Hard Sample Matters a Lot in Zero-Shot Quantization

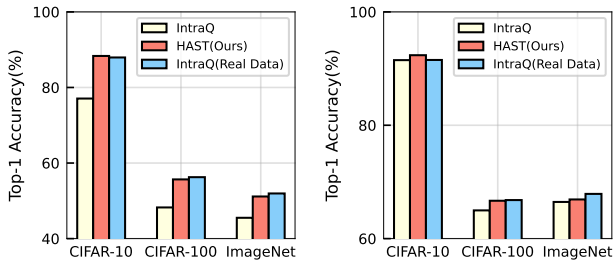
Huantong Li[1 6 *] Xiangmiao Wu[1] Fanbing Lv[2] Daihai Liao[2]
Thomas H. Li[7] Yonggang Zhang[3 * †] Bo Han[3] Mingkui Tan[1 4 5 †]
[1]South China University of Technology, [2]Changsha Hisense Intelligent System Research Institute Co., Ltd
[3]Hong Kong Baptist University, [4]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education
[5]PengCheng Laboratory, [6]Information Technology R&D Innovation Center of Peking University
[7]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China

Abstract

*Zero-shot quantization (ZSQ) is promising for compressing and accelerating deep neural networks when the data for training full-precision models are inaccessible. In ZSQ, network quantization is performed using synthetic samples, thus, the performance of quantized models depends heavily on the quality of synthetic samples. Nonetheless, we find that the synthetic samples constructed in existing ZSQ methods can be easily fitted by models. Accordingly, quantized models obtained by these methods suffer from significant performance degradation on hard samples. To address this issue, we propose **HA**rd sample **S**ynthesizing and **T**raining (HAST). Specifically, HAST pays more attention to hard samples when synthesizing samples and makes synthetic samples hard to fit when training quantized models. HAST aligns features extracted by full-precision and quantized models to ensure the similarity between features extracted by these two models. Extensive experiments show that HAST significantly outperforms existing ZSQ methods, achieving performance comparable to models that are quantized with real data.*

1. Introduction

Deep neural networks (DNNs) achieve great success in many domains, such as image classification [28, 43, 44], object detection [15, 16, 39], semantic segmentation [13, 53], and embodied AI [3, 4, 11]. These achievements are typically paired with the rapid growth of parameters and computational complexity, making it challenging to deploy DNNs on resource-constrained edge devices. In response to the challenge, *network quantization* proposes to represent the full-precision models, i.e., floating-point parameters and activations, using low-bit integers, resulting in a high compression rate and an inference-acceleration rate [24]. These methods implicitly assume that the training data of full-precision models are available for the process of network quantization. However, the training data, e.g., medical records, can be inaccessible due to privacy and security issues. Such a practical and challenging scenario has led to the development of *zero-shot quantization* (ZSQ) [2], quantizing networks without accessing the training data.



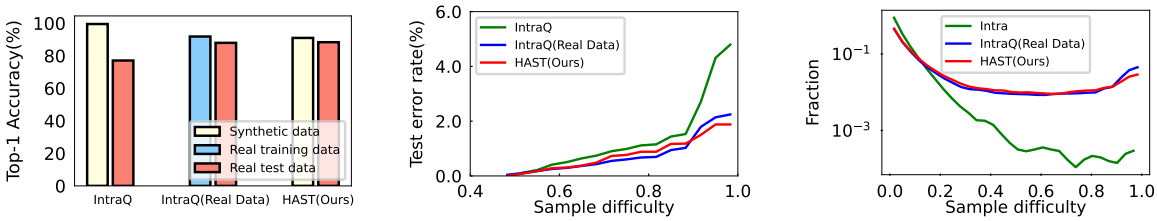
(a) 3-bit quantization. (b) 4-bit quantization.

Figure 1. Performance of the proposed HAST on three datasets compared with the state-of-the-art method IntraQ [50] and the method fine-tuning with real data [50]. HAST quantizes ResNet-20 on CIFAR-10/CIFAR-100 and ResNet-18 on ImageNet to 3-bit (left) and 4-bit (right), achieving performance comparable to the method fine-tuning with real data.

Many efforts have been devoted to ZSQ [2, 19, 38, 46, 48, 50]. In ZSQ, some works perform network quantization by weight equalization [38], bias correction [1], or weight rounding strategy [19], at the cost of some performance degradation. To promote the performance of quantized models, advanced works propose to leverage synthetic data for network quantization [2, 7, 46, 48, 50]. Specifically, they fine-tune quantized models with data synthesized using full-precision models, achieving promising improvement in performance.

Much attention has been paid to the generation of synthetic sample, since high-quality synthetic samples lead to high-performance quantized models [2, 46]. Recent works employ generative models to synthesize data with fruitful approaches, considering generator design [51], boundary sample generation [6], adversarial training scheme [35], and

*Equal contribution. Email: 202021046173@mail.scut.edu.cn
†Corresponding author. Email: mingkuitan@scut.edu.cn



(a) Performance of converged 3-bit ResNet-20. (b) Variation of error rate with sample difficulty. (c) Variation of fraction with sample difficulty.

Figure 2. Analysis on synthetic data. (a) Performance of converged 3-bit ResNet-20. We quantize ResNet-20 to 3-bit using IntraQ [50], Real Data [50], and Our HAST, respectively, The top-1 accuracy on both training data (synthetic data for ZSQ methods) and test data is reported. (b) The error rate of test samples with different difficulties. (c) Distribution visualization of sample difficulty using GHM [30]. For each converged quantized model, we randomly sample 10,000 synthetic/real samples and count the fraction of samples based on difficulty. Note that the y-axis uses a log scale since the number of samples with different difficulties can differ by order of magnitude.

effective training strategy [7]. Since the quality of synthetic samples is typically limited by the generator [36], advanced works treat synthesizing samples as a problem of noise optimization [2]. Namely, the noise distribution is optimized to approximate some specified properties of real data distributions, such as batch normalization statistics (BNS) and inception loss (IL) [20]. To promote model performance, IntraQ [50] focuses on the property of synthetic samples and endows samples with heterogeneity, achieving state-of-the-art performance as depicted in Figure 1.

Although existing ZSQ methods achieve considerable performance gains by leveraging synthetic samples, there is still a significant performance gap between models trained with synthetic data and those trained with real data [50]. To reduce the performance gap, we investigate the difference between real and synthetic data. Specifically, we study the difference in generalization error between models trained with real data and those trained with synthetic data. Our experimental results show that synthetic data lead to larger generalization errors than real data, as illustrated in Figure 2a. Namely, synthetic sample lead to a more significant gap between training and test accuracy than real data.

We conjecture that the performance gap stems from the misclassification of hard test samples. To verify the conjecture, we conduct experiments to study how model performance varies with sample difficulty, where GHM [30] is employed to measure the difficulty of samples quantitatively. The results shown in Figure 2b demonstrate that, on difficult samples, models trained with synthetic data perform worse than those trained with real data. This may result from that synthetic samples are easy to fit, which is consistent with the observation on inception loss of synthetic data [33]. We verify the assumption through a series of experiments, where we count the fraction of samples of different difficulties using GHM [30]. The results are reported in Figure 2c, where we observe a severe missing of hard samples in synthetic samples compared to real data. Consequently, quantized models fine-tuned with these synthetic data may fail to generalize well on hard samples in the test set.

In light of conclusions drawn from Figure 2, the samples synthesized for fine-tuning quantized models in ZSQ should be hard to fit. To this end, we propose a novel **HA**rd sample **S**ynthesizing and **T**raining (HAST) scheme. The insight of HAST has two folds: a) The samples constructed for fine-tuning models should not be easy for models to fit; b) The features extracted by full-precision and the quantized model should be similar. To this end, in the process of synthesizing samples, HAST pays more attention to hard samples in a re-weighting manner, where the weights are equal to the sample difficulty introduced in GHM [30]. Meanwhile, in the fine-tuning process, HAST further promotes the sample difficulty on the fly and aligns the features between the full-precision and quantized models.

To verify the effectiveness of HAST, we conduct comprehensive experiments on three datasets under two quantization precisions, following settings used in [50]. Our experimental results show that HAST using only 5,120 synthetic samples outperforms previous state-of-the-art method [50] and even achieves performance comparable with quantization with real data.

Our main contributions can be summarized as follows:

- We observe that the performance degradation in zero-shot quantization is attributed to the lack of hard samples. Namely, the synthetic samples used in existing ZSQ methods are easily fitted by quantized models, distinguishing models trained on synthetic samples from those trained on real data, as depicted in Figure 2.
- Built upon our empirical observation, we propose a novel **HA**rd sample **S**ynthesizing and **T**raining (HAST) scheme to promote the performance of ZSQ. Specifically, HAST generates hard samples and further promotes the sample difficulty on the fly when training models, paired with a feature alignment constraint to ensure the similarity of features extracted by these two models, as summerized in Algorithm 1.
- Extensive experiments demonstrate the superiority of HAST over existing ZSQ methods. More specifically,

HAST using merely 5,120 synthetic samples outperforms the previous state-of-the-art method and achieves performance comparable to models fine-tuned using real training data, as shown in Figure 1.

2. Related Work

In this section, we briefly review the most relevant works in network quantization.

2.1. Data-Driven Quantization

Network quantization is proposed to promote the compression rates and accelerate the inference by representing the full-precision models using low-bit integers. The straightforward and effective approach is to fine-tune models using the training data of full-precision models. In this regard, quantization aware training (QAT) methods focus on designing quantizers [5, 12, 17, 25, 31], training strategies [29, 52], and binary networks [34, 37, 41].

In many practical scenarios, the training data of full-precision models can be inaccessible, limiting the effectiveness of QAT methods. To address the challenge, post-training quantization (PTQ) methods perform network quantization with limited training data [1, 32]. Specifically, these methods approximate an optimal clipping value in the feature space using limited training samples [1] and introduce an allocation policy to quantize both activations and weights to 4-bit. However, both QAT and PTQ methods require training data to perform network quantization.

2.2. Zero-Shot Quantization

To further relax the privacy or security issue, ZSQ methods propose to perform network quantization without accessing the training data of full-precision models. A simple yet effective approach is calibrating model parameters without training data. For example, DFQ [38] equalizes the weight ranges in the network and utilizes the scale and shift parameters stored in the batch normalization layers to correct biased quantization errors. SQuant [19] performs data-free quantification using the diagonal Hessian approximation layer by layer. However, these methods may lead to significant performance degradation when quantizing models with ultra-low precision [50].

Advanced works propose to generate synthetic samples for fine-tuning quantized models, resulting in promoted model performance. GDFQ [46] first adopts generative models guided by both the batch normalization statistics and extra category label information to synthesize samples. Since the performance depends heavily on synthetic samples, many variants of GDFQ improve the performance by adopting better generator [51], adversarial training [35], boundary-supporting samples generation [6], and using an ensemble of compressed models [22]. Recently, it has been shown that data synthesis can be realized by optimizing random noise sampled from a pre-defined distribution [2]. The noise optimization scheme is further improved by carefully designing an appropriate mechanism for synthesizing data, such as enhancing the diversity of samples [48] and increasing the intra-class heterogeneity of synthetic samples [50]. Although these methods achieve considerable performance gain compared to data-driven quantization, there is still a performance gap between fine-tuning with synthetic and real data.

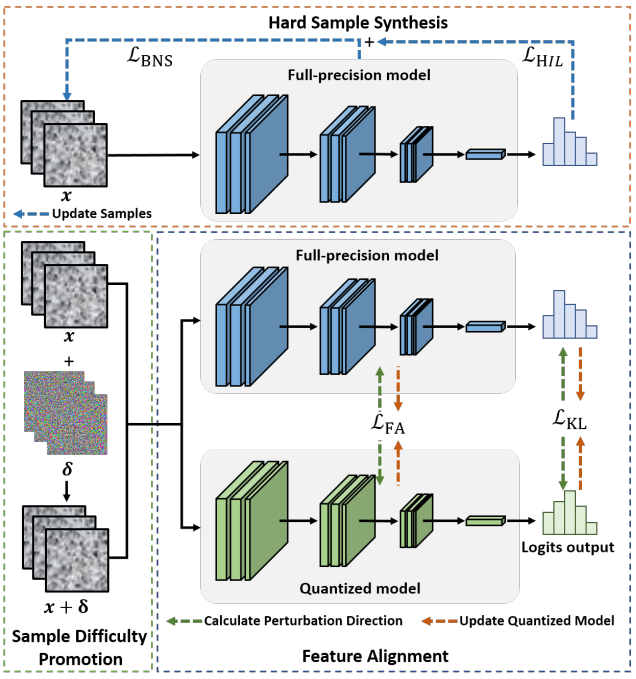


Figure 3. An overview of the proposed hard sample synthesizing and training (HAST) scheme. The hard sample synthesis focuses on generating hard samples in the data generation process. The sample difficulty promotion makes synthetic samples hard to fit. The feature alignment guides the quantized model to handle hard samples.

3. Methodology

In this section, we first introduce the process of zero-shot quantization (ZSQ). Then, we detail the proposed **HA**rd sample **S**ynthesizing and **T**raining (HAST) scheme containing three parts: *hard sample synthesis*, *sample difficulty promotion*, and *feature alignment* ensuring the similarity of representations from quantized models and full precision models. The overview of HAST is illustrated in Figure 3.

3.1. Preliminaries

Quantizer. We focus on the asymmetric uniform quantizer to implement network quantization in this work, following [46, 50]. Denoting θ as weights of a full-precision model, l and u as the lower and upper bound of θ, and n as the bit-width, the quantizer produces the quantized integer

θ^q as follows:

$$\theta^q = \lfloor \theta \times S - z \rceil, \; S = \frac{2^n - 1}{u - l}, \; z = S \times l + 2^{n-1} \quad (1)$$

where S is the scaling factor to convert the range of θ to n-bit, z decides which quantized value zero is mapped to, and "$\lfloor \rceil$" is an operation used to round its input to the nearest integer. Accordingly, the corresponding dequantized value can be calculated as:

$$\theta' = \frac{\theta^q + z}{S}. \quad (2)$$

Built upon the quantizer, the weights of full-precision models can be represented using low-bit integers. Consequently, the quantized model performs inference using the dequantized parameter θ', which is typically paired with considerable performance degradation.

Zero-Shot Quantization. To mitigate the performance degradation, a widely adopted approach is to optimize θ such that its dequantized parameter θ' can perform well on test sets [5]. This is typically achieved by optimizing θ using the data for training full-precision models [12], since θ' is inherently the function of θ. Zero-Shot Quantization (ZSQ) takes a step further to quantize models when data for training full-precision models are inaccessible.

In ZSQ, synthetic samples are usually employed for optimizing quantized models [2, 46]. The synthetic samples can be derived by noise optimization [2, 48, 50], which is typically instantiated by distribution approximation [2, 46]. Given a set of noise $\{\mathbf{x}_i\}_{i=1}^N$, synthetic samples are obtained by optimizing these noise to match the batch normalization statistics (BNS):

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{BNS} \triangleq \frac{1}{L}\sum_{l=1}^{L}(||\mu^l(\theta) - \mu^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)|| + ||\sigma^l(\theta) - \sigma^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)||), \quad (3)$$

where $\mu^l(\theta)/\sigma^l(\theta)$ are mean/variance parameters stored in the l-th BN layer of full-precision model parameterized with θ and $\mu^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)/\sigma^l(\theta, \{\mathbf{x}_i\}_{i=1}^N)$ are mean/variance parameters calculated on the sampled noise using θ. Besides the BNS alignment objective function, an inception loss is also employed for optimizing sampled noise:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{IL} \triangleq \frac{1}{N}\sum_{i=1}^{N} CE(p(\mathbf{x}_i; \theta), \mathbf{y}_i), \quad (4)$$

where $p(\cdot; \theta)$ stands for the probability predicted by full-precision model parameterized with θ, $CE(\cdot, \cdot)$ represents the cross-entropy loss, and $\mathbf{y}_i$ is the label assigned to $\mathbf{x}_i$ as a prior classification knowledge. Consequently, synthetic data are obtained by optimizing the final objective function composed of these two terms:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{FNL} \triangleq \mathcal{L}_{BNS} + \beta \mathcal{L}_{IL}, \quad (5)$$

where β is a hyper-parameter balancing the importance of two terms. Built upon the objective function $\mathcal{L}_{FNL}$, ZSQ can fine-tune the model parameter θ such that its dequantized parameter θ' can perform well on synthetic samples. More specifically, the quantized network is usually fine-tuned by a teacher-student framework with the cross-entropy loss CE and the Kullback-Leibler loss KL:

$$\min_{\theta'} \frac{1}{N}\sum_{i=1}^{N} CE(p(\mathbf{x}_i; \theta'), \mathbf{y}_i) + \alpha KL(p(\mathbf{x}_i; \theta)||p(\mathbf{x}_i; \theta')), \quad (6)$$

where α is a hyper-parameter balancing the importance of two terms. We use θ' to represent the parameter of quantized models and optimize θ'. This is equal to optimizing θ, as θ' is a function of θ according to Eq. (2).

3.2. General Scheme of Proposed Methods

The detailed procedure of the proposed HAST scheme is summarized in Algorithm 1. HAST consists of three parts: hard sample synthesis, sample difficulty promotion, and feature alignment that is illustrated in Sec. 3.

In the process of sample synthesis, we start with a batch of random input $\mathbf{x}_i$ sampled from a standard Gaussian distribution, following [50]. The proposed hard sample synthesis, i.e., Eq. (9), optimizes $\mathbf{x}_i$, aiming to increase the sample difficulty measured with full-precision models. As shown in the top of Figure 3, we feed $\mathbf{x}_i$ into the full-precision model and optimize it using the proposed hard-sample-enhanced final loss for synthesizing hard samples, i.e., Eq. (9) by computing the BNS alignment loss (Eq. (3)) and the hard-sample-enhanced inception loss (Eq. (8)).

In the process of network fine-tuning, we increase the sample difficulty on the fly using Eq. (10), as shown in the bottom left of Figure 3. Then, we perform feature alignment to ensure the similarity between the full-precision and quantized models, as shown in the bottom right of Figure 3.

3.3. Hard Sample Synthesis

Inspired by our experimental results in Figure 2, we propose to reconsider the difficulty of synthetic samples. Specifically, we increase the importance of hard samples while suppressing the importance of easy-to-fit samples, employing the GHM introduced in [30] to measure the sample difficulty d of $\mathbf{x}$:

$$d(\mathbf{x}, \theta) = 1 - p_{\mathbf{y}}(\mathbf{x}, \theta) \quad (7)$$

where $p_{\mathbf{y}}(\mathbf{x})$ is the probability on label $\mathbf{y}$ predicted by the model parameterized with θ. Through Eq. (7), we hope models focus more on transferable components [8].

Built upon the sample difficulty $d(\mathbf{x}, \theta)$, we propose a hard-sample-enhanced inception loss $\mathcal{L}_{HIL}$ to synthesize

Algorithm 1 Hard Sample Synthesizing and Training for Zero-shot Quantization.

Input: Pretrained full-precision model with parameter θ; Initial synthetic dataset $D_s = \varnothing$; Number of synthetic images N; Generation iterations $T_{generate}$; Fine-tuning iterations $T_{fine-tune}$.
Output: Low precision quantized model with parameter θ'.

```
while |D_s| ≤ N do
    Sample batch of Gaussian noise x_i and label y;
    for t = 1, ..., T_generate do
        Optimize x_i by minimizing Eq. (9);
    end for
    D_s ← D_s ∪ x_i
end while
Get Synthetic dataset D_s.
for t = 1, ..., T_fine-tune do
    Sample batch x_i from synthetic dataset X;
    Compute the perturbation δ by Eq. (10);
    Update quantized model by minimizing Eq. (12);
end for
Get Converged quantized model.
```

samples:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{HIL} \triangleq \frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}_i,\theta)^{\gamma} CE(p(\mathbf{x}_i;\theta),\mathbf{y}_i) \quad (8)$$

where the hyper-parameter γ controls how strongly the importance of hard samples is enhanced. The hard-sample-enhanced inception loss $\mathcal{L}_{HIL}$, i.e., Eq. (8), pays more attention to hard samples in the synthesis process than the original inception loss $\mathcal{L}_{IL}$. Figure 4 shows the curves of $\mathcal{L}_{IL}$ and $\mathcal{L}_{HIL}$. As sample difficulty decreases, loss drops and finally converges to zero. However, the sample optimized using $\mathcal{L}_{HIL}$ (0.7) is more difficult than that using $\mathcal{L}_{IL}$ (0.2). Consequently, the fraction of hard samples will increase, as depicted in Figure 2c, where samples are synthesized by optimizing the hard-sample-enhanced loss:

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{L}_{HFNL} \triangleq \mathcal{L}_{BNS} + \beta\mathcal{L}_{HIL}. \quad (9)$$

3.4. Sample Difficulty Promotion

Samples synthesized using Eq. (9) are hard to fit for the full-precision model parameterized with θ, but their difficulties for the quantized model θ' are lacking. This stems from the fact that the difficulty used in Eq. (9) is measured with θ using $d(\mathbf{x}_i,\theta)$. Therefore, we propose to promote the sample difficulty using the quantized model parameterized with θ'.

The sample difficulty promotion is different from the process of hard sample synthesis. To be specific, the latter utilizes full-precision models to synthesize samples for training quantized models, while the former leverages quantized models to enhance the sample difficulty for training themself. If the synthetic samples are changed in the process of sample difficulty promotion, the synthesized samples would be both easy and difficult for quantized models. Thus, we draw inspiration from adversarial training [18, 49], where samples are modified to be hard to fit on the fly.

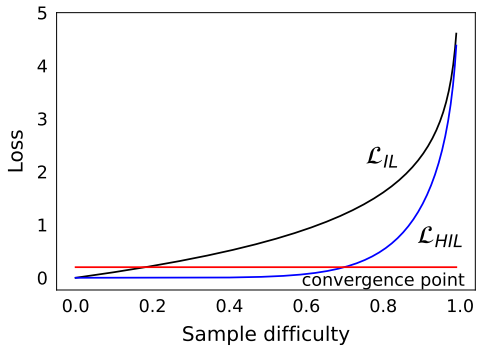


Figure 4. Curves of $\mathcal{L}_{IL}$ and $\mathcal{L}_{HIL}$.

More specifically, we increase the sample difficulty $d(\mathbf{x}_i,\theta')$ on the fly to make synthetic samples hard to fit by introducing a perturbation δ_i for $\mathbf{x}_i$ in each training iteration:

$$\delta_i = \arg\max_{||\delta'||_\infty \le \epsilon} d(\mathbf{x}_i+\delta_i',\theta'), \quad (10)$$

where $||\cdot||_\infty$ represents the ℓ_∞-norm and ϵ controls the strength of perturbation. We attach the perturbation δ_i for $\mathbf{x}_i$ to find nearby difficult samples with a larger sample difficulty in each iteration.

3.5. Minimizing Quantization Gap by Feature Alignment

Built upon the proposed hard sample synthesis and sample difficulty promotion, we are ready to train quantized models using hard samples. To ensure the quantized model performs the same as its full-precision model, an ideal result may be that the outputs of these two models (including intermediate features) are kept exactly the same, which shares the same spirit with [49] built upon a causal perspective. Along with this insight, we can align the features of the full-precision and quantized models:

$$\min_{\theta'} \mathcal{L}_{FA} \triangleq \frac{\lambda}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L} \phi^l(f^l(\mathbf{x}_i+\delta_i;\theta), f^l(\mathbf{x}_i+\delta_i;\theta')), \quad (11)$$

where $\phi^l(\cdot,\cdot)$ is a metric for the l-th layer measuring the difference of its inputs, e.g., mean square error, $f^l(\cdot;\theta)$ represents the feature at the l-th layer of a model parameterized with θ, and the hyper-parameter λ adjusts the order of magnitude of the loss.

The quantized models may not be able to match the outputs of full-precision models due to the limited model capac-

ity. Thus, we relax the feature alignment constraint:

$$\min_{\theta'} \hat{\mathcal{L}}_{FA} = \frac{\lambda}{z} \sum_{i,l \in S} (\phi(f^l(\mathbf{x}_i + \delta_i; \theta), f^l(\mathbf{x}_i + \delta_i; \theta')) + \alpha KL(p(\mathbf{x}_i; \theta) || p(\mathbf{x}_i; \theta'))), \tag{12}$$

where $z = N|S|$, S is the set of selected intermediate layers, $\phi(\cdot, \cdot)$ is the metric used for all intermediate layers, and KL divergence is employed as the metric for the final output layer, i.e., the predicted probability. For the metric used in intermediate layers, we instantiate them with an attention vector [47] rather than a mean square error ($||f^l(\mathbf{x}_i + \delta_i; \theta) - f^l(\mathbf{x}_i + \delta_i; \theta')||^2$). The metric of intermediate feature $f^l(\cdot; \theta) \in \mathbb{R}^{C \times W \times H}$ (C, W, and H is the number of channels, width, and height) is calculated using an attention vector $Att(\cdot)$ as follows:

$$\phi(f^l(\cdot; \theta), f^l(\cdot; \theta')) = ||Att(f^l(\cdot; \theta)) - Att(f^l(\cdot; \theta'))||_2^2, \tag{13}$$

where $Att(f^l(\cdot; \theta)) \in \mathbb{R}^C$ is a vector with the c-th element calculated as:

$$Att(f^l)(c) = \sum_{w=1}^{W} \sum_{h=1}^{H} f^l(c, w, h)^2, \tag{14}$$

where we denote $f^l(\cdot; \theta)$ by f^l for simplicity.

4. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our proposed HAST scheme. We organize this section as follows. We first provide the experiment setting in Sec. 4.1. Second, we compare our proposed method with existing ZSQ methods in Sec. 4.2 and Sec. 4.3. Last, we investigate the effect of the hyperparameters and the proposed component in our HAST scheme in Sec. 4.4. The code to reproduce our results is available [1].

4.1. Experimental Setup

Datasets. We evaluate our method on three datasets, including CIFAR-10 [26], CIFAR-100 [26] and ImageNet (ILSVRC2012) [27], which are commonly used in most ZSQ methods. We report top-1 accuracy of all methods.

Network Architectures. We choose ResNet-20 [21] for Cifar-10/100 and select ResNet-18 [21], MobileNetV1 [23] and MobileNetV2 [42] for ImageNet to evaluate our method. ResNet-20 and ResNet-18 are popular medium-sized models. MobileNetV1 and MobileNetV2 are widely-used lightweight models. All pretrained models are from pytorchcv library and all experiments are implemented with Pytorch [40].

Baselines. The compared methods based on noise optimization include ZeroQ [2], DSG [48], and IntraQ [50].

[1] https://github.com/lihuantong/HAST

Bit-width	Method	Generator	Acc.(%)
	full-precision	-	94.03
W4A4	Real Data	-	91.52
	SQuant [19]	-	92.24
	GDFQ [46]	✔	90.25
	AIT [7]	✔	91.23
	ZeroQ [2]+IL [20]	✘	89.66
	DSG [48]+IL [20]	✘	88.93
	IntraQ [50]	✘	91.49
	HAST(Ours)	✘	**92.36**±0.09
W3A3	Real Data	-	87.94
	SQuant [19]	-	79.19
	GDFQ [46]	✔	71.10
	AIT [7]	✔	80.49
	ZeroQ [2]+IL [20]	✘	69.53
	DSG [48]+IL [20]	✘	48.99
	IntraQ [50]	✘	77.07
	HAST(Ours)	✘	**88.34**±0.06

(a) Results on CIFAR-10.

Bit-width	Method	Generator	Acc.(%)
	full-precision	-	70.33
W4A4	Real Data	-	66.80
	SQuant [19]	-	63.96
	GDFQ [46]	✔	63.58
	AIT [7]	✔	65.80
	ZeroQ [2]+IL [20]	✘	63.97
	DSG [48]+IL [20]	✘	62.62
	IntraQ [50]	✘	64.98
	HAST(Ours)	✘	**66.68**±0.12
W3A3	Real Data	-	56.26
	SQuant [19]	-	40.36
	GDFQ [46]	✔	43.87
	AIT [7]	✔	48.64
	ZeroQ [2]+IL [20]	✘	26.35
	DSG [48]+IL [20]	✘	43.42
	IntraQ [50]	✘	48.25
	HAST(Ours)	✘	**55.67**±0.26

(b) Results on CIFAR-100.

Table 1. Results of ResNet-20 on CIFAR-10/100. WBAB indicates the weights and activations are quantized to B-bit.

The generator-based methods are also compared, including GDFQ [46], and AIT [7]. In addition, SQuant [19], an recent on-the-fly ZSQ method without synthesizing data, is also involved. For ZeroQ and DSG, we report the results with the inception loss (IL) [20]. Since AIT applies the method on three different generator-based methods, we report the best result. Note all the methods quantize all layers of the models to the ultra-low precisions, except SQuant which sets the last layer to 8-bit.

Implementation details. For data generation, the synthetic images are optimized by the loss function Eq. (9) using Adam optimizer with a momentum of 0.9 and the initial learning rate of 0.5. We update the synthetic images for 1,000 iterations and decay the learning rate by 0.1 each

Bit-width	Method	Generator	Acc.(%)
	full-precision	-	71.47
W4A4	Real Data	-	67.89
	SQuant [2]	-	66.14
	GDFQ [46]	✔	60.60
	AIT [7]	✔	66.83
	ZeroQ [2]+IL [20]	✘	63.38
	DSG [48]+IL [20]	✘	63.11
	IntraQ [50]	✘	66.47
	HAST(Ours)	✘	**66.91**±0.16
W3A3	Real Data	-	51.95
	SQuant [2]	-	25.74
	GDFQ [46]	✔	20.69
	AIT [7]	✔	36.70
	ZeroQ [2]+IL [20]	✘	44.68
	IntraQ [50]	✘	45.51
	HAST(Ours)	✘	**51.15**±0.27

Table 2. Results of ResNet-18 on ImageNet. WBAB indicates the weights and activations are quantized to B-bit.

time the loss of Eq. (9) stops decreasing for 50 iterations. For all datasets, the batch size is set to 256. We synthesize 5120 images following the settings of IntraQ [50]. For network fine-tuning, the quantized models are fine-tuned by the loss function Eq. (12) using SGD with Nesterov with a momentum of 0.9 and the weight decay of 10^{-4}. The batch size is 256 for CIFAR-10/100 and 16 for ImageNet. The initial learning rate is set to 10^{-5} and 10^{-6} for CIFAR-10/100 and ImageNet respectively. Both learning rates are decayed by 0.1 every 100 fine-tuning epochs and a total of 150 epochs are given. For data augmentation and hyper-parameters β in Eq. (9) and α in Eq. (12), we keep the same settings as [50] for fair comparison. In addition, there are three hyper-parameters in our method, including γ in Eq. (8), ϵ in Eq. (10) and λ in Eq. (12). They are respectively set to 2, 0.01, 1×10^3 for CIFAR-10; 2, 0.02, 2×10^3 for CIFAR-100 and 0.5, 0.01, 4×10^3 for ImageNet.

4.2. Comparison Results on CIFAR-10/100

In this section, We compare the performance on CIFAR-10/100 against existing ZSQ methods. From Table 1, our method achieves significant performance improvements on both CIFAR-10 and CIFAR-100. Specifically, compared to the advanced generator-based AIT, our method increases the top-1 accuracy of 3-bit quantized models by 7.85% on CIFAR-10 and 7.03% on CIFAR-100. When compared with IntraQ which obtains 3-bit accuracy of 77.07% and 48.25% on CIFAR-10 and CIFAR-100 respectively by exploiting the intra-class heterogeneity in the synthetic images, the proposed method reaches the higher performance of 88.34% and 55.67% using the same number of synthetic images. Notably, since the CIFAR-10 dataset is relatively easy when compared with CIFAR-100 and ImageNet, our method outperforms fine-tuning with real data. The gap is also only 0.59% when it comes to CIFAR-100. We obtain the similar results in 4-bit quantization, demonstrating the effectiveness of our proposed HAST.

Bit-width	Method	Generator	Acc.(%)
	full-precision	-	73.39
W5A5	Real Data	-	69.87
	SQuant [2]	-	64.20
	GDFQ [46]	✔	59.76
	AIT [7]	✔	-
	ZeroQ [2]+IL [20]	✘	67.11
	DSG [48]+IL [20]	✘	66.61
	IntraQ [50]	✘	68.17
	HAST(Ours)	✘	**68.52**±0.17
W4A4	Real Data	-	59.66
	SQuant [2]	-	10.32
	GDFQ [46]	✔	28.64
	AIT [7]	✔	-
	ZeroQ [2]+IL [20]	✘	25.43
	DSG [48]+IL [20]	✘	42.19
	IntraQ [50]	✘	51.36
	HAST(Ours)	✘	**57.70**±0.31

(a) Results of MobileNetV1.

Bit-width	Method	Generator	Acc.(%)
	full-precision		73.03
W5A5	Real Data	-	72.01
	SQuant [2]	-	66.83
	GDFQ [46]	✔	68.14
	AIT [7]	✔	**71.96**
	ZeroQ [2]+IL [20]	✘	70.95
	DSG [48]+IL [20]	✘	70.87
	IntraQ [50]	✘	71.28
	HAST(Ours)	✘	71.72±0.19
W4A4	Real Data	-	67.90
	SQuant [2]	-	22.07
	GDFQ [46]	✔	51.30
	AIT [7]	✔	**66.47**
	ZeroQ [2]+IL [20]	✘	60.15
	DSG [48]+IL [20]	✘	59.04
	IntraQ [50]	✘	65.10
	HAST(Ours)	✘	65.60±0.27

(b) Results of MobileNetV2.

Table 3. Results of MobileNetV1/V2 on ImageNet. WBAB indicates the weights and activations are quantized to B-bit

4.3. Comparison Results on ImageNet

We further compare with the competitors on the large-scale ImageNet. The quantized networks include ResNet-18 and MobileNetV1/V2. Similar to CIFAR-10/100, we quantize all layers of the networks. Differently, since MobileNetV1/V2 are lightweight models which suffer great performance degradation when quantized to ultra-low precision, we quantize MobileNetV1/V2 to 5-bit and 4-bit following the settings of most existing methods.

ResNet-18. Table 2 shows the experimental results of

ResNet-18. In the case of 4-bit, our HSAT (66.91%) slightly outperforms the IntraQ (66.47%) and AIT (66.83%). When it comes to 3-bit, our HSAT significantly surpasses AIT and IntraQ by 14.45% and 5.64% on ImageNet. Moreover, the performance of our method is very close to fine-tuning using real data regardless of 4-bit or 3-bit, with a gap of less than 1%. This indicates that our HSAT is able to mimic real data and generalize quantized model well even on large-scale and hard datasets.

MobileNetV1/V2. In Table 3, our method still outperforms almost all baselines except AIT on quantizing lightweight models. For example, while other methods achieve unexpectedly low performance on quantizing MobileNetV1 to 4-bit, our HSAT obtains 6.34% accuracy improvement when compared with IntraQ. However, Unlike other networks, we find that the fine-tuning process of MobileNetV2 is more unstable and the performance improvement during fine-tuning is smaller. This shows that the training process of MobileNetV2 has a greater impact on the performance recovery than the sample difficulty. AIT uses a dynamic learning rate for each convolution kernel to solve this problem, while we use a uniform fixed learning rate. Thus AIT performs better than HAST on MobileNetV2. It is worth mentioning that AIT can be used in combination with HAST as long as memory and time overheads allow.

In short, sufficient experiments over various network architectures demonstrate that the synthetic data generated by the proposed HSAT scheme is able to significantly improve the performance of the quantized model. The results also suggest that synthesizing and learning hard samples is important for improving the quantized model. Especially when model is quantized to lower bit-width, the effect of hard samples is more obvious on final performance.

4.4. Ablation Study

In this section, we conduct ablation studies of the hyper-parameters and investigate the effect of the three components of our method. We put the hyper-parameters ablation studies, further discussions and related experimental results in the appendix.

Effects of the proposed components. We further study the effectiveness of our proposed hard sample synthesis in Sec. 3.3, sample difficulty promotion in Sec. 3.4, and feature alignment in Sec. 3.5. Table 4 shows the experimental results. Note that when we use ZeroQ+IL as the baseline described in Sec. 3.1, we only obtain an accuracy of 44.68%. From Table 4, when the three components are individually added to synthesize and learn hard samples, the accuracy increases compared with the baseline. Among them, adversarial augmentation obtains a high accuracy improvement of 3.79%. This inspires us the importance of hard samples in training quantized models. Since attention transfer can help the quantized model learn more informative knowledge from the intermediate feature of the full-precision model, it still increases the performance by 1.39% without hard samples. Furthermore, the performance continues to increase when any two of them are used. When all of them are applied, we obtain the best performance of 51.15%.

HSS	SDP	FA	Acc.(%)
			44.68
✔			45.56
	✔		48.47
		✔	46.07
	✔	✔	50.22
✔	✔		49.17
✔		✔	47.94
✔	✔	✔	**51.15**

Table 4. Ablations on different components of our method. "HSS" indicates the hard sample synthesis, "SDP" indicates the sample difficulty promotion, and "FA" indicates the feature alignment. We report the top-1 accuracy of 3-bit ResNet-18 on ImageNet.

Limitation. In this work, we merely considered a limited number of scenarios. Thus, we will explore the potential power for more practical scenarios, such as federated learning [9, 45] (with limited communication bandwidth), vision transformer [10] (with large-scale model parameters), and the performance of quantized models over out-of-distribution scenarios [14].

5. Conclusion

In this paper, we investigate the state-of-the-art solutions based on noise optimization for zero-shot quantization and demonstrate that the synthetic samples are easy for the quantized model to fit, which harms the performance of the quantized model on real test data. Thus, hard samples matter a lot for zero-shot quantization. We achieve the goal by not only paying more attention to generating hard samples but also making samples harder to fit during fine-tuning. Feature alignment is applied in the fine-tuning process to help the quantized model learn hard samples better. Our method can achieve comparable performance with those fine-tuned using real data.

6. Acknowledgements

This work was partially supported by the Key-Area Research and Development Program of Guangdong Province 2019B010155002, Science and Technology Program of Guangzhou, China under Grants 202007030007, Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2017ZT07X183. YGZ and BH were supported by NSFC Young Scientists Fund No. 62006202 and Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652.

References

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, 2019. 1, 3

[2] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *CVPR*, 2020. 1, 2, 3, 4, 6, 7

[3] Peihao Chen, Dongyu Ji, Kunyang Lin, Weiwen Hu, Wenbing Huang, Thomas H. Li, Mingkui Tan, and Chuang Gan. Learning active camera for multi-object navigation. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1

[4] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas H Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *NeurIPS*, 2022. 1

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 3, 4

[6] Kanghyun Choi, Deokki Hong, Noseong Park, Youngsok Kim, and Jinho Lee. Qimera: Data-free quantization with synthetic boundary supporting samples. In *NeurIPS*, 2021. 1, 3

[7] Kanghyun Choi, Hyeyoon 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 *CVPR*, 2022. 1, 2, 6, 7

[8] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *CVPR*, pages 4022–4031, June 2020. 4

[9] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *CVPR*, pages 10164–10173, June 2022. 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 8

[11] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied AI: from simulators to research tasks. *IEEE Trans. Emerg. Top. Comput. Intell.*, 6(2):230–244, 2022. 1

[12] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *ICLR*, 2020. 3, 4

[13] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 1

[14] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *NeurIPS*, 2022. 8

[15] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[16] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[17] 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 *CVPR*, 2019. 3

[18] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 5

[19] 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 *ICLR*, 2022. 1, 3, 6

[20] Matan Haroush, Itay Hubara, Elad Hoffer, and Daniel Soudry. The knowledge within: Methods for data-free model compression. In *CVPR*, 2020. 2, 6, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[22] Xiangyu He, Jiahao Lu, Weixiang Xu, Qinghao Hu, Peisong Wang, and Jian Cheng. Generative zero-shot network quantization. In *CVPRW*, 2021. 3

[23] 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. 6

[24] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, 2018. 1

[25] Sangil Jung, Changyong Son, Seohyung Lee, JinWoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *CVPR*, 2019. 3

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *University of Toronto*, 2009. 6

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 6

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[29] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *CVPR*, 2021. 3

[30] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI*, 2019. 2, 4

[31] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *ICLR*, 2020. 3

[32] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021. 3

[33] 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 *ICCV*, 2021. 2

[34] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Yan Wang, Yongjian Wu, Feiyue Huang, and Chia-Wen Lin. Rotated binary neural network. In *NeurIPS*, 2020. 3

[35] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. In *CVPR*, 2021. 1, 3

[36] Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. Data-free knowledge transfer: A survey. *arXiv preprint arXiv:2112.15278*, 2021. 2

[37] Brais Martínez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. Training binary neural networks with real-to-binary convolutions. In *ICLR*, 2020. 3

[38] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *ICCV*, 2019. 1, 3

[39] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, 2019. 1

[40] Adam Paszke, Sam Gross, Francisco Massa, and Adam Lerer. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[41] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *CVPR*, 2020. 3

[42] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 6

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[45] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *International Conference on Machine Learning*, pages 21111–21132. PMLR, 2022. 8

[46] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *ECCV*, 2020. 1, 3, 4, 6, 7

[47] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 6

[48] 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 *CVPR*, 2021. 1, 3, 4, 6, 7

[49] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Schölkopf, and Kun Zhang. Causaladv: Adversarial robustness through the lens of causality. 2022. 5

[50] 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 *CVPR*, 2022. 1, 2, 3, 4, 6, 7

[51] Baozhou Zhu, H. Peter Hofstee, Johan Peltenburg, Jinho Lee, and Zaid Al-Ars. Autorecon: Neural architecture search-based reconstruction for data-free compression. In *IJCAI*, 2021. 1, 3

[52] Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian D. Reid. Training quantized neural networks with a full-precision auxiliary module. In *CVPR*, 2020. 3

[53] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian D. Reid. Structured binary neural networks for accurate image classification and semantic segmentation. In *CVPR*, 2019. 1