

# GenQ: Quantization in Low Data Regimes with Generative Synthetic Data

Yuhang Li<sup>1</sup>, Youngeun Kim<sup>1</sup>, Donghyun Lee<sup>1</sup>, Souvik Kundu<sup>2</sup>, and  
Priyadarshini Panda<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, Yale University

<sup>2</sup> Intel Labs, San Diego

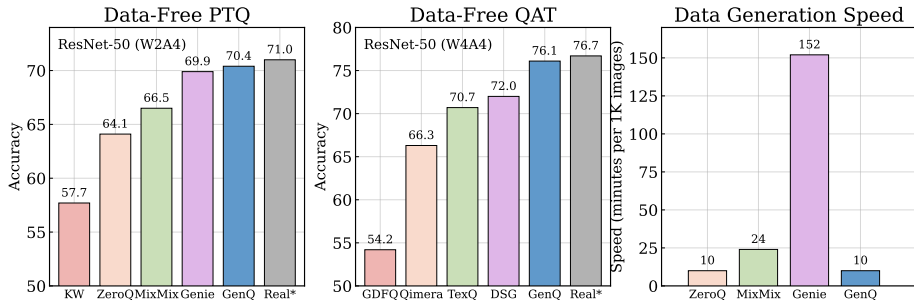
{yuhang.li, youngeun.kim, donghyun.lee, priya.panda}@yale.edu  
souvikk.kundu@intel.com

**Abstract.** In the realm of deep neural network deployment, low-bit quantization presents a promising avenue for enhancing computational efficiency. However, it often hinges on the availability of training data to mitigate quantization errors, a significant challenge when data availability is scarce or restricted due to privacy or copyright concerns. Addressing this, we introduce *GenQ*, a novel approach employing an advanced Generative AI model to generate photorealistic, high-resolution synthetic data, overcoming the limitations of traditional methods that struggle to accurately mimic complex objects in extensive datasets like ImageNet. Our methodology is underscored by two robust filtering mechanisms designed to ensure the synthetic data closely aligns with the intrinsic characteristics of the actual training data. In case of limited data availability, the actual data is used to guide the synthetic data generation process, enhancing fidelity through the inversion of learnable token embeddings. Through rigorous experimentation, *GenQ* establishes new benchmarks in data-free and data-scarce quantization, significantly outperforming existing methods in accuracy and efficiency, thereby setting a new standard for quantization in low data regimes. Code is released at <https://github.com/Intelligent-Computing-Lab-Yale/GenQ>.

## 1 Introduction

A variety of model compression techniques have been developed to deploy large deep learning models on embedded/mobile devices without significant accuracy drops. One prominent method is neural network quantization [18], which involves converting 32-bit floating-point models into a compact low-bit fixed-point format. This transformation leverages the efficiency of fixed-point computation and the benefits of reduced memory usage. Other notable techniques include pruning [5, 44] and knowledge distillation [24], both of which aim to reduce network size while preserving as much of the original model’s performance as possible.

While these methods effectively accelerate neural networks, they typically require finetuning on the original training dataset to mitigate any accuracy loss in the compressed model. However, accessing the original training data



**Fig. 1:** Comparison of *GenQ* with existing methods on ImageNet. (1) Data-Free PTQ, (2) Data-Free QAT, (3) Data Generation Speed. Real\* denotes using real ImageNet data in zero-shot quantization.

can be challenging due to privacy concerns or Intellectual Property (IP) protection issues [43]. This challenge has led to the emergence of *data-free quantization* [7, 21, 33]. This approach involves generating synthetic data from a pre-trained network and then finetuning the compressed or quantized model using this synthetic data, thereby circumventing the need for original training data.

Several prior works generate synthetic data by inverting the knowledge from pre-trained full-precision models. For instance, [7, 21, 69] propose to optimize the images by aligning them with the training data in terms of activation statistics. [39, 62] apply generative adversarial networks [20] to synthesize the images. However, these approaches often struggle to match the compression performance achievable with real training data. The primary challenge lies in the inherent complexity of reverse mapping from a lower to a higher dimension [35]. For example, in the context of the ImageNet dataset, this involves mapping from a 1000-dimensional space to an image space of  $224 \times 224 \times 3$ . Such transformation makes accurate recreation of complex objects a hard problem. Additionally, these methods cost a significant amount of time to generate the images.

In this paper, we introduce *GenQ*, a new pipeline utilizing the advanced Generative AI model such as Stable Diffusion [56] to generate high-quality data for quantization purposes. This approach aligns with the growing trend of applying AI-generated content (AIGC) in deep learning and vision applications [6, 58, 61, 67]. While direct application of generative AI data risks distribution shifts from original training data, we address this through two filtering mechanisms, ensuring the selection of in-distribution synthetic data. Our method excels in data-free quantization scenarios. More remarkably, in a majority of the model deployment scenarios with limited training data (*e.g.* one image per class), *GenQ* is significantly effective via prompt tuning to align generated images more closely with the real data. We conduct comprehensive experiments to validate the effectiveness of *GenQ*, including both Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT), on various neural network architectures (*e.g.* CNNs [22, 54] and ViTs [14]). Fig. 1 showcases effectiveness of *GenQ* on data-free quantization. Meanwhile, compared to other data generation ap-

proaches, our method achieves faster generation of synthetic data (up to  $15\times$  faster), increasing both efficiency and effectiveness.

We summarize our contributions as follows:

1. We propose *GenQ*, the first work to leverage advanced text-to-image synthetic data for quantization in data-free scenarios. To reduce the distribution gap between synthetic data and real data, we introduce a suite of filtering mechanisms to select in-distribution synthetic data, including energy score filtering and BatchNorm Distribution filtering.
2. In case of limited data availability, we also propose to learn the token embedding to guide the synthesis using real data.
3. We conduct extensive experiments to show the usefulness of *GenQ* in generating synthetic data that helps achieve SoTA quantization performance in both PTQ and QAT. For example, our 4-bit QAT-based ResNet-50 achieves 76.10% accuracy on the ImageNet dataset, outperforming the latest existing method [8] by 5.4%.

## 2 Related Work

**Quantization Methods.** Quantization approaches to compress pretrained Deep Neural Networks (DNNs) can be broadly divided into PTQ and QAT. PTQ performs *calibration* on a pre-trained DNN after quantizing its weights and activations to low bits to maintain the original accuracy [16, 41, 46]. For example, [3, 17] propose bias correction in the convolutional layers after quantization. [71] splits the outliers into additional channels to reduce the quantization error. Recently, a line of works [28, 34, 45, 60] leverage the weight rounding optimization to reconstruct the activation of the original model. While PTQ mostly adopts ad hoc strategies to prevent accuracy loss from quantization, QAT can significantly improve the accuracy of the quantized model through end-to-end finetuning. The success of QAT is mainly based on the usage of Straight-Through Estimator (STE) [65], enabling gradient-based optimization [9, 15, 27, 63, 68, 73]. QAT also explores optimizing the step size [15], clipping threshold [9], non-linear interval [33], non-uniform quantization levels [68] together with the weights.

**Data in Quantization.** To quantize a model to below 8 bits, data is essential, especially in QAT where end-to-end finetuning occurs. When access to real images is restricted due to privacy and copyright issues, ZeroQ [7] proposes to synthesize images in replacement of real data. There are two categories of creating synthetic images: (1) Inverting the images directly via gradient descent [7, 21, 32, 35, 69], and (2) Learning a Generative Adversarial Network (GAN) to continuously generate images [39, 62].

The image inversion [64] relies on Batch Normalization (BN) [7, 21] as an optimization metric to distill the data and obtain high-fidelity images. Other inversion methods based on loss function [72], model ensembling [35], and advanced convolutional layers [32] have been proposed to obtain better synthetic data. However, the inversion approach is limited in certain ways: (1) Inverting images adds a significant computation burden, which makes QAT with synthetic

data prohibitive; (2) Models that do not have BN layers, such as ViTs, cannot invert images [14]. The second category proposes to use GAN as an image synthesis engine [10, 39, 49, 62, 72]. These works finetune the generator and the quantized model simultaneously.

Our method opens a third category for quantization in low data regime, *i.e.*, using text-to-image diffusion models. Additionally, we investigate how to use limited real data to guide the generation of synthetic data. To the best of our knowledge, this scenario is being studied for the first time.

**Synthetic Data in Deep Learning.** Initial studies, such as those by [4, 31, 70], explored the use of GANs for assisting DNNs in classifier tuning, object segmentation, and contrastive learning. Recently, the emergence of text-to-image models [13, 42, 52, 53] has revolutionized the synthesis of high-quality data in deep learning, owing to their effectiveness and efficiency. For example, [23] utilized GLIDE [47] for image synthesis in classifier tuning on the CLIP model [51]. In a more advanced application, StableRep [58] leverages Stable Diffusion to generate datasets for contrastive learning, using synthetic data from different seeds as positive pairs. Meanwhile, [2] investigates the synthesis of data within the ImageNet label space observing accuracy improvements. Furthermore, Fill-up [55] employs synthetic data from text-to-image models to balance the long-tail distribution in training datasets.

### 3 Preliminaries

#### 3.1 Quantization

Uniform quantization maps full-precision weights into fixed-point numbers. For a step size  $s \in \mathbb{R}$ , the integer value of weights  $\mathbf{w}$  are given by

$$\mathbf{w}_{\text{int}} = \text{clip} \left( \left\lfloor \frac{\mathbf{w}}{s} \right\rfloor + \mathbf{z}, n, p \right), \quad (1)$$

where  $\lfloor \cdot \rfloor$  denotes the rounding-to-nearest operation, and  $n$  and  $p$  represent the lower and upper bound of the integers, respectively. For example, under the  $b$ -bit uniform quantization,  $n$  and  $p$  are set to 0 and  $2^{b-1}$ . The zero-point vector  $\mathbf{z}$  is an all- $z$  vector, where  $z = -\lfloor \frac{\min(\mathbf{w})}{s} \rfloor$ . Thus, the quantized weights  $\mathbf{w}^q$  are given by

$$\mathbf{w}^q = s(\mathbf{w}_{\text{int}} - \mathbf{z}). \quad (2)$$

**Quantization-Aware Training (QAT).** We utilize the Learned Step Size Quantization (LSQ) [15] to update the step size  $s$  and  $\mathbf{w}$  with gradient descent. The gradient to step size can be computed with Straight-Through Estimator (STE) [65], *i.e.*,  $\frac{\partial \lfloor x \rfloor}{\partial x} = 1$ . QAT requires a significant amount of data and GPU resources to perform end-to-end fine-tuning.

**Post-Training Quantization (PTQ).** Under this PTQ regime, the training data is much less than QAT, hindering the learning of step size. To obtain the desired step size  $s$  under PTQ, we perform layerwise quantization-error minimization [3]. In addition, we also conduct layer-wise tuning of the weight rounding

during PTQ as did in [32, 34, 45, 60], which can significantly improve the model performance with only 1K images.

### 3.2 Stable Diffusion Data Generation

Stable Diffusion utilizes the Denoising Diffusion Probabilistic Model (DDPM) [25] to implement the training and inference. It contains a text encoder  $\tau_\theta(\cdot)$ , image encoder  $\mathcal{E}(\cdot)$  and image decoder  $\mathcal{D}(\cdot)$ , and finally a denoising U-Net  $\epsilon_\theta(\cdot)$ .

**Inference Process.** During inference, a random noise image latent is sampled from Gaussian distribution  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Suppose the text prompt is  $\mathbf{s}$ , the denoising U-Net fuses the text embedding  $\tau_\theta(\mathbf{s})$  and the image and visual embedding through cross-attention layers and denoises the image latent gradually. After  $T$  timesteps of denoising, the image latent is decoded by  $\mathcal{D}(\cdot)$  to generate the high-resolution image.

**Training process.** Training the denoising U-Net essentially involves encoding the images first ( $\hat{\mathbf{x}}_0 = \mathcal{E}(\mathbf{x})$ ), and then diffusing the image through a Markov chain process  $q(\hat{\mathbf{x}}_t | \hat{\mathbf{x}}_{t-1}) = \mathcal{N}(\hat{\mathbf{x}}_t; \sqrt{1 - \beta} \hat{\mathbf{x}}_{t-1}, \beta \mathbf{I})$ . Note that  $t \in [1, T]$  is the number of diffusion timesteps and  $\beta$  is a hyper-parameter to control the perturbation. When  $t = T$ , then  $\hat{\mathbf{x}}_t = \mathbf{z}$ . Now the denoising U-Net can be trained by a reverse process by matching the output and the noises added before:

$$\min_{\theta} \mathbb{E}_{t \sim \mathcal{U}(1, T), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \lambda(t) \|\epsilon - \epsilon_\theta(\hat{\mathbf{x}}_t, t)\|_F^2, \quad (3)$$

where  $\lambda(t)$  is a positive weighting function [25],  $\epsilon$  is a noise vector predicted from  $\mathbf{x}_t$ . We use Stable Diffusion v1-5 [56] in our pipeline. We refer to [52] for more details about its principle.

## 4 Methodology

In this section, we introduce our methodology to synthesize the training data with Stable Diffusion. We discuss two quantization scenarios in low data regimes: (1) Data-Free Quantization (DFQ) where only the label space and a pre-trained full-precision model are provided, and (2) Data-Scarce Quantization (DSQ) where limited training data is provided in addition to the pre-trained model and label space. We first discuss GenQ’s data generation method under DFQ and DSQ and then introduce the quantization techniques we used.

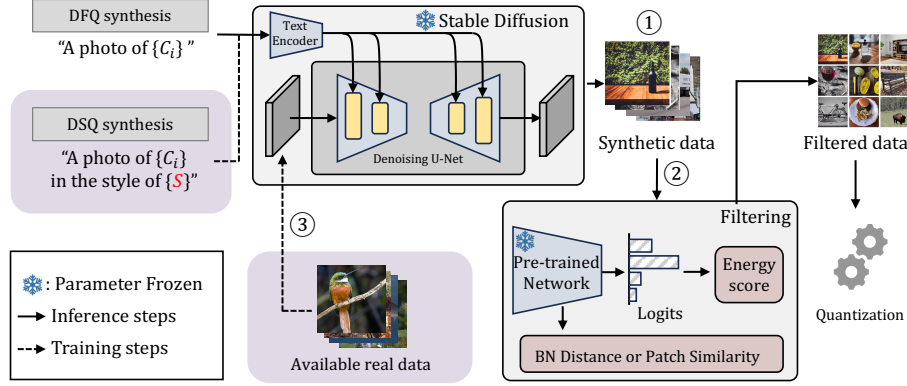
### 4.1 Data-Free GenQ

We propose a *label prompt* method that only uses the label as the prompt to synthesize the images without accessing the pre-trained model. For example, for an ImageNet pre-trained model [12], we can synthesize the images based on 1000 class names.

$$\text{Prompt} = \text{"A photo of a \{D\} \{C\}"}, \quad (4)$$

where  $\{D\}$  is a template adjective (*e.g.*, “nice, dark, small”) derived from the CLIP ImageNet template [51]<sup>3</sup>.  $\{C\}$  is a randomly chosen class label name. As

<sup>3</sup> We list the full template in Appendix A



**Fig. 2:** The overall image synthesis and filtering procedure of *GenQ*. ① For DFQ synthesis, we directly use the label as the text prompt, and ② we use several metrics to filter our out-of-distribution synthetic images. ③ For data-scarce synthesis, real images are used to optimize prompt {S} (Eq. (11)). We then generate the synthetic images with the optimized prompt.

an example, suppose we choose the class `hamster`, then a random prompt can be generated as "A photo of a small hamster". Subsequently, we directly use the text-to-image model like Stable Diffusion [52] to synthesize the data.

In our *label prompt* method, we generate the images relying solely on the label name of each class and the ability of Stable Diffusion (Fig. 2①). However, this operation does not leverage any prior knowledge in the original training data, which might cause a distribution shift from original training data to synthetic training data. To generate better-quality data for quantization, we propose *model-dependent selection* to leverage the prior knowledge embedded in the pre-trained full-precision model.

We propose a set of filtering mechanisms for model-dependent selection. We argue that the synthetic data selection can be treated as an Out-of-Distribution (OOD) detection problem [48]. However, contrary to OOD detection, we intend to select in-distribution data from the generated synthetic data for quantization purposes (Fig. 2②).

**Energy Filtering.** We use energy filtering to calculate the energy score [38] as,

$$E(\mathbf{x}, f) = -\alpha \sum_{i=1}^C e^{-f_i(\mathbf{x})/\alpha}, \quad (5)$$

where  $f_i(\mathbf{x})$  denotes the  $i^{th}$  value of the network’s output logits.  $C$  and  $\alpha$  denote the number of image classes and temperature, respectively. [38] has shown that the energy score is an OOD detector due to its theoretical connection with the likelihood function. The neural networks trained by cross-entropy loss inherently decrease the energy of the training data, hence, the OOD data will have relatively higher energy.

To select synthetic images that have a similar quantization effect to that of real image data, we pass the synthetic data through the full-precision pre-trained model. Then, we calculate the energy scores of all synthetic images and only select those images for quantization that yield energy scores lower than a certain user-defined threshold. We will experiment with the choice of this threshold in [Sec. 5.4](#).

**BatchNorm Distribution Filtering.** Originally proposed in [7, 64], the channel mean and channel variance distance computed in the BN layers [30] are regarded as a standard metric for optimizing the synthetic data in Convolutional Neural Networks (CNNs). We use this metric to evaluate the distance between synthetic images and original training images on the full-precision pre-trained model. We calculate the BN distance by

$$D_{BN} = \sum_{\ell=1}^n (\|\mu_{\ell}^s - \mu_{\ell}\|_F + \|\sigma_{\ell}^s - \sigma_{\ell}\|_F), \quad (6)$$

where  $\mu_l, \sigma_l$  denote the running mean and variance of the layer  $l$  activation from original training images, and  $\mu_l^s, \sigma_l^s$  are the current mean and variance of the layer  $l$  activation from synthetic images. Naively, for each generated synthetic data, we can evaluate the BN distance  $D_{BN}$  and filter out the ones with large  $D_{BN}$ . However, we find this approach does not bring improvement in the final data quality due to the difference between single data and batched data. Evaluating  $D_{BN}$  on a single image will lead to biased estimation as it ignores its interaction with other data. Ioffe [29] also demonstrates that with small batch sizes, the estimation of the batch mean and variance used during training become a less accurate approximation of the mean and variance used for testing.

To deal with this problem, we define a *BN sensitivity* metric, which measures the independence of one image from other images in a batch. Formally, given a batch of input images  $\{\mathbf{x}_i\}_{i=1}^B$  where  $B$  indicates the batch size, we define the BN sensitivity of the  $i$ -th image  $S(\mathbf{x}_i)$  as

$$S(\mathbf{x}_i) = D_{BN}(\{\mathbf{x}_j\}_{j=1}^B) - D_{BN}(\{\mathbf{x}_j\}_{j=1, j \neq i}^B). \quad (7)$$

Here, the BN sensitivity indicates the change in BN distance after removing the selected image. As such, a large BN sensitivity means the current image could potentially damage the internal distribution when it is batched with other images.

**Patch Similarity Filtering for ViTs.** As we discussed in [Sec. 2](#), since ViTs do not have BN layers, we adopt the patch similarity metric in [36] to do additional filtering of the synthetic images after the energy filtering. Formally, given the output features tensor  $\mathbf{o}$ , we first calculate the cosine similarity matrices between any two sub-tensors in the patch dimension, given by

$$\Gamma(\mathbf{o}_i, \mathbf{o}_j) = \frac{\mathbf{o}_i \cdot \mathbf{o}_j}{\|\mathbf{o}_i\| \|\mathbf{o}_j\|}, \quad (8)$$

where  $\mathbf{o}_i$  denotes the  $i$ -th patch of the output featuremap. After calculation, we get the  $N \times N$  ( $N$  is the #patches) matrix  $\mathbf{\Gamma}$ . To measure the diversity of the

patch similarities, we calculate differential entropy as follows

$$H = - \int \hat{f}_h(x) \cdot \log[\hat{f}_h(x)] dx. \quad (9)$$

where  $\hat{f}_h(x)$  is the continuous probability density function of  $\mathbf{\Gamma}$ , which can be obtained by the kernel density estimation [36]. A low entropy value indicates that the patch similarity distribution is less diverse. Hence, we can select the synthetic data that has the highest diversity in patch similarity.

**Filtering Pipeline.** The overall filtering pipeline is a two-stage procedure. First, regardless of the type of pre-trained model, we apply energy score filtering. Second, based on the category of the model (CNNs or ViTs), we either apply BatchNorm Distribution Filtering or Patch Similarity Filtering. The selected synthetic data is then used for quantization.

## 4.2 Data-Scarce *GenQ*

In this section, we describe how to synthesize and select synthetic data given a limited amount of training data (*i.e.*, data-scarce quantization). This case could be more common than the DFQ scenario as in practice it is not hard to obtain some permitted training data. We hypothesize that synthesizing a number of images based on this limited dataset can enhance quantization performance.

Considering the ImageNet-1k dataset [12] as the original training dataset, we assume that one image per class (*i.e.*, 1-shot) can be accessed by the cloud server. In this case, we propose to synthesize the images utilizing the existing information from the training data. Specifically, we optimize a text token embedding  $\{\mathbf{S}\}$  with the prompt shown below

$$\text{Prompt}_{[i]} = \text{"A photo of a } \{\mathbf{C}_{[i]}\} \text{ in the style of } \{\mathbf{S}\}\text{"}. \quad (10)$$

Here, the learnable token embedding  $\{\mathbf{S}\}$  indicates the dataset characteristics of ImageNet. To optimize this learnable token embedding, we associate each class name and the corresponding image into pairs  $(\{\mathbf{C}_{[i]}\}, \mathbf{x}_{[i]})$ , where  $[i]$  is the class index, and let the Stable Diffusion generate  $\mathbf{x}_{[i]}$  given  $\text{Prompt}_{[i]}$ . The optimization objective is given by:

$$\mathbb{E}_{t \sim \mathcal{U}(1, T), i \sim \mathcal{U}(1, M), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \lambda(t) \|\epsilon - \epsilon_{\theta}(\mathbf{x}_{[i]t}, t, \text{Prompt}_{[i]})\|_F^2, \quad (11)$$

where  $M$  is the number of all object classes. This method learns the token across multiple objects, aiming to characterize the whole dataset<sup>4</sup>. Fig. 2③ illustrates the data-scarce prompt optimization. After optimization, we directly used the learned token embedding and the class label to generate images and filter the output images using the same technique in Fig. 2②.

<sup>4</sup> We provide more technical details in Appendix A.



### 4.3 Quantization with Synthetic Data from *GenQ*

The selected synthetic data from *GenQ* is then used to quantize a full-precision pre-trained model using QAT or PTQ. For PTQ, we adopt the state-of-the-art reconstruction-based rounding optimization methods [32, 34]. As for QAT, we propose to finetune the quantized network on top of a PTQ model. Specifically, after the rounding optimization in PTQ, the quantization becomes

$$\mathbf{w}_{\text{int}} = \text{clip} \left( \left\lfloor \frac{\mathbf{w}}{s} \right\rfloor + \text{sgn}(\mathbf{v}) + \mathbf{z}, n, p \right), \quad (12)$$

where  $\text{sgn}(\mathbf{v})$  is the learned rounding indicating up or down, which is already well-optimized in the PTQ reconstruction stage. To initialize from the PTQ model and stabilize the training process, we freeze all previous learnable variables including  $\mathbf{w}$  and  $\mathbf{v}$ , and reinitialize an all-zero vector  $\mathbf{u}$ . The new quantization function is thus given by

$$\mathbf{w}_{\text{int}} = \text{clip} \left( \left\lfloor \frac{\mathbf{w}}{s} \right\rfloor + \text{sgn}(\mathbf{v}) + \left\lfloor \frac{\mathbf{u}}{s} \right\rfloor + \mathbf{z}, n, p \right). \quad (13)$$

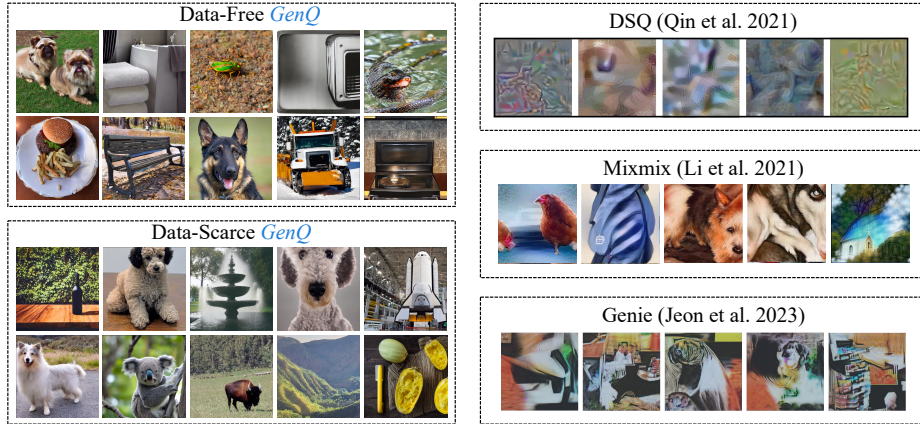
The de-quantization step remains the same with Eq. (2). By introducing an additional variable  $\mathbf{u}$  and freezing the previous variable, we start from the original PTQ model and avoid passing the gradient through floor operation. As a result, the  $\mathbf{u}$  can be safely updated through STE in finetuning. Note, during PTQ or QAT, we only use the synthetic data from *GenQ* to perform quantization.

## 5 Experiments

In this section, we empirically demonstrate the effectiveness and the efficiency of our *GenQ* synthetic data in both a qualitative and a quantitative way. For the PTQ scenario, we follow the conventional setup [32, 34, 35] to synthesize 1k images. As for the QAT scenarios, we synthesize 1.2M images with 1200 images in each object class to match the original ImageNet dataset volume. Unless specified, we use Stable Diffusion v1-5 [1] and set the guidance scale to 3.5. We will first provide the visualization of our synthetic data, the latency comparison of generating the data, and then compare them against existing state-of-the-art methods across multiple setups. Finally, we analyze *GenQ* by conducting various ablation studies. All experiments and accuracy noted are for the ImageNet dataset.

### 5.1 Analysis on Synthetic Data

**Visualization.** As shown in Fig. 3, the *GenQ* images after filtering are high-quality and visually similar to real images. To compare the synthetic data with existing data-free quantization, we also provide the images from [32, 35, 50]. It can be observed that our method can generate much higher quality synthetic data than previous inversion-based methods. For example, [50] only demonstrates



**Fig. 3:** Visualization of DF/DS *GenQ* and existing data synthesis method for quantization in low data regimes.

**Table 1:** Comparison of synthetic data generation approaches w.r.t. (a) generation speed and (b) data quality.

(a) Latency				(b) BatchNorm Distance		
Method	Type	Latency	Reusable	Data Source	Res50	MBV2
GENIE [32]	Image inversion	152 min	✓	GENIE [32]	$0.014 \pm 0.001$	$1.03 \pm 0.296$
TexQ [8]	GAN-based	10 min	✗	TexQ [8]	$0.045 \pm 0.013$	N/A
<i>GenQ</i>	Text-to-image	24 min	✓	<i>GenQ</i>	$0.024 \pm 0.002$	$0.78 \pm 0.106$

certain textures rather than objects, [35] shows some objects but the scene shows less resolution and clarity.

**Data Synthesis Cost.** We further study the cost of each synthetic data generation method. Specifically, we measure the latency of Genie [32], TexQ [8], and our method corresponding to image inversion, GAN-based, and text-to-image synthesis approaches, respectively, for 1000 images on the ImageNet dataset. The pre-trained model used for filtering or input optimization is ResNet-50. We report the latency in Table 1a, from which we can find using text-to-image synthesis provides faster data generation than image inversion-based techniques. Although GAN-based data generation is faster than *GenQ*, their data is not reusable across model types, which increases the overall latency if many models have to be quantized.

**Data Quality Assessment.** We further measure the quality of the synthetic data through some quantitative evaluation metrics. Specifically, we measure the average BN distances of the generated data as well as the original training data on ResNet-50 and MobileNetV2. The results are shown in Table 1b, from which we can find that *GenQ* largely closes the gap between synthetic data and real training data.

**Table 2:** Evaluation of data-free PTQ on CNN models (top-1 accuracy (%)).

Quant. Method	Syn. Method	#Bits W/A	Res18	Res50	MBV2	MB-b	MNas-1
Full Prec.	N/A	32/32	71.08	77.00	72.49	74.53	73.52
BRECQ [34]	Real Data	4/4	69.62	75.45	68.84	-	-
	ZeroQ <sup>‡</sup> [7]		69.32	73.73	49.83	55.93	52.04
	KW <sup>‡</sup> [21]		69.08	74.05	59.81	61.94	55.48
	IntraQ [72]		68.77	68.16	63.78	-	-
	Qimera [10]		67.86	72.90	58.33	-	-
	MixMix <sup>‡</sup> [35]		69.46	74.58	64.01	65.38	57.87
	GENIE-D		69.40	75.35	67.81	64.24	65.02
	<i>GenQ</i>		<b>69.52</b>	<b>75.47</b>	<b>68.34</b>	<b>67.08</b>	<b>66.33</b>
-----							
GENIE-M* [32]	Real Data	4/4	69.82	75.51	69.11	69.26	68.40
	GENIE-D [32]		69.72	<b>75.61</b>	68.62	67.31	67.03
	<i>GenQ</i>		<b>69.77</b>	75.50	<b>68.96</b>	<b>68.74</b>	<b>68.06</b>
BRECQ [34]	Real Data	2/4	65.25	70.65	54.22	-	-
	ZeroQ [7]		61.63	64.16 <sup>‡</sup>	34.39	23.53	13.83
	KW <sup>‡</sup> [21]		-	57.74	-	-	-
	IntraQ [72]		55.39	44.78	35.38	-	-
	Qimera [10]		47.80	49.13	3.73	-	-
	MixMix <sup>‡</sup> [35]		-	66.49	-	-	-
	GENIE-D [32]		63.93	69.72	49.75	38.01	45.53
	<i>GenQ</i>		<b>65.04</b>	<b>69.90</b>	<b>53.08</b>	<b>47.31</b>	<b>50.84</b>
-----							
GENIE-M* [32]	Real Data	2/4	66.05	70.96	56.42	55.00	54.66
	GENIE-D [32]		64.86	69.89	51.47	47.69	48.38
	<i>GenQ</i>		<b>65.72</b>	<b>70.35</b>	<b>54.82</b>	<b>52.77</b>	<b>52.76</b>

<sup>‡</sup> The figures are taken from [35].

\* Denotes our implementation based on the open-source code.

## 5.2 Quantization Performance Evaluation

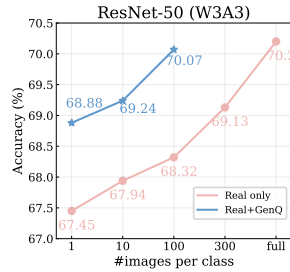
In this section, we test the quantization accuracy with our synthetic data and compare it with various existing methods. We will introduce the experiment setup in each scenario individually.

**Comparisons on Data-Free PTQ.** We start with evaluating our proposed method by testing it on data-free PTQ including (1) CNNs such as ResNet [19], MobileNet [26, 54], and MnasNet [57] and (2) Vision Transformers like original ViT [14]. For PTQ on CNNs, we employ the state-of-the-art PTQ algorithms, BRECQ [34] and GENIE-M [32] to perform the W4A4 and W2A4 quantization, which reconstruct the activation output in a block-wise manner. In this case, we use 1024 synthetic images for quantization evaluation. For PTQ on ViT, we perform PTQ4ViT [66] and RepQ-ViT [37] across 5 different models, including ViT-M [14], DeiT-M [59], and Swin-B [40]. We follow the open-source code implementation and generate 32 images as the calibration dataset to obtain the quantized ViTs. For all cases, we generate 2× synthetic data and filter 50% of them to match the volume.

For CNNs, we select SOTA existing methods including ZeroQ [7], the Knowledge Within [21], IntraQ [72], Qimera [10], MixMix [35], GENIE-D [32]. For reference, we also include the performance of PTQ using 1024 real ImageNet-1k images. We summarize all the accuracy results in Table 2. It can be observed that our *GenQ* achieves the highest accuracy in most cases. Interestingly, we

**Table 3:** Evaluation of data-free QAT on CNN models (top-1 accuracy (%)).

Method	#Real Data	#Syn Data	#Bits W/A	Res18	MBV2	Res50
LSQ [15]	1.2M	0	4/4	71.10	69.50	76.70
GDFQ [62]	0	1.2M		60.60	59.43	54.16
ZAQ [39]	0	4.6M		52.64	0.10	53.02
Qimera [10]	0	1.2M		63.84	61.62	66.25
IntraQ [72]	0	5k	4/4	66.47	65.10	-
ARC+AIT [11]	0	1.2M		65.73	66.47	68.27
DSG [50]	0	1.2M		62.18	60.46	71.96
AdaDFQ [49]	0	1.2M		66.53	65.41	68.38
TexQ [8]	0	1.2M		67.73	67.07	70.72
<i>GenQ</i>	0	1.2M		<b>70.03</b>	<b>69.65</b>	<b>76.10</b>
LSQ [15]	1.2M	0	3/3	70.20	65.30	75.80
GDFQ [62]	0	1.2M		20.23	1.46	0.31
AdaDFQ [49]	0	1.2M	3/3	38.10	28.99	17.63
TexQ [8]	0	1.2M		50.28	32.80	25.27
<i>GenQ</i>	0	1.2M		<b>68.18</b>	<b>59.15</b>	<b>73.99</b>

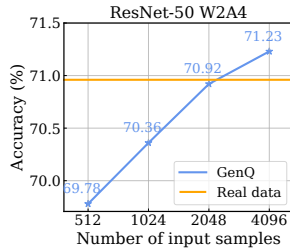
**Fig. 4:** Evaluation of data-scarce *GenQ* .**Table 4:** Evaluation of data-free PTQ on ViTs (top-1 accuracy (%))

Quant. Method	Syn. Method	#Bits W/A	ViT-B	DeiT-B	Swin-B
Full Prec.	N/A	32/32	84.54	81.80	85.27
	Real Data		58.07	63.57	75.20
PTQ4ViT [66]	PSAQ-ViT [36]	4/4	60.26	65.18	72.12
	<i>GenQ</i>		<b>63.17</b>	<b>67.08</b>	<b>74.27</b>
	Real Data		67.93	75.99	72.80
RepQ-ViT [37]	PSAQ-ViT [36]	4/4	60.18	74.69	54.21
	<i>GenQ</i>		<b>67.50</b>	<b>76.10</b>	<b>70.08</b>

find that the network architecture affects the performance of synthetic data. On ResNet architectures, *GenQ* and GENIE-D have similar accuracy (0.1-0.5%) levels, however, *GenQ* largely outperforms other data-free algorithms on lightweight network architectures due to their lower quantization resilience. As an example, *GenQ* increases 5% accuracy compared to GENIE-D on MobileNet-b W2A4 quantization and 4.4% accuracy on MnasNet W2A4 quantization.

Given that there is only one data-free algorithm for ViTs, PSAQ-ViT [36], we compare *GenQ* with PSAQ-ViT in Table 4. *GenQ* consistently outperforms PSAQ-ViT [36]. For example, on ViT-B and Swin-B, our method has 7.3% and 15% accuracy improvement using the RepQ-ViT method. This result proves that *GenQ* can achieve state-of-the-art performance on both CNNs and ViTs.

**Comparisons on Data-Free QAT.** We then compare *GenQ* with existing data-free QAT methods, such as GDFQ [62], ZAQ [39], Qimera [10], IntraQ [72], ARC [11], AdaDFQ [49], TexQ [8]. Note that these methods jointly optimize GAN and quantized model. Thus, they can generate *unlimited* synthetic data during finetuning. We originally have 1.6M synthetic images (1600 images/class), and then filter 0.4M (400 images/class) images using energy score and BN distance filtering. We use the SGD optimizer with a learning rate of 0.001 followed



**Fig. 5:** Accuracy vs # syn. data with data-free PTQ using Genie-M.

**Table 5:** Cross-model evaluation on synthetic data. We compare the data transferability between (a) existing methods and (b) *GenQ*.

(a) GENIE-D & PSAQ-ViT			(b) <i>GenQ</i>		
Source model	Res18	MBV2	Source model	Res18	MBV2
Res18	64.86	48.37	Res18	65.72	54.32
MBV2	64.21	51.47	MBV2	65.52	54.82
ViT-B-16	47.27	17.96	ViT-B-16	65.44	54.24

by a cosine annealing decay schedule for 50 epochs of QAT. Additionally, we report the LSQ results using 1.2M real ImageNet data. The results are summarized in Table 3. Our method largely outperforms other data-free QAT methods and nearly approaches the accuracy of LSQ baseline. Remarkably, *GenQ* improves the accuracy of W3A3 quantization by a large margin, for instance, *GenQ* exceeds the accuracy of TexQ, the best-performing method, by 48% on ResNet-50.

**Comparisons on Data-Scarce QAT.** In this section, we evaluate the data-scarce QAT scenarios. Our main comparison is LSQ [15] using different amounts of real data. We show how much of the real data can our *GenQ* generated synthetic data match in practice. We experiment this on a ResNet-50 with W3A3 quantization. We initialize the QAT model by performing GENIE-M on 1k real data and then finetune the model using {1k, 10k, 100k, 300k, 1.2M} real data, corresponding to 1-shot, 10-shot, 100-shot, 300-shot, and full-shot data regimes, respectively. To generate *GenQ* data, we initialize the token embedding as ImageNet, and train the embedding for 50k iterations, (see Appendix A for more details).

As demonstrated in Fig. 4, the performance of QAT is highly correlated with the number of real data. Our method can boost the performance of QAT under the low data regime. For example, when only 1k real images are provided (*i.e.*, 1-shot), *GenQ* can achieve 68.88% ImageNet accuracy, similar to the QAT performance that needs  $300\times$  more training data.

### 5.3 Data Transferability Evaluation

It is demonstrated that the synthetic data extracted from one model has low transferability on other models [35]. To demonstrate that *GenQ* has relatively high transferability, we conduct a cross-model evaluation, *e.g.*, synthesizing images based on model A, and evaluate the quantization on model B. In Table 5a we provide the results of GENIE-D and PSAQ-ViT, tested on data-free PTQ W2A4 scenarios. We then summarize the results of *GenQ* in Table 5b. We use Genie-M as our PTQ method for CNN architectures and PASQ-ViT for ViT. We observe that the *GenQ* data, although filtered by a different model, only drops 0.2-0.5% final accuracy, while the existing methods drop 0.6-33% accuracy, especially when using ViT synthetic data for MobileNetV2 (MBV2) quantization.

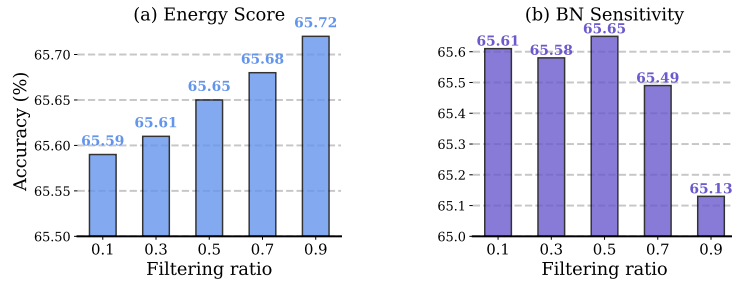


Fig. 6: Accuracy vs filtering strength with data-free PTQ.

#### 5.4 Ablation study

In this section, we conduct ablation studies on two variables, (1) the number of synthetic data, and (2) the image filtering strength.

**Number of synthetic data in PTQ.** In practice, synthesizing images using *GenQ* requires very low latency (1 image/second). Hence, we can safely increase the number of synthetic data with minimal overhead in PTQ cases. Yet retrieving more real data seems much more difficult if the training data is private or under IP protection. In Fig. 5, we show that increasing the synthetic data can consistently improve the PTQ performances of W2A4 quantized ResNet-50. *GenQ* even outperforms the quantized model baseline (optimized using PTQ on the 1k real training data) when using only 4k synthetic input images.

**Energy Score Filtering Strength.** Given a fixed number of synthetic data ( $N$ ) used for quantization and a percentage  $r$  representing how many images are filtered out, the total amount of generated images is  $\frac{N}{1-r}$ . We test different options of  $r$  from  $\{0.1, 0.3, 0.5, 0.7, 0.9\}$  to test the effect of energy score filtering. We choose ResNet-18 and use W2A4 data-free PTQ (with Genie-M) for evaluation. The test performance is shown in Fig. 6(a). We generally find that a higher filtering ratio leads to better test accuracy. However, it will also increase the number of generated images. Nevertheless, the accuracy variation is rather small.

**BN Sensitivity Filtering Strength.** We run the same test for the BN sensitivity filtering mechanism. The results are shown in Fig. 6(b). Unlike the energy score, setting the ratio to 50% has the best performance. We hypothesize that the networks need more diverse synthetic data to effectively quantize a model.

## 6 Conclusion

In this paper, we have introduced *GenQ*, a novel attempt to synthesize images with text-to-image models for data-scarce quantization. Our *GenQ* generates images in both data free (no real data) and data scarce (few real data) regimes and refines the images with several filtering mechanisms and a token embedding learning algorithm. Extensive experiments show that *GenQ* establishes a new state of the art in both PTQ and QAT.

## Acknowledgment

This work was supported in part by CoCoSys, a JUMP2.0 center sponsored by DARPA and SRC, the National Science Foundation (CAREER Award, Grant #2312366, Grant #2318152), and the DoE MMICC center SEA-CROGS (Award #DE-SC0023198)

## References

1. runwayml/stable-diffusion-v1-5 · Hugging Face — huggingface.co. <https://huggingface.co/runwayml/stable-diffusion-v1-5>, [Accessed 13-11-2023] 9
2. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint arXiv:2304.08466 (2023) 4
3. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems* **32** (2019) 3, 4
4. Besnier, V., Jain, H., Bursuc, A., Cord, M., Pérez, P.: This dataset does not exist: training models from generated images. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2020) 4
5. Blalock, D., Gonzalez Ortiz, J.J., Frankle, J., Gutttag, J.: What is the state of neural network pruning? *Proceedings of machine learning and systems* **2**, 129–146 (2020) 1
6. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 2
7. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13169–13178 (2020) 2, 3, 7, 11
8. Chen, X., Wang, Y., Yan, R., Liu, Y., Guan, T., He, Y.: Texq: Zero-shot network quantization with texture feature distribution calibration. In: *Thirty-seventh Conference on Neural Information Processing Systems* (2023) 3, 10, 12
9. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018) 3
10. Choi, K., Hong, D., Park, N., Kim, Y., Lee, J.: Qimera: Data-free quantization with synthetic boundary supporting samples. *Advances in Neural Information Processing Systems* **34**, 14835–14847 (2021) 4, 11, 12
11. Choi, K., Lee, H.Y., Hong, D., Yu, J., Park, N., Kim, Y., Lee, J.: It’s all in the teacher: Zero-shot quantization brought closer to the teacher. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8311–8321 (2022) 12
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009) 5, 8
13. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* **34**, 8780–8794 (2021) 4



14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) [2](#), [4](#), [11](#)
15. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. arXiv preprint arXiv:1902.08153 (2019) [3](#), [4](#), [12](#), [13](#)
16. Fang, J., Shafiee, A., Abdel-Aziz, H., Thorsley, D., Georgiadis, G., Hassoun, J.H.: Post-training piecewise linear quantization for deep neural networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 69–86. Springer (2020) [3](#)
17. Finkelstein, A., Almog, U., Grobman, M.: Fighting quantization bias with bias. arXiv preprint arXiv:1906.03193 (2019) [3](#)
18. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630 (2021) [1](#)
19. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT Press (2016) [11](#)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020) [2](#)
21. Haroush, M., Hubara, I., Hoffer, E., Soudry, D.: The knowledge within: Methods for data-free model compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8494–8502 (2020) [2](#), [3](#), [11](#)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) [2](#)
23. He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., Qi, X.: Is synthetic data from generative models ready for image recognition? arXiv preprint arXiv:2210.07574 (2022) [4](#)
24. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) [1](#)
25. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) [5](#)
26. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) [11](#)
27. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research **18**(1), 6869–6898 (2017) [3](#)
28. Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., Soudry, D.: Improving post training neural quantization: Layer-wise calibration and integer programming. arXiv preprint arXiv:2006.10518 (2020) [3](#)
29. Ioffe, S.: Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. Advances in neural information processing systems **30** (2017) [7](#)
30. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015) [7](#)
31. Jahanian, A., Puig, X., Tian, Y., Isola, P.: Generative models as a data source for multiview representation learning. arXiv preprint arXiv:2106.05258 (2021) [4](#)



32. Jeon, Y., Lee, C., Kim, H.y.: Genie: Show me the data for quantization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12064–12073 (2023) [3](#), [5](#), [9](#), [10](#), [11](#)
33. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4350–4359 (2019) [2](#), [3](#)
34. Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., Gu, S.: Brecq: Pushing the limit of post-training quantization by block reconstruction. arXiv preprint arXiv:2102.05426 (2021) [3](#), [5](#), [9](#), [11](#)
35. Li, Y., Zhu, F., Gong, R., Shen, M., Dong, X., Yu, F., Lu, S., Gu, S.: Mixmix: All you need for data-free compression are feature and data mixing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4410–4419 (2021) [2](#), [3](#), [9](#), [10](#), [11](#), [13](#)
36. Li, Z., Ma, L., Chen, M., Xiao, J., Gu, Q.: Patch similarity aware data-free quantization for vision transformers. In: European Conference on Computer Vision. pp. 154–170. Springer (2022) [7](#), [8](#), [12](#)
37. Li, Z., Xiao, J., Yang, L., Gu, Q.: Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17227–17236 (2023) [11](#), [12](#)
38. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020) [6](#)
39. Liu, Y., Zhang, W., Wang, J.: Zero-shot adversarial quantization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1512–1521 (2021) [2](#), [3](#), [4](#), [12](#)
40. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021) [11](#)
41. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Advances in Neural Information Processing Systems **34**, 28092–28103 (2021) [3](#)
42. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022) [4](#)
43. Mehmood, A., Natgunanathan, I., Xiang, Y., Hua, G., Guo, S.: Protection of big data privacy. IEEE access **4**, 1821–1834 (2016) [2](#)
44. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440 (2016) [1](#)
45. Nagel, M., Amjad, R.A., Van Baalen, M., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization. In: International Conference on Machine Learning. pp. 7197–7206. PMLR (2020) [3](#), [5](#)
46. Nahshan, Y., Chmiel, B., Baskin, C., Zheltonozhskii, E., Banner, R., Bronstein, A.M., Mendelson, A.: Loss aware post-training quantization. Machine Learning **110**(11-12), 3245–3262 (2021) [3](#)
47. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021) [4](#)

48. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. *Signal processing* **99**, 215–249 (2014) [6](#)
49. Qian, B., Wang, Y., Hong, R., Wang, M.: Adaptive data-free quantization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7960–7968 (2023) [4](#), [12](#)
50. Qin, H., Ding, Y., Zhang, X., Wang, J., Liu, X., Lu, J.: Diverse sample generation: Pushing the limit of generative data-free quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) [9](#), [12](#)
51. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *International conference on machine learning*. pp. 8748–8763. PMLR (2021) [4](#), [5](#)
52. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10684–10695 (2022) [4](#), [5](#), [6](#)
53. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: *ACM SIGGRAPH 2022 Conference Proceedings*. pp. 1–10 (2022) [4](#)
54. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4510–4520 (2018) [2](#), [11](#)
55. Shin, J., Kang, M., Park, J.: Fill-up: Balancing long-tailed data with generative models. *arXiv preprint arXiv:2306.07200* (2023) [4](#)
56. StabilityAI: Stable diffusion public release (Aug 2023), <https://stability.ai/blog/stable-diffusion-public-release> [2](#), [5](#)
57. Tan, M., Chen, B., Pang, R., Vasudevan, V., Le, Q.V.: Mnasnet: platform-aware neural architecture search for mobile. *corr abs/1807.11626* (2018). *arXiv preprint arXiv:1807.11626* (2018) [11](#)
58. Tian, Y., Fan, L., Isola, P., Chang, H., Krishnan, D.: Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *arXiv preprint arXiv:2306.00984* (2023) [2](#), [4](#)
59. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: *International conference on machine learning*. pp. 10347–10357. PMLR (2021) [11](#)
60. Wei, X., Gong, R., Li, Y., Liu, X., Yu, F.: Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv preprint arXiv:2203.05740* (2022) [3](#), [5](#)
61. Wu, W., Zhao, Y., Chen, H., Gu, Y., Zhao, R., He, Y., Zhou, H., Shou, M.Z., Shen, C.: Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems* **36** (2024) [2](#)
62. Xu, S., Li, H., Zhuang, B., Liu, J., Cao, J., Liang, C., Tan, M.: Generative low-bitwidth data free quantization. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII* 16. pp. 1–17. Springer (2020) [2](#), [3](#), [4](#), [12](#)
63. Yamamoto, K.: Learnable companding quantization for accurate low-bit neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5029–5038 (2021) [3](#)
64. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion.

- In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8715–8724 (2020) 3, 7
65. Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., Xin, J.: Understanding straight-through estimator in training activation quantized neural nets. arXiv preprint arXiv:1903.05662 (2019) 3, 4
  66. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptg4vit: Post-training quantization for vision transformers with twin uniform quantization. In: European Conference on Computer Vision. pp. 191–207. Springer (2022) 11, 12
  67. Zhang, C., Zhang, C., Zheng, S., Qiao, Y., Li, C., Zhang, M., Dam, S.K., Thwal, C.M., Tun, Y.L., Huy, L.L., et al.: A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need? arXiv preprint arXiv:2303.11717 (2023) 2
  68. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 365–382 (2018) 3
  69. Zhang, X., Qin, H., Ding, Y., Gong, R., Yan, Q., Tao, R., Li, Y., Yu, F., Liu, X.: Diversifying sample generation for accurate data-free quantization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15658–15667 (2021) 2, 3
  70. Zhang, Y., Ling, H., Gao, J., Yin, K., Laffleche, J.F., Barriuso, A., Torralba, A., Fidler, S.: Datasetgan: Efficient labeled data factory with minimal human effort. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10145–10155 (2021) 4
  71. Zhao, R., Hu, Y., Dotzel, J., De Sa, C., Zhang, Z.: Improving neural network quantization without retraining using outlier channel splitting. In: International conference on machine learning. pp. 7543–7552. PMLR (2019) 3
  72. Zhong, Y., Lin, M., Nan, G., Liu, J., Zhang, B., Tian, Y., Ji, R.: Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12339–12348 (2022) 3, 4, 11, 12
  73. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016) 3