based on this generator, we propose a fake-data driven data free quantization method that uses the generated data to take advantage of the latent knowledge for quantization. Furthermore, we introduce Mean Square Error alignment during the fine-tuning of the quantized model to more strictly and directly learn knowledge, making it more suitable for data free quantization. Extensive experiments on image classification demonstrate the effectiveness of our method, achieving higher accuracy than existing data free quantization methods, particularly as the quantization bit decreases. For example, on ImageNet, the 4-bit data free quantized ResNet-18 has less than a 1.2% accuracy decline compared to quantization with real data. The source code is available at https://github.com/ZSHsh98/KMDFQ.**

***Index Terms*— Data privacy and security, model compression, data free quantization, data generation.**

I. Introduction

DEEP neural networks (DNNs) have shown promising results in the field of computer vision [2], [3], [4], [5], [6], [7], [8]. However, a large number of parameters and high computational cost of DNNs make them difficult to deploy on embedded or edge devices with limited computing, storage, and battery resources [9], [10], [11]. One way to reduce the model size and inference latency of DNNs is through network quantization [12], [13], [14], [15], [16], [17], [18], [19], [20], which involves quantizing floating-point values into lower precision (*e.g.*, 8-bit to 4-bit). This can help reduce the computational overhead of DNNs and make them more suitable for use on resource-constrained devices.

Existing normal quantification methods can be classified into two categories based on the training strategy: post-training quantization [16], [17], [20], [21], [22] and quantization aware training [12], [13], [14], [15]. The former uses the original data for calibration without fine-tuning, while the latter uses the original data to fine-tune the quantized model for better performance. However, both of these methods require training data, which may not be available in many real-world applications (*e.g.*, medical [23], [24], autopilot [25], [26], finance [27]) due to the data privacy and security concerns. In these scenarios, conventional quantization methods cannot be applied, as performing network quantization without data results in significant performance degradation. Therefore, it is of great practical value to consider an effective quantization method that does not require the original data, known as data free quantization, especially in the current era when the value of data is increasing and personal data privacy is becoming more important.

To address this issue, one possible approach is to directly sample random inputs from some distribution and use them to quantize the model so that the output distributions of the full-precision model and the quantized model are as close as possible. However, for a CNN model, its input space is extremely large, but the original training data only belongs to a small part of this space. Since random inputs can be far from the original training data distribution, they contain

Manuscript received 24 February 2023; revised 23 April 2023; accepted 9 May 2023. Date of publication 23 May 2023; date of current version 7 December 2023. This work was supported in part by the Key Realm Research and Development Program of Guangzhou under Grant 202007030007, in part by the Key-Area Research and Development Program Guangdong Province under Grant 2019B010155002, in part by the National Natural Science Foundation of China under Grant 62072190 and Grant U20B2052, and in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant 2017ZT07*X*183. An earlier version of this paper was presented in part at the European Conference on Computer Vision (ECCV), Glasgow, U.K., in August 2020 [DOI: 10.1007/978-3-030-58610-2_1]. This article was recommended by Associate Editor Z. Tang. *(Shoukai Xu and Shuhai Zhang contributed equally to this work.) (Corresponding authors: Mingkui Tan; Bohan Zhuang.)*

Shoukai Xu is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: sexsk@mail.scut.edu.cn).

Shuhai Zhang is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: mszhangshuhai@mail.scut.edu.cn).

Jing Liu and Bohan Zhuang are with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (e-mail: jing.liu1@monash.edu; Bohan.Zhuang@monash.edu).

Yaowei Wang is with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: wangyw@pcl.ac.cn).

Mingkui Tan is with the School of Software Engineering, and the Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, South China University of Technology, Guangzhou 510006, China (e-mail: mingkuitan@scut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3279281.

Digital Object Identifier 10.1109/TCSVT.2023.3279281

little semantic information for model quantization, resulting in significant performance degradation. Another approach is to use generative adversarial networks (GANs) [28] to obtain data. However, GANs require original training data in the discriminator to optimize the generator for semantically generated data. Since training data is not available, GANs cannot be applied in data free quantization either. Therefore, constructing meaningful data for model quantization is very challenging but crucial.

To generate meaningful data that can be used in quantization, it is important and necessary to exploit data information and latent knowledge from a pre-trained model. A recent study [29] showed that a well-trained over-parameterized model maintains sufficient information about the entire data set. However, it is still unclear what information exists, which information is helpful for quantization, and how to exploit such information.

In image classification tasks, one of the most important information is the classification boundary. Although the original data is missing, a pre-trained model can learn a classification boundary to divide data into different classes (see Fig. 1 (a)) and implicitly embed this knowledge in its parameters. Given only a pre-trained model without the training data, it is still possible to mine the classification boundary information to generate meaningful fake data. Additionally, batch normalization [31] is commonly used to stabilize the training process of a neural network. The statistics in batch normalization layers reflect the distribution of the original data and their features, which can be directly obtained from the pre-trained model. Furthermore, given input data, a model outputs a set of scores for each class after the classifier. These classification scores contain abundant representative information that can be transformed from the full-precision model to the quantized model. However, existing data free quantization methods disregard this information and only focus on a single sample [30] or network parameters [32]. For example, ZeroQ [30] ignores the class information of the original data and exploits the information of a single sample instead of the entire data, causing the constructed distribution to be far from the real data distribution (see Fig. 1 (c)). The question of how to construct meaningful data by effectively exploiting classification boundary knowledge, data distribution information, and latent model knowledge from a pre-trained model remains open. Moreover, most existing methods just focus on generating meaningful data, such as DSG [33] and ZAQ [34], while overlooking the critical role of knowledge matching in data free quantization. These methods often employ KL divergence for knowledge matching, but it may not be the appropriate choice for data free quantization. The question of how to perform more effective knowledge matching for data free quantization remains to be studied.

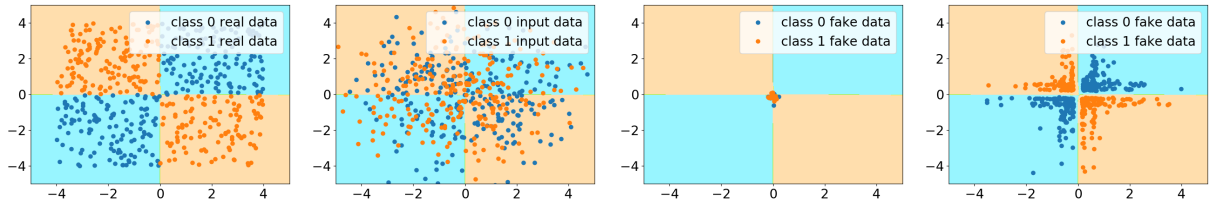
In this paper, we propose a universal and effective method called Generative Data Free Model Quantization with Knowledge Matching for Classification to perform network quantization without the original training data. To do this, we first learn a knowledge matching generator to produce meaningful data. The generator mines the classification boundary and distribution information of the original data from the pre-trained full-precision model. We then propose a fake-data-driven quantization scheme that can quantize a pre-trained model using the generated data and fine-tune the quantized model with the fixed batch normalization statistics (BNS) for more stable accuracy. Additionally, we introduce a Mean Square Error alignment to learn more knowledge directly from the pre-trained model, which significantly improves the performance of data free quantization.

Our main contributions are summarized as follows.

- We propose Generative Data Free Model Quantization with Knowledge Matching for Classification (KMDFQ) which performs quantization without any real data. Given just a pre-trained full-precision model, our method mines available latent information and uses it for model quantization.
- To generate more meaningful fake data, we design an effective knowledge matching generator to construct data using the classification boundary and data distribution information from the pre-trained full-precision model. We then introduce a fake-data driven quantization scheme to quantize and fine-tune the model by using the generated data and mining knowledge of the pre-trained model.
- To perform more effective knowledge alignment, we propose to use the MSE alignment to thoroughly transfer knowledge from the full-precision models to the quantized ones. The MSE alignment has a strict constraint which is superior for the data free situation and reasonable for the same architecture before and after quantization.
- Extensive experiments on CIFAR-10/100 and ImageNet demonstrate the effectiveness of our method compared with the existing data free quantization approaches. The accuracy of our data free method approximates that of quantization using the original data.

Compared to the conference version, this manuscript is extended with the following new contents:

1) We apply Mean Square Error alignment to effectively extract more knowledge from the pre-trained model. Compared to KL divergence in the preliminary version, the proposed MSE alignment method is able to achieve more complete logit matching, leading to a boost in accuracy. We conduct sufficient experiments to demonstrate that the MSE alignment is better for our task. To the best of our knowledge, our method is the first work to conduct a comparative analysis of KL and MSE in this context.
2) We theoretically derive a generalization bound of our proposed method, which not only helps us understand how feature alignment takes effect but also shows the superiority of the MSE alignment compared to the preliminary version with KL divergence.
3) We extensively investigate the effectiveness of our methods by conducting more ablative studies as well as more experiments on the hyper-parameters of the KMDFQ.



(a) Real data (b) Gaussian input data (c) Fake data (ZeroQ [30]) (d) Fake data (Ours)

Fig. 1. The comparisons of generated fake data. ZeroQ generates fake data by gradient updating; however, since it neglects the inter-class information, it is hard to capture the original data distribution. Meanwhile, our proposed knowledge matching generator (shown in Fig. 2 in Section IV-A) produces fake data with label distribution, and it is more likely to produce data distributed near the classifier boundaries.

4) We provide more experiments on classical compact models (*i.e.*, MobileNet and ShuffleNet) with more quantization bitwidths. Our proposed method achieves much higher accuracy than the existing data free quantization methods, especially in low-bitwidth scenarios.

II. Related Work

A. Data Driven Quantization

Model quantization aims to quantize weights, activations and even gradients to low-precision and yield highly compact models. As a result, the expensive multiplication operations can be replaced by efficient additions or bit-wise operations. Existing quantization method can be roughly categorized into binary neural networks (BNNs) [13], [14], [18], [35], [36] and fixed-point quantization [15], [17], [37], [38]. Moreover, quantization studies mainly focus on tackling two core bottlenecks, including designing accurate quantizers to fit the categorical distribution to the original continuous distribution [39], [40], [41], and approximating gradients of the non-differential quantizer during back-propagation [42], [43], [44]. In addition, the first practical 4-bit post-training quantization approach was introduced in [45]. To improve the performance of neural network quantization without retraining, outlier channel splitting (OCS) [21] proposed to move affected outliers toward the center of the distribution. Besides, many mainstream deep learning frameworks support 16-bit half-precision or 8-bit fixed-point quantization, such as TensorFlow [46], PyTorch [47], PaddleSlim, etc. In particular, these platforms provide both post-training quantization and quantization-aware training. However, all of them require original data. In short, whether in scientific studies or practical applications, if the original data are unavailable, it is hard for quantization methods to work properly. In contrast, our method is able to get rid of data dependence to achieve data free quantization.

B. Data Free Quantization

Quantization also faces the situation without original data while previous quantization methods generally need original data to improve their performance. However, in some instances, it is difficult to get the original data. Recently, some studies focused on data free model quantization. DFQ [32] argued that data free quantization performance could be improved by equalizing the weight ranges of each channel in the network and correcting biased quantization error. ZeroQ [30], a novel zero-shot quantization framework, enabled mixed-precision quantization without any access to the training or validation data. Based on ZeroQ, DSG [33] finds that the data generation constrained by batch normalization statistics suffers serious homogenization at both distribution and sample levels, which causes a significant performance drop of the quantized model. So DSG slacks the alignment constraint of BNS to at the distribution level and reinforces specific layers for different data samples. Based on GDFQ [1], ZAQ [34] proposes a two-level discrepancy modeling to synthesize informative data and an adversarial knowledge transfer to optimize the generator and the quantized model. However, these quantization methods work well for 8-bit but get poor performance in aggressively low bitwidth regimes such as 4-bit.

C. Data Free Knowledge Distillation

Knowledge Distillation (KD) [48] aims to transfer knowledge from a pre-trained larger model (teacher network) into a small model (student network). Different from model quantization, KD always employs the different network architectures for the teacher network and the student network. Due to data privacy or confidentiality concerns, recently, a few attempts [49], [50], [51], [52], [53], [54] have been studying data free KD when original training data are not accessible.

Lopes et al. [55] reconstructed a new data set based solely on the pre-trained model. The metadata recorded in the form of activation statistics and finally distilled the pre-trained "teacher" model into the smaller "student" network. Instead, Haroush et al. [49] proposed to use metadata (e.g., channel-wise mean and standard deviation) from the Batch Normalization (BN) [31], [56] layer for KD. Different from the above methods, Chen et al. [51] exploited generative adversarial networks to perform data free knowledge distillation. KEGNET [52] proposed a novel data free low-rank approximation approach assisted by knowledge distillation. In Adversarial Belief Matching [53], a generator generated data on which the student mismatched the teacher, and then the student network was trained using these data. Fang et al. [54] considered both the teacher and the student as the discriminator to reduce the discrepancy between them.

Constraining the BN statistics (BNS) of the generated data is widely employed in data free knowledge distillation [49], [50], which is consistent with the process of crafting the fake data in data free quantization. However, some techniques of KD cannot better adapt for the data free quantization such

as the feature alignment loss (*e.g.*, KL loss v.s. MSE loss), where the MSE has a more negligible accuracy degradation compared with the KL in data free quantization. This paper will give a further analysis for this in Section V.

III. Problem Definition

A. Data Free Quantization Problem

Quantization usually requires the original data for calibration or fine-tuning. Unfortunately, in many practical scenarios, the original data may not be available due to individual privacy, commercial value or even confidential issues. If someone needs to quantize a pre-trained model but cannot use any original data, the general scheme of the network quantization will lose efficacy and even fail to work completely, resulting in a quantized model with inferior performance. To solve this realistic difficult problem, a data free quantization method is necessary, especially the data free quantization-aware training method that is able to obtain better performance.

Given a model M with full-precision weights θ, data free quantization aims to construct fake data $(\hat{\mathbf{x}}, y)$, where $\hat{\mathbf{x}} \in \mathcal{X}$ is a generated fake sample, $y \in \mathcal{Y}$ is the assigned label, and meanwhile, obtain a model Q with low-bitwidth weights θ_q from the model M_θ. Furthermore, to compensate for the accuracy loss from quantization, data free quantization-aware training method fine-tunes the quantized model with the generated data by solving the following problem:

$$\min_{Q} \mathbb{E}_{\hat{\mathbf{x}},y}\left[\ell(Q(\hat{\mathbf{x}}), y)\right], \tag{1}$$

where $\ell(\cdot,\cdot)$ is some loss function, such as cross-entropy loss and mean squared error.

B. Challenges of Data Free Quantization

To address the data absence issue, one possible way to realize data free quantization is to construct fake data that is helpful for calibration and fine-tuning from a pre-trained full-precision model. Although the full-precision model may contain rich data information, such latent information alone is hard to exploit for recovering the original data. In practice, the performance of quantization highly depends on the quality of constructed data. Constructing meaningful data with the limited information of the pre-trained model is very challenging. The key issues to construct meaningful data are what information exists in the model, which information is helpful for quantization and how to exploit such information for data generation.

Recently, one data free quantization method (ZeroQ) [30] constructs fake data by using a linear operator with gradient update information. With the help of constructed data, the performance of the quantized models can be significantly improved. However, ZeroQ has insufficient information to improve the performance of quantization with the following two issues. First, it constructs fake data without considering label information, and thus neglects to exploit the classification boundary knowledge from the pre-trained model. Second, it enforces the batch normalization statistics of a single data point instead of the whole data, leading to being far away from the real data distribution. The classification boundary knowledge and data distribution information are valuable for data generation. They are hidden in the parameters of the pre-trained model so that we are able to capture them although the data are missing. As a result, one can construct a generator G to produce fake data by considering label and distribution information,

$$\hat{\mathbf{x}} = G(\mathbf{z}|y), \quad \mathbf{z} \sim p(\mathbf{z}), \tag{2}$$

where $\mathbf{z}$ is a random vector drawn from some prior distribution $p(\mathbf{z})$, *e.g.*, Gaussian distribution or uniform distribution, y is the label assigned to the generated data $\hat{\mathbf{x}}$. By using the generator, we are able to construct fake data to improve quantization performance. However, how to learn a good generator to exploit the classification boundary knowledge and data distribution information in the pre-trained model remains to be answered. Furthermore, existing methods primarily focus on generating meaningful data, such as ZeroQ [30], DSG [33] and ZAQ [34], while neglecting the crucial role of knowledge alignment in data-free quantization. These methods typically use KL divergence for knowledge alignment, which may not be suitable for data-free quantization. Therefore, there is a need for further investigation on how to improve the effectiveness of knowledge alignment in data-free quantization.

IV. Generative Data Free Model Quantization With Knowledge Matching

In many practical scenarios, training data are unavailable; thus, an existing method [30] performs data free quantization by constructing data with gradient update information. However, without efficiently exploiting knowledge from a pre-trained model, directly constructing data lacks important label and data distribution information. Thus, the synthetic data can be far away from the real data distribution (See Fig. 1 (c)) and affect the quantization performance. To address this issue, we aim to design a knowledge matching generator to obtain effective fake data. Specifically, we consider the classification boundary knowledge and data distribution information to train the generator. Then, we are able to perform fake-data driven quantization to obtain the quantized model and use knowledge distillation with Mean Square Error alignment to improve the performance of the quantized models. The overall framework is shown in Fig. 2.

A. Knowledge Matching Generator

When a data owner trains a deep neural network by the original training data, the network captures sufficient data information and learns latent data knowledge to make a decision [29]. In this sense, a pre-trained DNN itself contains some knowledge and information of the training data, *e.g.*, classification boundary knowledge and distribution information. We therefore propose a knowledge matching generator and elaborate how to effectively construct informative data from only the pre-trained model instead of the original data.

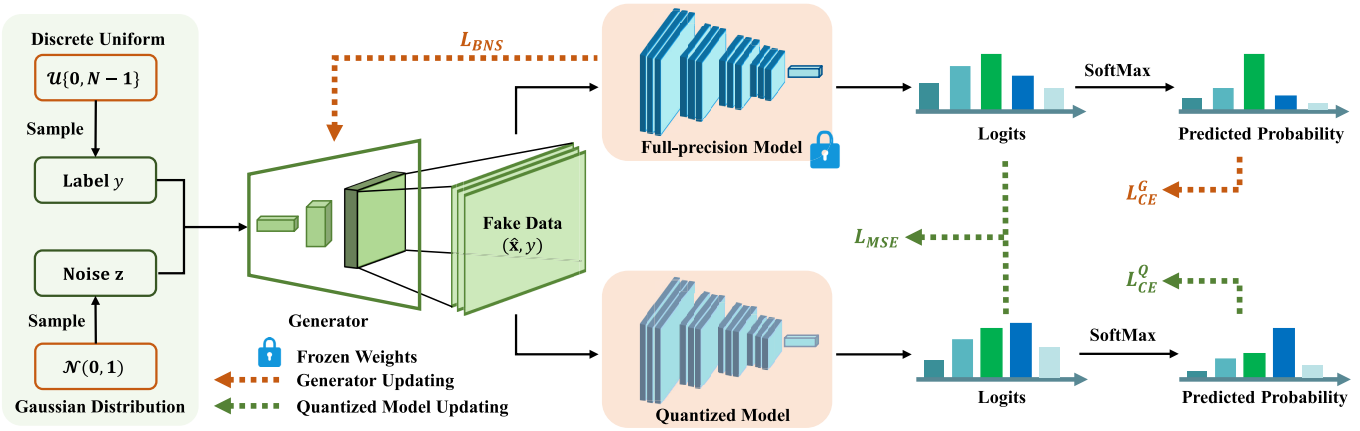


Fig. 2. An overview of the proposed method. Given Gaussian noise and the label as input, the generator creates fake data and feeds them into both the full-precision model and the quantized model. The fixed full-precision model provides knowledge for updating the generator. The quantized model learns latent knowledge from the generator and the full-precision model.

1) Classification Boundary Knowledge Matching: A pre-trained model contains valuable classification boundary knowledge, however, such information is difficult to be exploited to recover the data near the classification boundary. Recently, generative adversarial networks (GANs) [28], [57], [58] have achieved considerable success in producing data. Unfortunately, since the real data are unavailable, the discriminator in a GAN will lose efficacy. Without the discriminator, learning a generator to produce meaningful data is difficult. Therefore, the GAN methods cannot be directly used to generate effective data in the data free situations. To solve this issue, in this paper, we propose a knowledge matching generator to produce informative fake data merely from a pre-trained model for the data free quantization task.

First, the generator should match the classification boundary knowledge. As shown in Fig. 1 (a), different categories of data should be distributed in different data spaces. Although the original data cannot be observed in this task, we are able to easily confirm the number of categories C of original data by the classifier of a pre-trained model. In this way, we get the set of label space $y \in \{0, 1, \ldots, K-1\}$. To generate fake data, we introduce a noise vector $\mathbf{z}$ conditioned on an assigned label y. Here, we sample noise from a normal distribution and uniformly sample a label from $\{0, 1, \ldots, K-1\}$. Then, the generator maps a prior input noise vector $\mathbf{z}$ and the given label y to the fake data $\hat{\mathbf{x}}$. Here, y is assigned to $\hat{\mathbf{x}}$ as the ground truth label. Formally, we define the knowledge matching generator as follows:

$$\hat{\mathbf{x}} = G(\mathbf{z}|y), \quad \mathbf{z} \sim \mathcal{N}(0, 1). \tag{3}$$

The generated data $\hat{\mathbf{x}}$ is supposed to be close to the classification boundary so that it can be effective to fine-tune a quantized model to improve the quantization performance. To this end, we make the knowledge matching generator capture boundary information from the pre-trained model. Specifically, given a Gaussian noise $\mathbf{z}$ and the corresponding label y, the generator should generate fake data that is classified to the same label y by the full-precision pre-trained model M. Therefore, we introduce the following cross-entropy (CE) loss to train the generator G:

$$\hat{\mathbf{o}} = M(G(\mathbf{z}|y)) \in \mathbb{R}^K,$$
$$\mathcal{L}_{\text{CE}}^{G} = -\sum_{i=0}^{K-1} \mathbb{1}_{\{i=y\}} \log \frac{\exp(\hat{\mathbf{o}}_i)}{\sum_{j=0}^{K-1} \exp(\hat{\mathbf{o}}_j)}, \tag{4}$$

where $\hat{\mathbf{o}}$ is the logits of the pre-trained full-precision model and $\mathbb{1}_{\{\cdot\}}$ is an indicator function. By minimizing Eq. (4), we enable the generator G to match the classification boundary knowledge learned from the pre-trained model M.

2) Distribution Information Matching: The pre-trained model contains the data distribution information of training data which is also crucial for data generation. Such distribution information is included in the batch normalization (BN) layers [31] which are used to control the change of the distribution. Specifically, BN layers have batch normalization statistics (BNS), *i.e.*, the mean and variance. While training the full-precision model, the BNS are computed dynamically. For every batch, batch normalization layers only compute statistics on the current mini-batch inputs and accumulate them with momentum. Finally, the exponential moving average (EMA) batch normalization statistics will be obtained and then they will be frozen in network validation and test.

Obviously, the data distribution information in the BNS can be learned by the generator although the original data are absent. To retain the BNS information, the mean and variance of the generated distribution of the faked data should be the same as those of the real data distribution. To this end, we learn our knowledge matching generator G using $\mathcal{L}_{\text{BNS}}$:

$$\mathcal{L}_{\text{BNS}} = \sum_{l=1}^{L} \|\boldsymbol{\mu}_l^r - \boldsymbol{\mu}_l\|_2^2 + \|\boldsymbol{\sigma}_l^r - \boldsymbol{\sigma}_l\|_2^2, \tag{5}$$

where $\boldsymbol{\mu}_l^r$ and $\boldsymbol{\sigma}_l^r$ are the mean and variance of the fake data's distribution at the l-th BN layer, respectively, and $\boldsymbol{\mu}_l$ and $\boldsymbol{\sigma}_l$ are the corresponding mean and variance stored in the l-th BN layer of the pre-trained full-precision model, respectively. In this way, we are able to learn a good generator to keep the distribution information from the training data.

3) Training Generator G: First, we randomly sample a batch of noise vectors $\{\mathbf{z}_0, \ldots, \mathbf{z}_{B-1}\}$ from a Gaussian distribution $\mathcal{N}(0, 1)$ with labels $\{y_0, \ldots, y_{B-1}\}$ from $\{0, 1, \ldots,$

$K-1\}$. Here, B is the batch size and K is the number of classes. Then we use G to generate fake data from the distribution and input them into the pre-trained model. Finally, we get the predictions and BNS parameters to train the generator. The final generator loss $\mathcal{L}_G$ is formulated as follows:

$$\mathcal{L}_G = \mathbb{E}\left[\mathcal{L}_{\text{CE}}^G\right] + \beta \mathcal{L}_{\text{BNS}}, \tag{6}$$

where β is a trade-off parameter.

B. Fake-Data Driven Quantization

With the help of the proposed knowledge matching generator, we are able to obtain meaningful fake data. Then we quantize a model using the generated data and proceed with the quantization-aware training. Although the generated data capture some knowledge from the pre-trained model, fine-tuning an accurate quantized model is difficult. To address this, we introduce a fake-data driven quantization method and solve the optimization problem by transferring the knowledge from the pre-trained model.

1) Quantization: To simplify networks, quantization compresses the weights and activations of neural networks from continuous value to discrete fixed-point integer. Here, we use the linear quantization [59] to quantize both weights and activations of each layer. The quantization operation for activations is the same as it for weights.

Specifically, taking weights as an example, given the full-precision weights θ, we first pick the minimum l and maximum u of θ across each layer. According to the quantization bit-width b, we calculate Δ and η as follows:

$$\Delta = \frac{2^b - 1}{u - l}, \tag{7}$$

$$\eta = l \cdot \Delta + 2^{b-1}. \tag{8}$$

Then, we get the discrete weights θ' by the linear quantization:

$$\theta' = \lfloor \Delta \cdot \theta - \eta \rceil, \tag{9}$$

where $\lfloor \cdot \rceil$ is the round function and η is the zero point of quantization [59]. At last, we obtain the final quantized weights θ_q by clipping the θ' to the b-bit range $[-2^{b-1}, 2^{b-1}-1]$. Since Eq. (9) is not differentiable, we use the straight through estimator (STE) [60] to estimate the quantized model.

2) Fake-Data Driven Fine-tuning With Fixed BNS: To maintain the data distribution information contained in the pre-trained model, we directly use the batch normalization statistics (BNS) of the pre-trained model and fix them in the quantized model during fine-tuning. In this way, the BNS in the quantized model are corresponding to the statistics of the real data distribution. With the help of fixed BNS, the quantized model always inherits the real data distribution information to improve quantization performance, meanwhile, the training process is more stable (refer to Sec.VI-D.4).

Quantization may result in severe performance degradation when the real training data are unavailable. To address this issue, we aim to train the quantized model to approximate the full-precision model based on the generated data through the fine-tuning process. To this end, given the same input fake data, a well fine-tuned quantized model Q should have similar classification ability compared with the pre-trained model M, *i.e.*, Q should classify the fake data correctly. Specifically, we use the cross-entropy loss function to update Q:

$$\bar{\mathbf{o}} = Q(\hat{\mathbf{x}}) \in \mathbb{R}^K,$$
$$\mathcal{L}_{\text{CE}}^Q = -\sum_{i=0}^{K-1} \mathbb{1}_{\{i=y\}} \log \frac{\exp(\bar{\mathbf{o}}_i)}{\sum_{j=0}^{K-1} \exp(\bar{\mathbf{o}}_j)}, \tag{10}$$

where $\bar{\mathbf{o}}$ is the logits of the quantized model. By minimizing Eq. (10), the quantized model can be trained with the generated data to perform multi-class classification.

After all, the generated data are fake. The gap between the fake data and the original data cannot be ignored so that the traditional fine-tuning process with a common classification loss function is insufficient. Fortunately, with the help of fake data, we are able to apply knowledge distillation to improve the quantization performance. Specifically, given the same inputs, the outputs of a quantized model and full-precision model should be close enough to guarantee that the quantized model is able to achieve near performance compared with the full-precision model.

Quantization task uses the full-precision model as teacher and the quantized model as student. Different from the knowledge distillation between models with different architectures, the student model is a quantized version of a teacher network in which the model architecture is preserved. Therefore, we consider using a more strict measure metric, *i.e.*, mean square error alignment to push the quantized model similar to the full-precision model and recover the performance of the quantized model, which is reasonable based on the same architectures. Specifically, the quantized model can be fine-tuned using the following Mean Squared Error (MSE) loss function:

$$\mathcal{L}_{\text{MSE}} = \|\bar{\mathbf{o}} - \hat{\mathbf{o}}\|_2^2, \tag{11}$$

where $\hat{\mathbf{o}}$ is the logits of the pre-trained full-precision model. By optimizing the loss in Eq. (11), the quantized model can learn more knowledge from the full-precision model, which plays an important role to improve the performance of a quantized model.

3) Training Quantized Model Q: We first quantize the model according to Eq. (9). Then, we replace the BNS in the quantized model with the fixed batch normalization statistics (FBNS) as described in Section IV-B.2. So far, the quantized model has inherited the data distribution information contained in BNS and a part of latent knowledge from the parameters of the pre-trained model. Based on G, we obtain fake samples and use them to fine-tune the quantized model Q. The final quantized model loss function $\mathcal{L}_Q$ is formulated as follows:

$$\mathcal{L}_Q = \mathbb{E}\left[\mathcal{L}_{\text{CE}}^Q + \gamma \mathcal{L}_{\text{MSE}}\right], \tag{12}$$

where γ is a trade-off parameter. Note that we keep the pre-trained full-precision model fixed at all times.

Algorithm 1 Generative Data Free Model Quantization with Knowledge Matching for Classification

Input: A pre-trained full-precision model M.
Output: A quantized model Q and a knowledge matched generator G.

Update G several times as a warm-up process.
Quantize model M using Eq. (9), get quantized model Q.
Fix the batch normalization statistics for all layers in Q.
for $t = 1, \ldots, T_f$ **do**
  Obtain random noise $\mathbf{z} \sim \mathcal{N}(0, 1)$ and label y.
  Generate fake image $\hat{\mathbf{x}}$ using Eq. (3).
  Update generator G by minimizing Loss (6).
  Update quantized model Q by minimizing Loss (12).
end for

C. Training Strategy

The overall training algorithm of our proposed KMDFQ is shown in Algorithm 1. In the fine-tuning process, we alternately optimize the generator G and the quantized model Q in every epoch. In our alternating training strategy, the generator is able to generate different data with each update. By increasing the diversity of data, the quantized model Q can be trained to improve the performance. In addition, to make the fine-tuning of Q more stable, we firstly train G solely several times as a warm-up process. We do not stop updating G because if we have a better G, the fake data will be more similar to real training data and the upper limit of optimizing Q will be improved. The overall process is shown in Algorithm 1. In contrast, one can train the generator G until convergence, and then fix the generator and separately train the quantized model Q. However, in this separate training strategy, Q can only use the generated data of a fixed G so the data diversity is limited, which limits the performance of the quantized model. The further experiments are conducted in Table XII.

V. Difference Between Kullback-Leibler Divergence and Mean Squared Error Loss

Kullback-Leibler (KL) Divergence is the loss function commonly used in the knowledge distillation tasks [48]. Given a sample $\mathbf{x}$, the KL loss function for distillation is defined as

$$\mathcal{L}_{\mathrm{KL}}\left(\mathbf{p}^s(\tau), \mathbf{p}^t(\tau)\right) = \tau^2 \sum_{i=0}^{K-1} \mathbf{p}_i^t(\tau) \log \frac{\mathbf{p}_i^t(\tau)}{\mathbf{p}_i^s(\tau)}, \tag{13}$$

where τ is a temperature parameter, $\mathbf{p}_i(\tau) = \frac{\exp(\mathbf{o}_i/\tau)}{\sum_{j=0}^{K-1}\exp(\mathbf{o}_j/\tau)}$, $\mathbf{p}^s(\tau)$ and $\mathbf{p}^t(\tau)$ denote the probability distributions of the outputs of the student and teacher model, respectively.

KL divergence measures the difference of two probability distributions, promoting the student network learning the classification probability distribution of the teacher network. Based on the pre-trained teacher model and the real training data, KL divergence is able to successfully transfer the knowledge [48]. However, it is more difficult to transfer the knowledge when the real training data are unavailable. In this case, we can only fine-tune the model using the generated fake data. When the inputs are fake data, only learning the classification probability is not effective enough because of the distribution shift caused by the softmax function, which is formulated as below.

Proposition 1 (Distribution Shift): The probability distributions after employing the softmax operation on logits $\mathbf{o}$ *and* $\mathbf{o} + \epsilon (\forall \epsilon \in \mathbb{R})$ *are the same.*

Note that for all $\epsilon \in \mathbb{R}$, we have

$$\begin{aligned} \mathbf{p}_i &= \frac{\exp(\mathbf{o}_i)}{\sum_{j=0}^{K-1}\exp(\mathbf{o}_j)} \\ &= \frac{\exp(\mathbf{o}_i + \epsilon)}{\sum_{j=0}^{K-1}\exp(\mathbf{o}_j + \epsilon)}. \end{aligned} \tag{14}$$

Eq. (14) indicates that any bias ϵ added on the logits $\mathbf{o}$ leads to the same probability distribution when enforcing the softmax function on $\mathbf{o}$. This suggests those in the KL loss function Eq. (13), even $\mathcal{L}_{\mathrm{KL}}\left(\mathbf{p}^s(\tau), \mathbf{p}^t(\tau)\right)$ reaches the minimum, *i.e.*, $\mathbf{p}^s(\tau) = \mathbf{p}^t(\tau)$, the logits of the student model can be still far away from that of the teacher model, coinciding with the results in Fig. 3. On the contrary, the MSE loss is able to directly minimize the ℓ_2 distance between the logits of the student and teacher model, which leads to a satisfactory optimization solution for the data free quantization task than that with the KL divergence.

Next, we analyze the generalization bound for the proposed MSE alignment method. We use the generalization error of quantizing the model with MSE to measure how accurately the quantized model predicts for the unseen test data. To this end, following [61], we introduce the definition of Rademacher complexity to analyze our proposed method.

Definition 1 (Rademacher Complexity): Given an underlying distribution $\mathcal{D}$ *and a sample set* $\mathcal{S} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ *with fixed size* N*, where* $\mathbf{d}_i = (\hat{\mathbf{x}}_i, y_i)$*. Then the Rademacher complexity of our quantization paradigm in Eq. (12) is defined as*:

$$\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}}) = \mathbb{E}_{\mathcal{S}\sim\mathcal{D}}\left[\widehat{\mathbf{R}}_{\mathcal{S}}(\Phi_Q^{\mathrm{MSE}})\right], \tag{15}$$

where Φ_Q^{MSE} *is the function space of* $\mathcal{L}_Q$ *and* $\hat{\mathbf{R}}_{\mathcal{S}}(\Phi_Q^{\mathrm{MSE}})$ *is its empirical Rademacher complexity, which is defined as*:

$$\hat{\mathbf{R}}_{\mathcal{S}}(\Phi_Q^{\mathrm{MSE}}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup \frac{1}{N}\sum_{i=1}^{N}\sigma_i\left(\mathcal{L}_{\mathrm{CE}}^Q + \gamma\mathcal{L}_{\mathrm{MSE}}\Big|_{\mathbf{d}_i}\right)\right], \tag{16}$$

where $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N]$ *are independent uniform random variables valued in* $\{-1, +1\}$.

From Definition 1, the Rademacher complexity captures the capacity of a family of functions by measuring the degree they fit random noises [61]. Let $\widehat{\mathcal{L}}_Q$ be the empirical loss of $\mathcal{L}_Q$, based on Definition 1, we obtain the following generalization bound of our method.

Theorem 1 (Generalization Performance of KMDF): Let $\mathcal{L}_{\mathrm{CE}}^Q + \gamma\mathcal{L}_{\mathrm{MSE}}$ *be a mapping from* $\mathcal{X}\times\mathcal{Y}$ *to* $[0, U]$ *with upper bound* U *and the function space* Φ_Q^{MSE} *be infinite. For any* $\delta > 0$*, all* $\mathcal{L}_Q \in \Phi_Q^{\mathrm{MSE}}$*, with probability at least* $1-\delta$*, the*

generalization error $\mathcal{L}_Q$, i.e., the expected loss, satisfies

$$\mathcal{L}_Q \leq \widehat{\mathcal{L}}_Q + 2\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}}) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}, \tag{17}$$

where $\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}})$ is the Rademacher complexity of our proposed quantized model training paradigm based on the MSE loss. Let $\mathcal{B}_{\mathrm{MSE}}$ be the generalization bound w.r.t. $\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}})$, i.e., $\mathcal{B}_{\mathrm{MSE}}=2\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}})+U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}$, we thus have

$$\mathcal{B}_{\mathrm{MSE}} \leq \mathcal{B}_{KL}, \tag{18}$$

where $\mathcal{B}_{KL}$ is the generalization bound of GDFQ [1] w.r.t. the Rademacher complexity $\mathbf{R}_N(\Phi_Q^{KL})$.

Proof: For any sample $\mathcal{S} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ and any $\mathcal{L}_Q \in \Phi_Q^{\mathrm{MSE}}$, let $\widehat{\mathcal{L}}_Q^{\mathcal{S}}$ be the empirical loss of $\mathcal{L}_Q$ over $\mathcal{S}$. Applying McDiarmid's inequality [62] to function Ψ defined for any sample $\mathcal{S}$ by

$$\Psi(\mathcal{S}) = \sup_{\mathcal{L}_Q \in \Phi_Q^{\mathrm{MSE}}} \mathcal{L}_Q - \widehat{\mathcal{L}}_Q^{\mathcal{S}}. \tag{19}$$

Based on McDiarmid's inequality [62] and Theorem 3.1 in [61], for any $\delta > 0$, with probability at least $1-\delta$, we have

$$\begin{aligned}\Psi(\mathcal{S}) &\leq E_{\mathcal{S}}[\Psi(\mathcal{S})] + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)} \\ &\leq 2\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}}) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}.\end{aligned} \tag{20}$$

Combining Equation (19) and Inequalities (20), we have

$$\mathcal{L}_Q \leq \widehat{\mathcal{L}}_Q + 2\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}}) + U\sqrt{\frac{1}{2N}\log\left(\frac{1}{\delta}\right)}, \tag{21}$$

with probability at least $1-\delta$. Based on the definition of the Rademacher complexity, we have

$$\mathbf{R}_N(\Phi_Q^{\mathrm{MSE}}) \leq \mathbf{R}_N(\Phi_Q^{\mathrm{KL}}), \tag{22}$$

since the capacity of the function space Φ_Q^{MSE} is smaller than that of Φ_Q^{KL} due to the arbitrary property of ϵ in Proposition 1. With the same number of training samples, we thus have

$$\mathcal{B}_{\mathrm{MSE}} \leq \mathcal{B}_{\mathrm{KL}}. \tag{23}$$

■

This theorem shows the generalization bound of our training quantized model paradigm with the MSE loss, which relies on the Rademacher complexity of a function space Φ_Q^{MSE}. From Theorem 1, quantized paradigm with the MSE loss has a smaller generalization bound than that with the KL loss, which helps to achieve more accurate classification predictions. Note that in comparison to KL, MSE does not introduce additional computational complexity. The computational complexity of calculating these two losses is solely dependent on the output dimension d of the model, and the time complexity for both is $O(d)$.

The structures of the models remain the same before and after quantization. The quantized model should be as similar as possible to the full-precision model. Due to the distribution shift (Eq. (14)), the distribution of logits of the student can be still far away from that of the teacher model even if the KL divergence reaches the minimum. If we could not take full advantage of the logits information in the teacher model, we would only obtain an under-performing model. Especially for the data free situation, the knowledge from the teacher is particularly crucial because the information and knowledge of real data are unavailable. So we introduce the MSE alignment to solve the distribution shift problem. The logits distribution of the student is closer to that of the teacher model when the MSE distance reaches the minimum than KL distance. In addition to theoretical analysis, we also design experiments in Section VI-C and Section VI-D.2 to verify the superior of MSE alignment in data free quantization tasks. In brief, MSE alignment is superior to KL alignment for the data free quantization.

VI. Experiments

A. Data Sets and Implementation Details

We evaluate the proposed method on well-known data sets including CIFAR-10 [63], CIFAR-100 [63], and ImageNet [64]. CIFAR-10 consists of 60k images from 10 categories, with 6k images per category. There are 50k images for training and 10k images for testing. CIFAR-100 has 100 classes and each class contains 500 training images and 100 testing images. ImageNet is one of the most challenging and largest benchmark data sets for image classification, which has around 1.2 million real-world images for training and 50k images for validation.

Based on the full-precision pre-trained models from pytorchcv,[1] we quantize ResNet-20 [65] on CIFAR-10/100 and ResNet-18 [65], BN-VGG-16 [66], Inception v3 [67], MobileNet v2 [68] and ShuffleNet [69] on ImageNet. In all experiments, we quantize all layers including the first and last layers of the network following [30] and the activation clipping values are per-layer granularity. All implementations are based on PyTorch. Our experiments can be conducted using only one GPU NVIDIA 3090.

For CIFAR-10/100, we construct the generator following ACGAN [70] and the dimension of noise is 100. During training, we optimize the generator and quantized model using Adam [71] and SGD with Nesterov [72] respectively, where the momentum term and weight decay in Nesterov are set to 0.9 and 1×10^{-4}. Moreover, the learning rates of quantized models and generators are initialized to 1×10^{-4} and 1×10^{-3}, respectively. Both of them are decayed by 0.1 for every 100 epochs. In addition, we train the generator and quantized model for 400 epochs with 200 iterations per epoch. To obtain a more stable clip range for activation, we calculate the moving average of activation's range in the first four epochs without updating the quantized models and then fix this range for subsequent training. For $\mathcal{L}_G$ and $\mathcal{L}_Q$, we set $\beta = 1$ and $\gamma = 1$ after a simple grid search. For ImageNet, we replace the generator's standard batch normalization layer with the categorical conditional batch normalization layer for fusing

[1] https://pypi.org/project/pytorchcv/

TABLE I
COMPARISONS ON CIFAR FOR RESNET-20

Data Set	FP32	Method	Top-1 Accuracy (%)			
			W8A8	W6A6	W5A5	W4A4
Cifar-10	93.89	FT	94.20	94.09	93.82	92.93
		FT+Dist	94.24	94.14	93.82	92.99
		DFQ	93.70	92.30	83.85	11.12
		ZeroQ	93.96	93.06	90.08	71.95
		GDFQ	94.02	93.79	93.35	90.14
		Ours	**94.05**	**93.94**	**93.67**	**92.24**
Cifar-100	70.33	FT	70.69	70.61	69.89	68.52
		FT+Dist	70.79	70.55	69.86	68.76
		DFQ	69.99	66.86	47.43	17.08
		ZeroQ	70.20	68.52	64.36	44.85
		GDFQ	68.37	68.47	67.94	63.89
		Ours	**70.53**	**70.35**	**69.68**	**67.15**

TABLE II
COMPARISONS ON IMAGENET

Model	Method	Top-1 Accuracy (%)			
		W8A8	W6A6	W5A5	W4A4
ResNet-18 (71.58)	FT	71.85	70.49	70.21	64.88
	FT+Dist	71.89	71.53	70.25	65.54
	DFQ	69.36	56.64	0.15	0.10
	ZeroQ	**71.48**	70.05	62.51	24.77
	GDFQ	68.85	68.51	67.64	60.8
	Ours	70.73	**70.61**	**69.93**	**64.39**
BN-VGG16 (74.38)	FT	74.68	74.36	73.23	67.94
	FT+Dist	74.68	74.37	73.23	68.00
	DFQ	73.76	61.60	1.77	0.10
	ZeroQ	**73.96**	65.12	43.26	0.19
	GDFQ	71.80	71.53	71.48	68.09
	Ours	72.33	**72.29**	**71.89**	**68.79**
Inception v3 (77.63)	FT	77.70	77.48	76.45	71.10
	FT+Dist	77.70	77.47	76.42	71.57
	DFQ	77.54	75.98	59.44	0.94
	ZeroQ	**78.75**	73.66	68.17	18.55
	GDFQ	76.20	75.96	75.08	68.40
	Ours	76.45	**76.43**	**75.54**	**71.22**
MobileNet v2 (73.08)	FT	73.09	72.78	71.80	66.93
	FT+Dist	73.07	72.75	71.77	67.02
	DFQ	65.76	52.98	12.65	0.14
	ZeroQ	**72.86**	70.21	59.93	10.81
	GDFQ	70.93	70.13	67.54	59.17
	Ours	72.64	**72.21**	**71.03**	**63.48**
ShuffleNet (65.16)	FT	65.11	63.29	57.65	33.01
	FT+Dist	65.15	63.14	57.65	33.11
	DFQ	56.55	42.46	1.23	0.92
	ZeroQ	**64.77**	51.59	7.91	0.92
	GDFQ	53.74	52.52	45.10	22.89
	Ours	61.90	**60.95**	**56.73**	**28.26**

label information following SN-GAN [73] and set the initial learning rate of the quantized model as 1×10^{-6}. Other training settings are the same as those on CIFAR-10/100.

B. Toy Demonstration for Classification Boundary Matching

To evaluate that the fake data generated by our G are able to match the classification boundary knowledge, we design a toy experiment. The results are shown in Fig. 1. First, we create a toy binary classification data set by uniform sampling from -4 to $+4$, and the label is shown in Fig. 1 (a). Second, we construct a simple neural network T, which is composed of several linear layers, BN layers, and ReLU layers, and we train it using the toy data. The classification boundaries are shown in each subfigure. To simulate the process of our method, we sample noises from Gaussian distribution and every noise has a random label of 0 or 1 (Fig. 1 (b)). Then, we generate fake data from noises by learning from the pre-trained model T. Fig. 1 (c) and Fig. 1 (d) show the fake data generated by the ZeroQ method and our method, respectively. The data generated by ZeroQ does not capture the real data distribution since it neglects the inter-class information; while our method is able to produce fake data that not only have label information but also match the classification boundaries.

C. Comparison Results

To further evaluate the effectiveness of our method, we include the following methods for study. **FP32:** the full-precision pre-trained model. **FT:** quantizing with real training data for fine-tuning the quantized model by minimizing $\mathcal{L}_{\mathrm{CE}}^{Q}$. **FT+Dist**: the same training condition as **FT** but by minimizing $\mathcal{L}_Q$. **DFQ:** a data free post-training quantization method. Here, we obtain the result from the repetition code of DFQ [32]. **ZeroQ:** a data free post-training quantization method. We obtain the result from the publicly released code of ZeroQ [30]. **DSG:** a data free post-training quantization method. We obtain the results published in the paper DSG [33] because it does not release the source code. **GDFQ:** a data free quantization-aware training method. This is the conference version GDFQ [1] of our method. **ZAQ:** a data free quantization-aware training method. We obtain the results published in the paper ZAQ [34]. Although it has the publicly released code, the code cannot run correctly. **DFQAD:** a data free quantization-aware training method. We obtain the results following DFQAD [74].

We quantize model weights and activations to 4-bit, 5-bit, 6-bit and 8-bit to prove the effectiveness of our method in different precisions. We report the comparison results on CIFAR and ImageNet data sets with DFQ, ZeroQ and GDFQ in Table I and Table II, respectively. The results are all running out by ourselves using the published code. To guarantee the comparison as fair as possible, we use the same pre-trained full-precision model. For CIFAR-10, our method achieves much higher accuracy than those of other methods. When the number of categories increases in CIFAR-100, our method suffers a much smaller degradation in accuracy compared with others. The main reason is that our method gains more prior knowledge from the full-precision model. These results demonstrate the effectiveness of our method on simple data sets for the data free quantization task. For large scale and categories data set, such as ImageNet, existing data free quantization methods suffer from severe performance degradation. This phenomenon is dramatically more severe when taking ultra-low-precision such as 4 bit or 5 bit. However, our proposed method is able to mitigate this impact since our generated images contain abundant category information and similar distribution with real data. As a result, our method recovers the accuracy of quantized models significantly with the help of generated fake data and knowledge distillation. Note that our framework also achieves the comparable performance with post-training quantization

TABLE III
COMPARISONS WITH DSG [33] ON IMAGENET

Model	Bit	Method	FP32	QNN	Acc. ↓ (%)
ResNet-18	W6A6	DSG	71.47	70.46	1.01
		Ours	71.58	70.61	**0.97**
	W4A4	DSG	71.47	34.53	36.94
		Ours	71.58	64.39	**7.19**
Inception v3	W6A6	DSG	78.80	76.52	2.28
		Ours	77.63	76.43	**1.20**
	W4A4	DSG	78.80	34.89	43.91
		Ours	77.63	71.22	**6.41**
ShuffleNet	W8A8	DSG	65.07	64.77	**0.30**
		Ours	65.16	61.90	3.26
	W6A6	DSG	65.07	44.88	20.19
		Ours	65.16	60.95	**4.21**
ResNet-50	W8A8	DSG	77.72	77.68	**0.04**
		Ours	77.76	77.50	0.26
	W6A6	DSG	77.72	76.07	1.65
		Ours	77.76	77.21	**0.55**

TABLE IV
COMPARISONS WITH ZAQ [34]

Data Set	Model	Bit	Method	FP32	QNN	Acc. ↓ (%)
Cifar-100	ResNet-20	W5A5	ZAQ	69.58	67.94	1.64
			Ours	70.33	69.68	**0.65**
ImageNet	MoblieNet v2	W8A8	ZAQ	71.88	71.43	0.45
			Ours	73.08	72.64	**0.44**
	ResNet-50	W4A4	ZAQ	76.13	70.06	**6.07**
			Ours	77.76	68.84	8.92

TABLE V
COMPARISONS WITH DFQAD [74]

Model	Data set	Method	W4A8	W8A8
ResNet-20	CIFAR-10	DFQAD	93.42	93.93
		ours	**93.8**	**94.05**
ResNet-20	CIFAR-100	DFQAD	69.04	70.3
		ours	**69.61**	**70.53**
ResNet-18	ImageNet	DFQAD	67.9	70.63
		ours	**68.43**	**70.73**

methods (*e.g.*, ZeroQ and DFQ) in 8 bit quantization. Moreover, following the settings in [74], we compare our method with DFQAD [74] on CIFAR-10/100 and ImageNet in Table V. The results show that our method achieves better quantization performance than DFQAD. Note that under other quantization settings like W4A4 and W6A6, the method DFQAD will lead to numerical errors (*i.e.*, NAN).

Furthermore, we conduct experiments compared with DSG [33] and ZAQ [34] to verify the effectiveness of our proposed method. Due to these two methods do not release the source code, we directly report the results published in the papers in TABLE III and TABLE IV. To guarantee a fair comparison as possible, we employ the **Accuracy** ↓ as a criterion, which denotes the dropped classification accuracy of a quantized model (**QNN**) compared with **FP32**. From the comparison results, we find that our proposed quantization method achieves a state-of-the-art or comparable performance with those of other two methods on several models.

D. Ablation Studies

In this section, we first evaluate the effectiveness of each component in $\mathcal{L}_G$ and $\mathcal{L}_Q$. Second, we explore how fixed BNS affects our method. Then, we compare our method with different quantization methods. Last, we further study the effect of different stopping conditions. All the ablation experiments are conducted on the CIFAR-10/100 data sets.

TABLE VI
EFFECT OF DIFFERENT LOSS FUNCTIONS OF GENERATOR G. WE QUANTIZE BOTH THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-100

Model	CE Loss	BNS Loss	Acc. (%)
ResNet-20 (4-bit)	×	×	48.11
	✓	×	64.51
	×	✓	64.18
	✓	✓	**67.15**

TABLE VII
EFFECT OF DIFFERENT LOSS FUNCTIONS OF Q. WE KEEP THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-100

Model	CE Loss	MSE Loss	Acc. (%)
ResNet-20 (4-bit)	✓	×	58.52
	×	✓	67.01
	✓	✓	**67.15**

1) Effect of Different Components of Losses: To verify the effectiveness of different components in our method, we conduct a series of ablation experiments on CIFAR-100 with ResNet-20. Table VI reports the top-1 accuracy of quantized models with different components of $\mathcal{L}_G$. In this ablation experiment, we fine-tune quantized models with complete $\mathcal{L}_Q$. Since we do not use both CE loss and BNS loss, we have no way to optimize G, which means we use the fake data generated from the initial G to fine-tune the quantized model. In this case, the distribution of the fake data is far away from that of original data because the generator receives no guidance from the full-precision model. Therefore, the quantized model suffers from large performance degradation. To utilize the knowledge in the full-precision model, we use CE loss to optimize G and achieve a better quantized model. In this case, the generator produces fake data that can be classified with high confidence by the full-precision model. Last, we combine CE loss and BNS loss with a coefficient and achieve the best result. The BNS loss encourages the generator to generate fake data that matches the statistics encoded in full-precision model's BN layers so that these fake data have a much similar distribution with real data. In summary, both CE loss and BNS loss contribute to the better performance of the quantized model.

We further conduct ablation experiments to analyze the effectiveness of each component in $\mathcal{L}_Q$. Table VII reports the top-1 accuracy of quantized models with different components of $\mathcal{L}_Q$. In this experiment, we optimize the generator with complete $\mathcal{L}_G$. When only introducing the MSE loss, the quantized model receives knowledge from the full-precision model's prediction and achieves 67.01% on the top-1 accuracy. To use the additional label information, we combine MSE loss with CE loss. The resultant model achieves a 0.14% improvement on the top-1 accuracy.

2) Effect of MSE Alignment: We conduct experiments on CIFAR-100 for ResNet-20 to compare the effect of the MSE alignment and the KL divergence. In this experiment, we quantize the models to 6-bit and 4-bit and then fine-tune them. We keep other loss functions unchanged but only change the knowledge transfer loss by $\mathcal{L}_{\text{KL}}$ with multiple distillation

TABLE VIII

COMPARISONS BETWEEN THE KL LOSS AND MSE LOSS. WE KEEP THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-100

Loss		Accuracy Top-1 (%)	
		W4A4	W6A6
$\mathcal{L}_{\mathrm{KL}}$	τ=1	58.99	66.53
	τ=3	63.4	68.25
	τ=5	63.68	68.59
	τ=20	63.87	68.47
	$\tau = \infty$	58.79	67.28
$\mathcal{L}_{\mathrm{MSE}}$		**67.15**	**70.35**

TABLE IX

EFFECT OF THE PARAMETER β OF LOSS FUNCTION $\mathcal{L}_{\mathrm{BNS}}$. WE KEEP THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-100

β	0.01	0.1	1	10
Acc. (%)	66.09	66.92	**67.15**	66.92

temperatures τ. Table VIII reports the top-1 accuracy of quantized models. For both 4-bit and 6-bit, fine-tuning with $\mathcal{L}_{\mathrm{MSE}}$ achieves better performance than that with $\mathcal{L}_{\mathrm{KL}}$ for different temperatures τ. Moreover, when the quantization bit is lower, the performance discrepancy is more obvious.

Particularly, to verify our theoretical results proposed in Section V, we calculate the $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{MSE}}$ when we fine-tuning the quantized model either with KL divergence or MSE alignment. Fig. 3(a) shows the loss curves of MSE distance during the training process with $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{MSE}}$ respectively. The MSE distance can measure the degree of similarity between the outputs of the pre-trained model and the quantized model, where is the lower the better. When we fine-tune the quantized model with KL divergence, the MSE distance is still quite high. When we use our MSE alignment to fine-tune the quantized model, the MSE distance obviously falls to a lower value. So our MSE alignment encourages the quantized model to be more similar to the full-precision model for better performance. One may say that we use the MSE distance as the optimization objective, so the MSE distance must be lower than that of the KL divergence as the optimization objective. To further show the effectiveness of our MSE alignment, we show the loss curves of KL distance during the training process with $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{MSE}}$ respectively in Fig. 3(b). We use the KL divergence as the optimized objective, however, the KL distance trained with KL is still higher than that trained with MSE. Therefore, fine-tuning with the MSE alignment not only decreases the MSE distance but also the KL distance, which is more effective for the data free quantization task.

3) Effect of Parameters in Loss Functions: We study how hyper-parameters β and γ affect the performance of our quantized models by numerical experiments on CIFAR-100 dataset in TABLE IX and TABLE X, respectively. We observe that our proposed quantization method is not very sensitive to the choices of the β or γ. Empirically, when $\beta = 1$ and $\gamma = 1$, our method is able to obtain a good quantized model with high classification accuracy.

TABLE X

EFFECT OF THE PARAMETER γ OF LOSS FUNCTION $\mathcal{L}_{\mathrm{MSE}}$. WE KEEP THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-100

γ	0.01	0.1	1	10
Acc. (%)	63.57	65.84	**67.15**	65.24

TABLE XI

ABLATION EXPERIMENTS ON THE FIXED BNS (FBNS). WE KEEP THE WEIGHTS AND ACTIVATIONS OF THE MODELS TO BE 4-BITS AND REPORT THE TOP-1 ACCURACY ON CIFAR-10/100. WE USE "W/O FBNS" TO REPRESENT THAT WE USE FAKE DATA TO FINE-TUNE THE QUANTIZED MODELS WITHOUT FIXED BNS. SIMILARLY, WE USE "W/ FBNS" TO REPRESENT THE FINE-TUNING PROCESS WITH FIXED BNS

Data Set	w/o FBNS (%)	w/ FBNS (%)
CIFAR-10	89.80	**92.24**
CIFAR-100	65.02	**67.15**

TABLE XII

COMPARISON OF SEPARATE TRAINING AND ALTERNATING TRAINING OF G AND Q

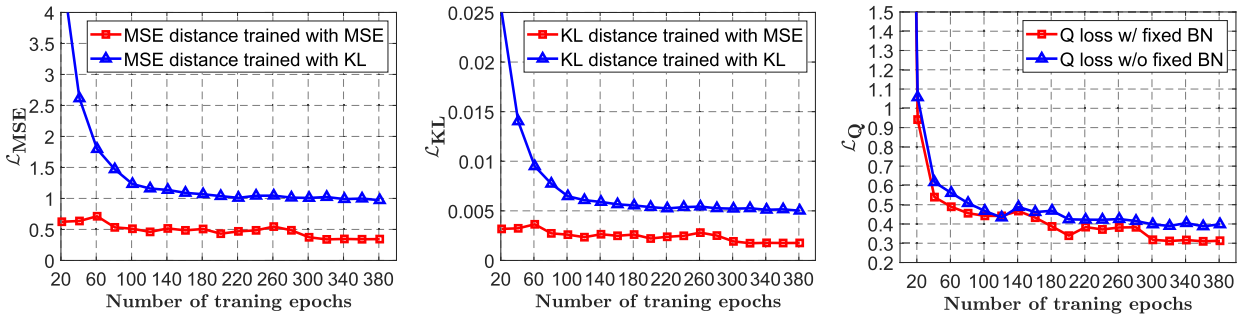
Training Strategy	Acc. (%)
Separate Training	67.07
Alternating Training	**67.15**

4) Effect of the Fixed BNS: To verify the effectiveness of fixing batch normalization, we conduct ablation studies with ResNet-20 on CIFAR-10/100. The results are shown in Table XI. When we fix batch normalization statistics during fine-tuning, we narrow the statistics gap between the quantized model and the full-precision model. As a result, we achieve a much higher top-1 accuracy than that with standard batch normalization. Moreover, to show the effect and stability of the fixed BNS more significantly, we illustration $\mathcal{L}_Q$ decline curve during training in Fig 3(c). The $\mathcal{L}_Q$ with the fixed BNS is always lower than that without the fixed BNS. On the other hand, the decreasing loss curve also proves that the proposed fake-data driven quantization can properly fine-tune the quantized model based on the generated fake data.

E. Further Experiments

1) Effect of Two Training Strategies: We investigate the effect of two kinds of training strategies. 1) Training generator and quantized model in two steps. We first train the generator by minimizing the loss (6) until convergence. Then, we train the quantized model by minimizing the loss (12). 2) Training the generator and quantized model alternately in each iteration following Algorithm 1. From the results in Table XII, alternating training performs better than separate training. Therefore, we use alternating training in other experiments.

2) Effect of Different Thresholds in Stopping Condition: In this experiment, we stop the training of the generator if the classification accuracy of the full-precision model on fake data is larger than a threshold η. Table XIII reports the results of different thresholds η in the stopping condition. When increasing the threshold, the generator is trained with



(a) MSE distance (b) KL distance (c) w/ and w/o fixed BN

Fig. 3. The comparisons of training with KL, MSE loss and fixed BN.

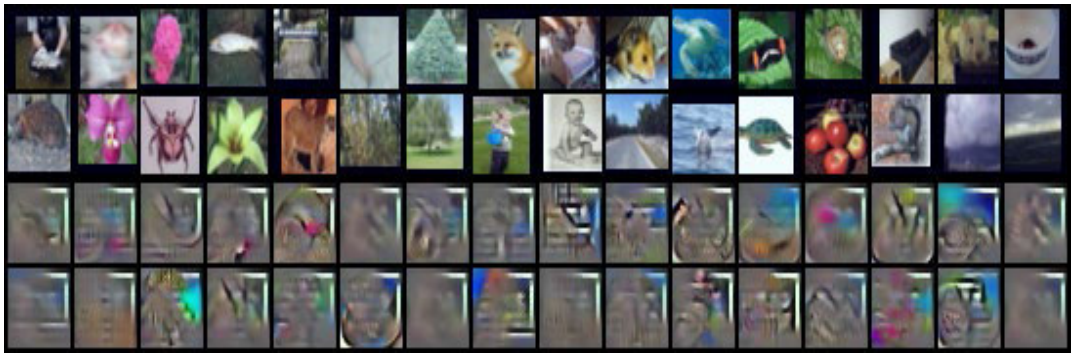


Fig. 4. The comparisons of generated samples and real samples on CIFAR100. The first two lines are real data, while the last two lines are the generated data with our method.

TABLE XIII

EFFECT OF DIFFERENT THRESHOLDS IN THE STOPPING CONDITION OF G

Threshold η (%)	Acc. (%)
90.00	65.34
95.00	65.92
99.00	66.32
99.50	66.36
without stopping condition	**67.15**

quantized models for more epochs, and we get a better fine-tuning result. We achieve the best performance when we do not stop optimizing the generator. This demonstrates that optimizing the generator and quantized model simultaneously increases the diversity of data, which is helpful for fine-tuning quantized models.

3) Case Study of Generated Images: As a starting point, we aim to train a good generator to learn data distribution from the full-precision model by exploiting the classification boundary information. So we want to investigate whether the generator is able to capture the helpful knowledge for quantization. To this end, we visualize the generated samples compared with real samples on CIFAR-100 in Fig. 4. We observe that the generated samples are informative with perceptually relevant features (such as edges, contours and corners) of the real images. At the same time, they have good confidentiality without the privacy of the original data. This observation suggests that there may exist some approach to effectively exploit the original data information with a neural network. We believe that this phenomenon warrants an in-depth investigation and we view our experiments as only exploratory.

VII. CONCLUSION

In this paper, we have proposed a Generative Data Free Model Quantization with Knowledge Matching for Classification scheme to eliminate the data dependence of quantization methods. First, we have constructed a knowledge matching generator to produce fake data for the fine-tuning process. The generator is able to learn the classification boundary knowledge and distribution information from the pre-trained full-precision model. Next, we have quantized the full-precision model and fine-tuned the quantized model using the fake data. Extensive experiments on various image classification data sets have demonstrated the effectiveness of our data free method.

REFERENCES

[1] S. Xu et al., "Generative low-bitwidth data free quantization," in *Proc. Eur. Conf. Comp. Vis.*, 2020, pp. 1–17.

[2] X. Xiong, W. Min, Q. Wang, and C. Zha, "Human skeleton feature optimizer and adaptive structure enhancement graph convolution network for action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 342–353, Jan. 2023.

[3] Y. Chen, H. Ge, Y. Liu, X. Cai, and L. Sun, "AGPN: Action granularity pyramid network for video action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jan. 9, 2023, doi: 10.1109/TCSVT.2023.3235522.

[4] L. Yan, Y. Qin, and J. Chen, "Scale-balanced real-time object detection with varying input-image resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 242–256, Jan. 2023.

[5] X. Chen, H. Li, Q. Wu, K. N. Ngan, and L. Xu, "High-quality R-CNN object detection using multi-path detection calibration network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 715–727, Feb. 2021.

[6] W. Liu, G. Lin, T. Zhang, and Z. Liu, "Guided co-segmentation network for fast video object segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1607–1617, Apr. 2021.

[7] M. Saïd Allili, D. Ziou, N. Bouguila, and S. Boutemedjet, "Image and video segmentation by combining unsupervised generalized Gaussian mixture modeling and feature selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1373–1377, Oct. 2010.

[8] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3943–3956, Nov. 2020.

[9] M. Zhao, J. Peng, S. Yu, L. Liu, and N. Wu, "Exploring structural sparsity in CNN via selective penalty," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1658–1666, Mar. 2022.

[10] J. Guo, W. Zhang, W. Ouyang, and D. Xu, "Model compression using progressive channel pruning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1114–1124, Mar. 2021.

[11] Z. Chen, Z. Chen, J. Lin, S. Liu, and W. Li, "Deep neural network acceleration based on low-rank approximated channel pruning," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 4, pp. 1232–1244, Apr. 2020.

[12] W. Xu et al., "Improving extreme low-bit quantization with soft threshold," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 4, pp. 1549–1563, Apr. 2022.

[13] C. Liu et al., "RB-Net: Training highly accurate and efficient binary neural networks with reshaped point-wise convolution and balanced activation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6414–6424, Sep. 2022.

[14] J. Ye, S. Zhang, T. Huang, and Y. Rui, "CDbin: Compact discriminative binary descriptor learned with efficient neural network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 862–874, Mar. 2020.

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

[16] D. T. Nguyen, H. Kim, and H. Lee, "Layer-specific optimization for mixed data flow with mixed precision in FPGA design for CNN-based object detectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2450–2464, Jun. 2021.

[17] W. Hong, T. Chen, M. Lu, S. Pu, and Z. Ma, "Efficient neural image decoding via fixed-point inference," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3618–3630, Sep. 2021.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 525–542.

[19] S. Liang, S. Yin, L. Liu, W. Luk, and S. Wei, "FP-BNN: Binarized neural network on FPGA," *Neurocomputing*, vol. 275, pp. 1072–1086, Jan. 2018.

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

[21] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7543–7552.

[22] R. Banner, Y. Nahshan, E. Hoffer, and D. Soudry, "Aciq: Analytical clipping for integer quantization of neural networks," in *Proc. ICLR*, 2018, pp. 1–11.

[23] B. Hu, S. Zhou, Z. Xiong, and F. Wu, "Cross-resolution distillation for efficient 3D medical image registration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7269–7283, Oct. 2022.

[24] B. Chen, Z. Zhang, Y. Li, G. Lu, and D. Zhang, "Multi-label chest X-ray image classification via semantic similarity graph embedding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2455–2468, Apr. 2022.

[25] L. Tang, Y. Wang, and L. Chau, "Weakly-supervised part-attention and mentored networks for vehicle re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8887–8898, Dec. 2022.

[26] W. Xing, Y. Yang, S. Zhang, Q. Yu, and L. Wang, "NoisyOTNet: A robust real-time vehicle tracking model for traffic surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2107–2119, Apr. 2022.

[27] Y. Zhang, P. Zhao, Q. Wu, B. Li, J. Huang, and M. Tan, "Cost-sensitive portfolio selection via deep reinforcement learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 236–248, Jan. 2022.

[28] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–11.

[29] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–11.

[30] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M. W. Mahoney, and K. Keutzer, "ZeroQ: A novel zero shot quantization framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13166–13175.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[32] M. Nagel, M. V. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1325–1334.

[33] X. Zhang et al., "Diversifying sample generation for accurate data-free quantization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15653–15662.

[34] Y. Liu, W. Zhang, and J. Wang, "Zero-shot adversarial quantization," 2021, *arXiv:2103.15263*.

[35] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[36] B. Zhuang, C. Shen, M. Tan, L. Liu, and I. Reid, "Structured binary neural networks for accurate image classification and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 413–422.

[37] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," in *Proc. ICLR*, 2020, pp. 1–11.

[38] B. Zhuang, C. Shen, M. Tan, L. Liu, and I. Reid, "Towards effective low-bitwidth convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7920–7928.

[39] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave Gaussian quantization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5406–5414.

[40] S. Jung et al., "Learning to quantize deep networks by optimizing quantization intervals with task loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4345–4354.

[41] D. Zhang, J. Yang, D. Ye, and G. Hua, "LQ-Nets: Learned quantization for highly accurate and compact deep neural networks," in *Proc. Eur. Conf. Comp. Vis.*, 2018, pp. 365–382.

[42] J. Yang et al., "Quantization networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7300–7308.

[43] C. Louizos, M. Reisser, T. Blankevoort, E. Gavves, and M. Welling, "Relaxed quantization for discretized neural networks," in *Proc. ICLR*, 2019, pp. 1–10.

[44] B. Zhuang, L. Liu, M. Tan, C. Shen, and I. Reid, "Training quantized neural networks with a full-precision auxiliary module," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1485–1494.

[45] R. Banner, Y. Nahshan, and D. Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–9.

[46] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[47] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[48] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[49] M. Haroush, I. Hubara, E. Hoffer, and D. Soudry, "The knowledge within: Methods for data-free model compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8494–8502.

[50] H. Yin et al., "Dreaming to distill: Data-free knowledge transfer via DeepInversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8715–8724.

[51] H. Chen et al., "Data-free learning of student networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3513–3521.

[52] J. Yoo, M. Cho, T. Kim, and U. Kang, "Knowledge extraction with no observable data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2701–2710.

[53] P. Micaelli and A. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," 2019, *arXiv:1905.09768*.

[54] G. Fang, J. Song, C. Shen, X. Wang, D. Chen, and M. Song, "Data-free adversarial distillation," 2019, *arXiv:1912.11006*.

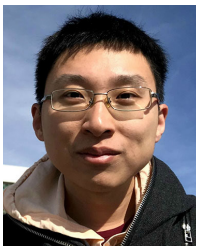
[55] R. Gontijo Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," 2017, *arXiv:1710.07535*.

[56] M. M. Kalayeh and M. Shah, "Training faster by separating modes of variation in batch-normalized models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1483–1500, Jun. 2020.
[57] J. Cao, Y. Guo, Q. Wu, C. Shen, and M. Tan, "Adversarial learning with local coordinate coding," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1–11.
[58] J. Cao, L. Mo, Y. Zhang, K. Jia, C. Shen, and M. Tan, "Multi-marginal Wasserstein GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.
[59] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.
[60] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.
[61] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[62] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.
[63] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–14.
[67] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[68] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[69] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[70] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.
[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–11.
[72] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O\left(\frac{1}{k^2}\right)$," in *Proc. USSR Acad. Sci.*, vol. 269, 1983, pp. 543–547.
[73] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.
[74] Y. Choi, J. Choi, M. El-Khamy, and J. Lee, "Data-free network quantization with adversarial knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 3047–3057.

Shoukai Xu is currently pursuing the Ph.D. degree with the South China University of Technology, China. He has published papers in ECCV. His research interests include model compression, knowledge distillation, and foundation model application.

Shuhai Zhang is currently pursuing the Ph.D. degree with the South China University of Technology, China. He has published articles in neural networks. His research interests include machine learning and mainly focus on model compression and adversarial attack.

Jing Liu received the bachelor's and master's degrees from the School of Software Engineering, South China University of Technology, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University, Clayton Campus, Australia. His research interests include computer vision, model compression, and acceleration.



Bohan Zhuang received the bachelor's degree in electronic and information engineering from the Dalian University of Technology, China, in 2014, and the Ph.D. degree from the School of Computer Science, The University of Adelaide, Australia, in 2018. From 2018 to 2020, he was a Senior Research Associate with the School of Computer Science, The University of Adelaide. He is currently an Assistant Professor with the Faculty of Information Technology, Monash University, Clayton Campus, Australia. His research interests include computer vision and machine learning.

Yaowei Wang received the Ph.D. degree in computer science from the Graduate University of Chinese Academy of Sciences in 2005. He was an Assistant Professor with the School of Information and Electronics, Beijing Institute of Technology, and a Guest Assistant Professor with the National Engineering Laboratory for Video Technology, Peking University, China. He is currently a Researcher with the Peng Cheng Laboratory, Shenzhen, China. He has been the author or coauthor of more than 50 refereed journals and conference papers. His research interests include machine learning and multimedia content analysis and understanding.

Mingkui Tan (Member, IEEE) received the bachelor's degree in environmental science and engineering and the master's degree in control science and engineering from Hunan University, Changsha, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014. From 2014 to 2016, he was a Senior Research Associate of computer vision with the School of Computer Science, The University of Adelaide, Australia. He is currently a Professor with the School of Software Engineering, South China University of Technology. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.