Q-Diffusion: Quantizing Diffusion Models

Xiuyu Li[1] Yijiang Liu[2] Long Lian[1] Huanrui Yang[1] Zhen Dong[1]
Daniel Kang[3] Shanghang Zhang[4] Kurt Keutzer[1]
[1]UC Berkeley [2]Nanjing University [3]University of Illinois Urbana-Champaign [4]Peking University

Abstract

Diffusion models have achieved great success in image synthesis through iterative noise estimation using deep neural networks. However, the slow inference, high memory consumption, and computation intensity of the noise estimation model hinder the efficient adoption of diffusion models. Although post-training quantization (PTQ) is considered a go-to compression method for other tasks, it does not work out-of-the-box on diffusion models. We propose a novel PTQ method specifically tailored towards the unique multi-timestep pipeline and model architecture of the diffusion models, which compresses the noise estimation network to accelerate the generation process. We identify the key difficulty of diffusion model quantization as the changing output distributions of noise estimation networks over multiple time steps and the bimodal activation distribution of the shortcut layers within the noise estimation network. We tackle these challenges with timestep-aware calibration and split shortcut quantization in this work. Experimental results show that our proposed method is able to quantize full-precision unconditional diffusion models into 4-bit while maintaining comparable performance (small FID change of at most 2.34 compared to >100 for traditional PTQ) in a training-free manner. Our approach can also be applied to text-guided image generation, where we can run stable diffusion in 4-bit weights with high generation quality for the first time.

1. Introduction

Diffusion models have shown great success in generating images with both high diversity and high fidelity [43, 13, 44, 42, 6, 33, 36, 34]. Recent work [16, 15] has demonstrated superior performance than state-of-the-art GAN models, which suffer from unstable training. As a class of flexible generative models, diffusion models demonstrate their power in various applications such as image super-resolution [37, 18], inpainting [44], shape generation [3], graph generation [31], image-to-image translation [40], and molecular conformation generation [47].

However, the generation process for diffusion models can be slow due to the need for an iterative noise estimation of 50 to 1,000 time steps [13, 42] using complex neural networks. While previous state-of-the-art approaches (e.g., GANs) are able to generate multiple images in under 1 second, it normally takes several seconds for a diffusion model to sample a single image. Consequently, speeding up the image generation process becomes an important step toward broadening the applications of diffusion models. Previous work has been solving this problem by finding shorter, more effective sampling trajectories [42, 30, 39, 22, 1, 24], which reduces the number of steps in the denoising process. However, they have largely ignored another important factor: the noise estimation model used in each iteration itself is compute- and memory-intensive. This is an orthogonal factor to the repetitive sampling, which not only slows down the inference speed of diffusion models, but also poses crucial challenges in terms of high memory footprints.

This work explores the quantization [50, 8, 49, 7, 29] of the noise estimation model used in the diffusion model to accelerate the denoising of all time steps. Specifically, we propose exploring post-training quantization (PTQ) on the diffusion model. PTQ has already been well studied in other learning domains like classification and object detection [4, 2, 20, 11, 23], and has been considered a go-to compression method given its minimal requirement for training data and the straightforward deployment on real hardware devices. However, the iterative computation process of the diffusion model and the model architecture of the noise estimation network brings unique challenges to the PTQ of diffusion models. PTQ4DM [41] presents an inaugural application of PTQ to compress diffusion models down to 8-bit, but it primarily focuses on smaller datasets and lower resolutions.

Our work, evolving concurrently with [41], offers a comprehensive analysis of the novel challenges of performing PTQ on diffusion models. Specifically, as visualized in Figure 1(a), we discover that the output distribution of the noise estimation network at each time step can be largely different, and naively applying previous PTQ calibration methods with an arbitrary time step leads to poor performance. Furthermore, as illustrated in Figure 1(b), the iterative inference of the noise estimation network leads to an accumulation of quantization error, which poses higher demands on design-

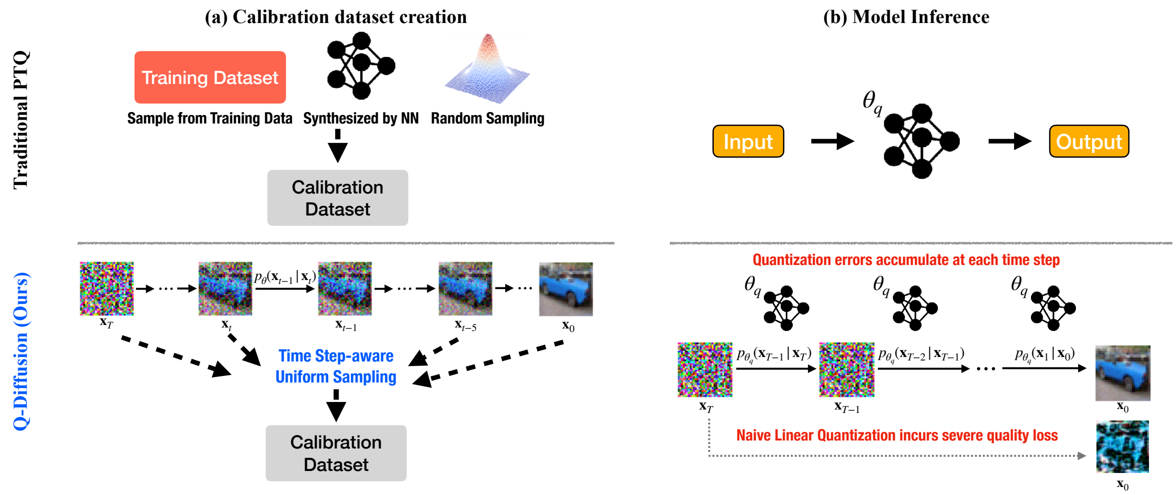

Figure 1: Conventional PTQ scenarios and Q-Diffusion differ in (a) calibration dataset creation and (b) model inference workflow. Traditional PTQ approaches sample data randomly [11], synthesize with statistics in model layers [4], or draw from the training set to create calibration dataset [28, 20], which either contains inconsistency with real inputs during the inference time or are not data-free. In contrast, Q-Diffusion constructed calibration datasets with inputs that are an accurate reflection of data seen during the production in a data-free manner. Traditional PTQ inference only needs to go through the quantized model θ_q one time, while Q-Diffusion needs to address the accumulated quantization errors in the multi-time step inference.

ing novel quantization schemes and calibration objectives for the noise estimation network.

To address these challenges, we propose **Q-Diffusion**, a PTQ solution to compress the cumbersome noise estimation network in diffusion models in a data-free manner, while maintaining comparable performance to the full precision counterparts. We propose a time step-aware calibration data sampling mechanism from the pretrained diffusion model, which represents the activation distribution of all time steps. We further tailor the design of the calibration objective and the weight and activation quantizer to the commonly used noise estimation model architecture to reduce quantization error. We perform thorough ablation studies to verify our design choices, and demonstrate good generation results with diffusion models quantized to only 4 bits.

In summary, our contributions are:

1. We propose **Q-Diffusion**, a data-free PTQ solution for the noise estimation network in diffusion models.
2. We identify the novel challenge of performing PTQ on diffusion models as the activation distribution diversity and the quantization error accumulation across time steps via a thorough analysis.
3. We propose time step-aware calibration data sampling to improve calibration quality, and propose a specialized quantizer for the noise estimation network.
4. Extensive results show Q-Diffusion enables W4A8 PTQ for both pixel-space and latent-space unconditional diffusion models with an FID increment of only 0.39-2.34 over full precision models. It can also produce qualitatively comparable images when plugged into Stable Diffusion [34] for text-guided synthesis.

2. Related work

Diffusion Models. Diffusion models generate images through a Markov chain, as illustrated in Figure 2. A forward diffusion process adds Gaussian noise to data $\mathbf{x}_0 \sim q(\mathbf{x})$ for T times, resulting in noisy samples $\mathbf{x}_1, ..., \mathbf{x}_T$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (1)$$

where $\beta_t \in (0,1)$ is the variance schedule that controls the strength of the Gaussian noise in each step. When $T \to \infty$, $\mathbf{x}_T$ approaches an isotropic Gaussian distribution.

The reverse process removes noise from a sample from the Gaussian noise input $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to gradually generate high-fidelity images. However, since the real reverse conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unavailable, diffusion models sample from a learned conditional distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_{\theta,t}(\mathbf{x}_t), \tilde{\beta}_t\mathbf{I}). \quad (2)$$

With the reparameterization trick in [13], the mean $\tilde{\boldsymbol{\mu}}_{\theta,t}(\mathbf{x}_t)$ and variance $\tilde{\beta}_t$ could be derived as follows:

$$\tilde{\boldsymbol{\mu}}_{\theta,t}(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\theta,t}) \quad (3)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \quad (4)$$

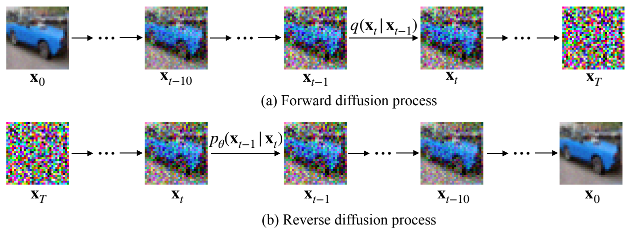

Figure 2: The forward diffusion process **(a)** repeatedly adds Gaussian noise. The reverse diffusion process **(b)** uses a trained network to denoise from a standard Gaussian noise image in order to generate an image.

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. We refer readers to [26] for a more detailed introduction.

In practice, the noise at each time step t are computed from $\mathbf{x}_t$ by a noise estimation model, with the same weights for all time steps. The UNet [35] dominates the design of the noise estimation model in diffusion models [42, 34, 33, 36], with some recent exploration on Transformer [32]. This work designs the PTQ method for the acceleration of the noise estimation model, especially for the common UNet.

Accelerated diffusion process. Related methods include simulating the diffusion process in fewer steps by generalizing it to a non-Markovian process [42], adjusting the variance schedule [30], and the use of high-order solvers to approximate diffusion generation [22, 1, 24, 25]. Others have employed the technique of caching and reusing feature maps [19]. Efforts to distill the diffusion model into fewer time steps have also been undertaken [39, 27], which have achieved notable success but involve an extremely expensive retraining process. Our work focuses on accelerating the noise estimation model inference in each step, with a training-free PTQ process.

Post-training Quantization. Post-training quantization (PTQ) compresses deep neural networks by rounding elements w to a discrete set of values [10], where the quantization and de-quantization can be formulated as:

$$\hat{w} = \mathrm{s} \cdot \mathrm{clip}(\mathrm{round}(\frac{w}{s}), c_{\text{min}}, c_{\text{max}}), \tag{5}$$

where s denotes the quantization scale parameters, c_{min} and c_{max} are the lower and upper bounds for the clipping function $\mathrm{clip}(\cdot)$. These parameters can be calibrated with the weight and activation distribution estimated in the PTQ process. The operator $\mathrm{round}(\cdot)$ represents rounding, which can be either rounding-to-nearest [46, 4] or adaptive rounding [20].

Previous PTQ research in classification and detection tasks focused on the calibration objective and the acquisition of calibration data. For example, EasyQuant [46] determines appropriate c_{min} and c_{max} based on training data, and BRECQ [20] introduces Fisher information into the objective. ZeroQ [4] employs a distillation technique to generate proxy input images for PTQ, and SQuant [11] uses random samples with objectives based on sensitivity determined through the Hessian spectrum. For diffusion model quantization, a training dataset is not needed as the calibration data can be constructed by sampling the full-precision model with random inputs. However, the multi-time step inference of the noise estimation model brings new challenges in modeling the activation distribution. In parallel to our work, PTQ4DM [41] introduces the method of Normally Distributed Time-step Calibration, generating calibration data across all time steps with a specific distribution. Nevertheless, their explorations remain confined to lower resolutions, 8-bit precision, floating-point attention activation-to-activation matmuls, and with limited ablation study on other calibration schemes. This results in worse applicability of their method to lower precisions (see Appendix). Our work delves into the implications of calibration dataset creation in a holistic manner, establishing an efficient calibration objective for diffusion models. We fully quantize act-to-act matmuls, validated by experiments involving both pixel-space and latent-space diffusion models on large-scale datasets up to resolutions of 512×512.

3. Method

We present our method for post-training quantization of diffusion models in this section. Different from conventionally studied deep learning models and tasks such as CNNs and VITs for classification and detection, diffusion models are trained and evaluated in a distinctive multi-step manner with a unique UNet architecture. This presents notable challenges to the PTQ process. We analyze the challenges brought by the multi-step inference process and the UNet architecture in Section 3.1 and 3.2 respectively and describe the full Q-Diffusion PTQ pipeline in Section 3.3.

3.1. Challenges under the Multi-step Denoising

We identify two major challenges in quantizing models that employ multi-step inference process. Namely, we investigate the accumulation of quantization error across time steps and the difficulty of sampling a small calibration dataset to reduce the quantization error at each time step.

Challenge 1: Quantization errors accumulate across time steps. Performing quantization on a neural network model introduces noise on the weight and activation of the well-trained model, leading to quantization errors in each layer's output. Previous research has identified that quantization errors are likely to accumulate across layers [5], making deeper neural networks harder to quantize. In the case of diffusion models, at any time step t, the input of the denoising model (denoted as $\mathbf{x}_t$) is derived by $\mathbf{x}_{t+1}$, the output of the model at the previous time step $t+1$ (as depicted by Equation 2). This process effectively multiplies the number of layers involved in the computation by the number of de-

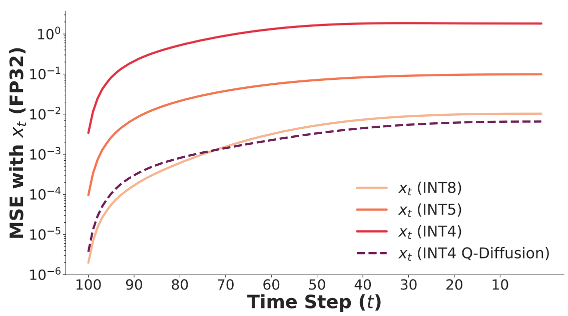

Figure 3: MSE between FP32 outputs and weight-quantized outputs of different precisions with Linear Quantization and our approach across time steps. Here the data is obtained by passing a batch with 64 samples through a model trained on CIFAR-10 [17] with DDIM sampling steps 100.

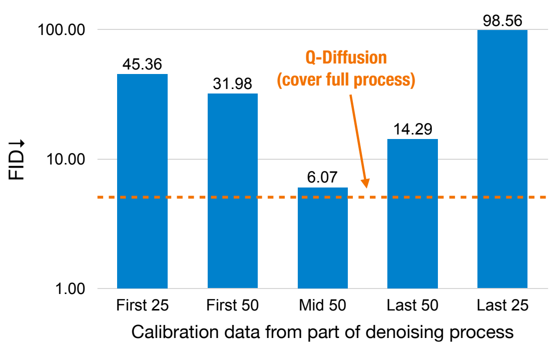

Figure 4: Effects of time steps in calibration dataset creation on 4-bit weights quantization results with DDIM on CIFAR-10. First n, Mid n, Last n denotes that 5120 samples are selected uniformly from the first, middle, last n time steps.

noising steps for the input $\mathbf{x}_t$ at time step t, leading to an accumulation of quantization errors towards later steps in the denoising process.

We run the denoising process of DDIM [42] on CIFAR-10 [17] with a sampling batch size of 64, and compare the MSE differences between the full-precision model and the model quantized to INT8, INT5, and INT4 at each time step. As shown in Figure 3, there is a dramatic increase in the quantization errors when the model is quantized to 4-bit, and the errors accumulate quickly through iterative denoising. This brings difficulty in preserving the performance after quantizing the model down to low precision, which requires the reduction of quantization errors at all time steps as much as possible.

Challenge 2: Activation distributions vary across time steps. To reduce the quantization errors at each time step, previous PTQ research [28, 20] calibrates the clipping range and scaling factors of the quantized model with a small set of calibration data. The calibration data should be sampled to resemble the true input distribution so that the activation distribution of the model can be estimated correctly for proper calibration. Given that the Diffusion model uses the same noise estimation network to take inputs from all time steps, determining the data sampling policy across different time steps becomes an outstanding challenge. Here we start by analyzing the output activation distribution of the UNet model across different time steps. We conduct the same CIFAR-10 experiment using DDIM with 100 denoising steps, and draw the activations ranges of 1000 random samples among all time steps. As Figure 5 shows, the activation distributions gradually change, with neighboring time steps being similar and distant ones being distinctive. This is also echoed by the visualized $\mathbf{x}_t$ in Figure 2.

The fact that the output activations distribution varies across time steps further brings challenges to quantization. Calibrating the noise estimation model using only a few time steps that do not reflect the full range of activations seen among all time steps by the noise estimation model during the denoising process can cause overfitting to the activation distribution described by those specific time steps, while not generalizing to other time steps, which hurts the overall performance. For instance, here we try to calibrate the quantized DDIM on the CIFAR-10 dataset with data sampled from different parts of the denoising process. As shown in Figure 4, if we simply take 5120 samples from time steps that fall into a certain stage of the denoising process, significant performance drops will be induced under 4-bit weights quantization. Note that the case with samples taken from the middle 50 time steps caused smaller drops compared to cases with samples taken from either the first or the last n time steps, and with n increases, the drops are also alleviated. These results illustrate the gradual "denoising" process as depicted in Figure 5: the activations distribution changes gradually throughout time steps, with the middle part capturing the full range to some degree, while parts of the distant endpoints differing the most. To recover the performance of the quantized diffusion models, we need to select calibration data in a way that comprehensively takes into account the distributions of the output of different time steps.

3.2. Challenges on Noise Estimation Model Quantization

Most diffusion models (Imagen [36], Stable Diffusion [34], VDMs [14]) adopt UNets as denoising backbones that downsample and upsample latent features. Although recent studies show that transformer architectures are also capable of serving as the noise estimation backbone [32], convolutional UNets are still the de facto choice of architecture today. UNets utilize shortcut layers to merge concatenated deep and shallow features and transmit them to subsequent layers. Through our analysis presented in Figure 6, we observe that input activations in shortcut layers exhibit abnormal value ranges in comparison to other layers.

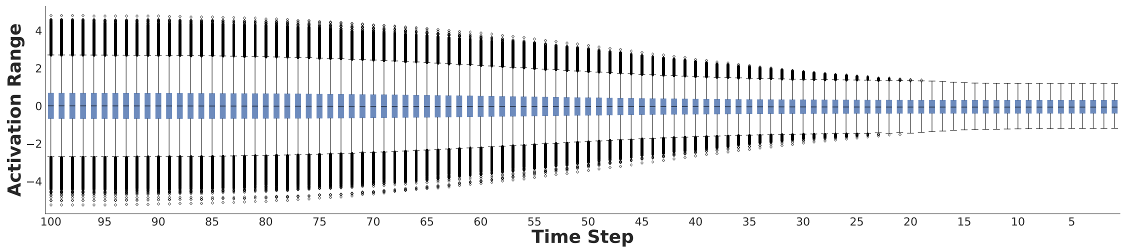

Figure 5: Activation ranges of x_t across all 100 time steps of FP32 DDIM model on CIFAR-10.

Notably, the input activations in DDIM's shortcut layers can be up to 200 times larger than other neighboring layers.

To analyze the reason for this, we visualize the weight and activation tensor of a DDIM shortcut layer. As demonstrated in the dashed box in Figure 6, the ranges of activations from the deep feature channels (X_1) and shallow feature channels (X_2) being concatenated together vary significantly, which also resulted in a bimodal weight distribution in the corresponding channels (see also Figure 7). Naively quantizing the entire weight and activation distribution with the same quantizer will inevitably lead to large quantization errors.

3.3. Post-Training Quantization of Diffusion Model

We propose two techniques: *time step-aware calibration data sampling* and *shortcut-splitting quantization* to tackle the challenges identified in the previous sections respectively.

3.3.1 Time step-aware calibration

Since the output distributions of consecutive time steps are often very similar, we propose to randomly sample intermediate inputs uniformly in a fixed interval across all time steps to generate a small calibration set. This effectively balances the size of the calibration set and its representation ability of the distribution across all time steps. Empirically, we have found that the sampled calibration data can recover most of the INT4 quantized models' performance after the calibration, making it an effective sampling scheme for calibration data collection for quantization error correction.

To calibrate the quantized model, we divide the model into several reconstruction blocks [20], and iteratively reconstruct outputs and tune the clipping range and scaling factors of weight quantizers in each block with adaptive rounding [28] to minimize the mean squared errors between the quantized and full precision outputs. We define a core component that contains residual connections in the diffusion model UNet as a block, such as a Residual Bottleneck Block or a Transformer Block. Other parts of the model that do not satisfy this condition are calibrated in a per-layer manner. This technique has been shown to improve the performance compared to fully layer-by-layer calibration since it address the inter-layer dependencies and generalization better [20]. For activation quantization, since activations are constantly changing during inference, doing adaptive rounding is infeasible. Therefore, we only adjust the step sizes of activation quantizers according to to [9]. The overall calibration workflow is described in Alg. 1.

Algorithm 1 Q-Diffusion Calibration

Require: Pretrained full precision diffusion model and the quantized diffusion model $[W_\theta, \hat{W}_\theta]$
Require: Empty calibration dataset $\mathcal{D}$
Require: Number of denoising sampling steps T
Require: Calibration sampling interval c, amount of calibration data per sampling step n
  for $t = 1, \dots, T$ time step **do**
    if t % c = 0 **then**
      Sample n intermediate inputs $\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(n)}$ randomly at t from W_θ and add them to $\mathcal{D}$
    end if
  end for
  for all $i = 1, \dots, N$ blocks **do**
    Update the weight quantizers of the i-th block in $\hat{W}_\theta$ with $\mathcal{D}$ and W_θ
  end for
  if do activation quantization **then**
    for all $i = 1, \dots, N$ blocks **do**
      Update the activation quantizers step sizes of the i-th block with $\hat{W}_\theta$, W_θ, $\mathcal{D}$.
    end for
  end if

3.3.2 Shortcut-splitting quantization

To address the abnormal activation and weight distributions in shortcut layers, we propose a "split" quantization technique that performs quantization prior to concatenation, requiring negligible additional memory or computational resources. This strategy can be employed for both activation and weight quantization in shortcut layers, and is expressed

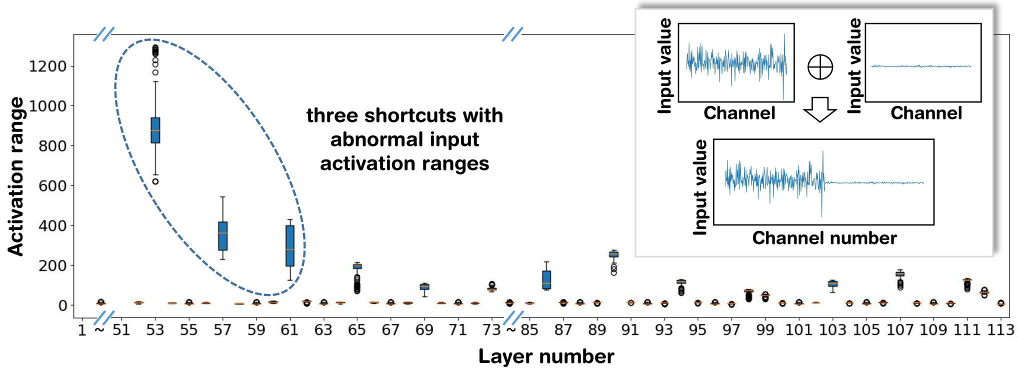

Figure 6: Activation ranges of DDIM's FP32 outputs across layers averaging among all time steps. We point out three shortcuts with the largest input activation ranges compared to other neighboring layers. Figures in the dashed box illustrate concatenation along channels. $\oplus$ denotes the concatenation operation.

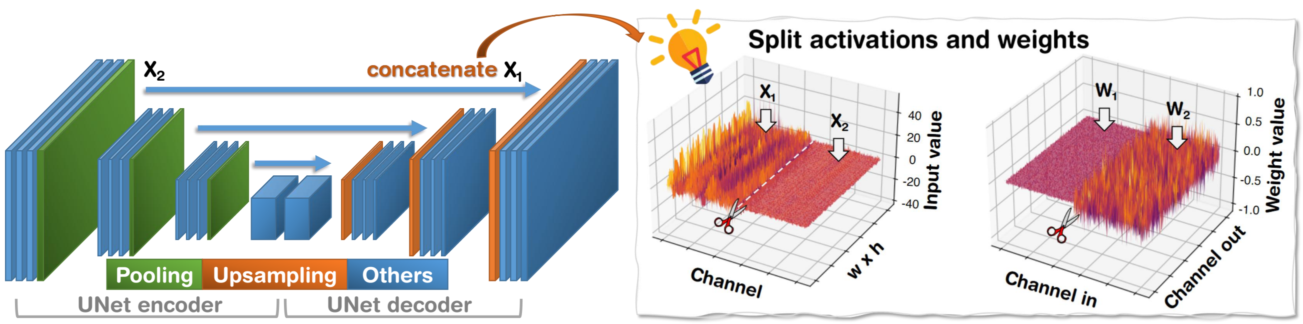

Figure 7: (Left) The typical UNet architecture with shortcut layers that concatenate features from the deep and shallow layers. (Right) The ranges of activations from the deep (X_1) and shallow (X_2) feature channels vary significantly, which also results in a bimodal weight distribution in the corresponding channels.

mathematically as follows:

$$\mathcal{Q}_X(X) = \mathcal{Q}_{X_1}(X_1) \oplus \mathcal{Q}_{X_2}(X_2) \tag{6}$$
$$\mathcal{Q}_W(W) = \mathcal{Q}_{W_1}(W_1) \oplus \mathcal{Q}_{W_2}(W_2) \tag{7}$$

$$\begin{aligned}\mathcal{Q}_X(X)\mathcal{Q}_W(W) =& \mathcal{Q}_{X_1}(X_1)\mathcal{Q}_{W_1}(W_1) \\ &+ \mathcal{Q}_{X_2}(X_2)\mathcal{Q}_{W_2}(W_2)\end{aligned} \tag{8}$$

where $X \in \mathbb{R}^{w \times h \times c_{in}}$ and $W \in \mathbb{R}^{c_{in} \times c_{out}}$ are the input activation and layer weight, which can be naturally split into $X_1 \in \mathbb{R}^{w \times h \times c_1}$, $X_2 \in \mathbb{R}^{w \times h \times c_2}$, $W_1 \in \mathbb{R}^{c_1 \times c_{out}}$, and $W_2 \in \mathbb{R}^{c_2 \times c_{out}}$, respectively. c_1 and c_2 are determined by the concatenation operation. $\mathcal{Q}(\cdot)$ denotes the quantization operator and $\oplus$ denotes the concatenation operator.

4. Experiments

4.1. Experiments Setup

In this section, we evaluate the proposed Q-Diffusion framework on pixel-space diffusion model DDPM [13] and latent-space diffusion model Latent Diffusion [34] for unconditional image generation. We also visualize the images generated by Q-Diffusion on Stable Diffusion. To the best of our knowledge, there is currently no published work done on diffusion model quantization. Therefore, we report the basic channel-wise round-to-nearest Linear Quantization (i.e., Equation 5) as a baseline. We also re-implement the state-of-the-art data-free PTQ method SQuant [11] and include the results for comparison. Furthermore, we apply our approach to text-guided image synthesis with Stable Diffusion [34]. Experiments show that our approach can achieve competitive generation quality to the full-precision scenario on all tasks, even under INT4 quantization for weights.

4.2. Unconditional Generation

We conducted evaluations using the 32 $\times$ 32 CIFAR-10 [17], 256 $\times$ 256 LSUN Bedrooms, and 256 $\times$ 256 LSUN Church-Outdoor [48]. We use the pretrained DDIM sampler [42] with 100 denoising time steps for CIFAR-10 experiments and Latent Diffusion (LDM) [34] for the higher resolution LSUN experiments. We evaluated the performance in terms of Frechet Inception Distance (FID) [12] and additionally evaluated the Inception Score (IS) [38] for CIFAR-10 results, since IS is not an accurate reference for datasets that

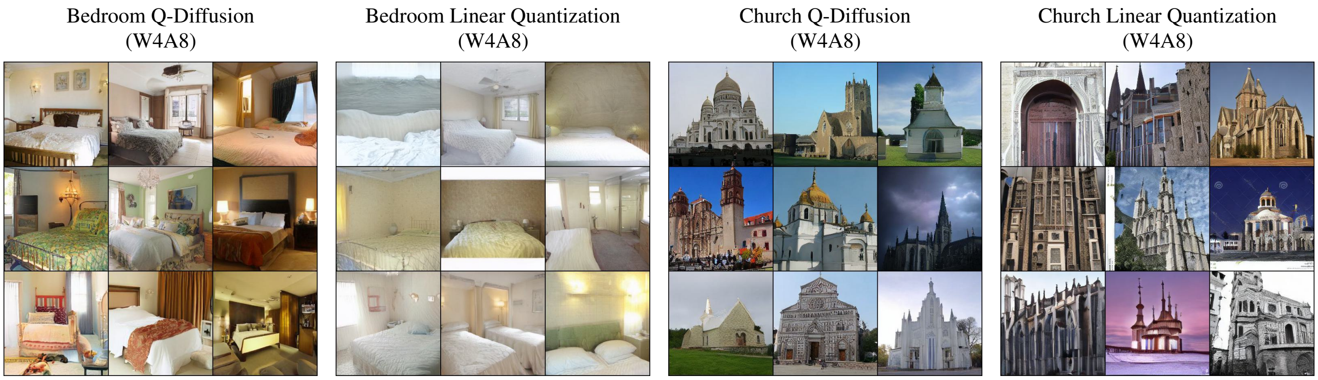

Figure 8: 256 $\times$ 256 unconditional image generation results using Q-Diffusion and Linear Quantization under W4A8 precision.

Table 1: Quantization results for unconditional image generation with DDIM on CIFAR-10 (32 $\times$ 32).

Method	Bits (W/A)	Size (Mb)	GBops	FID↓	IS↑
Full Precision	32/32	143.2	6597	4.22	9.12
Linear Quant	8/32	35.8	2294	4.71	8.93
SQuant	8/32	35.8	2294	4.61	8.99
Q-Diffusion	8/32	35.8	2294	**4.27**	**9.15**
Linear Quant	4/32	17.9	1147	141.47	4.20
SQuant	4/32	17.9	1147	160.40	2.91
Q-Diffusion	4/32	17.9	1147	**5.09**	**8.78**
Linear Quant	8/8	35.8	798	118.26	5.23
SQuant	8/8	35.8	798	464.69	1.17
Q-Diffusion	8/8	35.8	798	**3.75**	**9.48**
Linear Quant	4/8	17.9	399	188.11	2.45
SQuant	4/8	17.9	399	456.21	1.16
Q-Diffusion	4/8	17.9	399	**4.93**	**9.12**

differ significantly from ImageNet's domain and categories. The results are reported in Table 1- 3 and Figure 8, where Bops is calculated for one denoising step without considering the decoder compute cost for latent diffusion.

The experiments show that Q-Diffusion significantly preserves the image generation quality and outperforms Linear Quantization by a large margin for all resolutions and types of diffusion models tested when the number of bits is low. Although 8-bit weight quantization has almost no performance loss compared to FP32 for both Linear Quantization and our approach, the generation quality with Linear Quantization drops drastically under 4-bit weight quantization. In contrast, Q-Diffusion still preserves most of the perceptual quality with at most 2.34 increase in FID and imperceptible distortions in produced samples.

4.3. Text-guided Image Generation

We evaluate Q-Diffusion on Stable Diffusion pretrained on subsets of 512 $\times$ 512 LAION-5B for text-guided image generation. Following [34], we sample text prompts from the MS-COCO [21] dataset to generate a calibration dataset with texts condition using Algorithm 1. In this work, we fix

Table 2: Quantization results for unconditional image generation with LDM-4 on LSUN-Bedrooms (256 $\times$ 256). The downsampling factor for the latent space is 4.

Method	Bits (W/A)	Size (Mb)	TBops	FID↓
Full Precision	32/32	1096.2	107.17	2.98
Linear Quant	8/32	274.1	37.28	3.02
SQuant	8/32	274.1	37.28	**2.94**
Q-Diffusion	8/32	274.1	37.28	2.97
Linear Quant	4/32	137.0	18.64	82.69
SQuant	4/32	137.0	18.64	149.97
Q-Diffusion	4/32	137.0	18.64	**4.86**
Linear Quant	8/8	274.1	12.97	6.69
SQuant	8/8	274.1	12.97	4.92
Q-Diffusion	8/8	274.1	12.97	**4.40**
Linear Quant	4/8	137.0	6.48	24.86
SQuant	4/8	137.0	6.48	95.92
Q-Diffusion	4/8	137.0	6.48	**5.32**

Table 3: Quantization results for unconditional image generation with LDM-8 on LSUN-Churches (256 $\times$ 256). The downsampling factor for the latent space is 8.

Method	Bits (W/A)	Size (Mb)	TBops	FID↓
Full Precision	32/32	1179.9	22.17	4.06
Linear Quant	8/32	295.0	10.73	**3.84**
SQuant	8/32	295.0	10.73	4.01
Q-Diffusion	8/32	295.0	10.73	4.03
Linear Quant	4/32	147.5	5.36	32.54
SQuant	4/32	147.5	5.36	33.77
Q-Diffusion	4/32	147.5	5.36	**4.45**
Linear Quant	8/8	295.0	2.68	14.62
SQuant	8/8	295.0	2.68	54.15
Q-Diffusion	8/8	295.0	2.68	**3.65**
Linear Quant	4/8	147.5	1.34	14.92
SQuant	4/8	147.5	1.34	24.50
Q-Diffusion	4/8	147.5	1.34	**4.12**

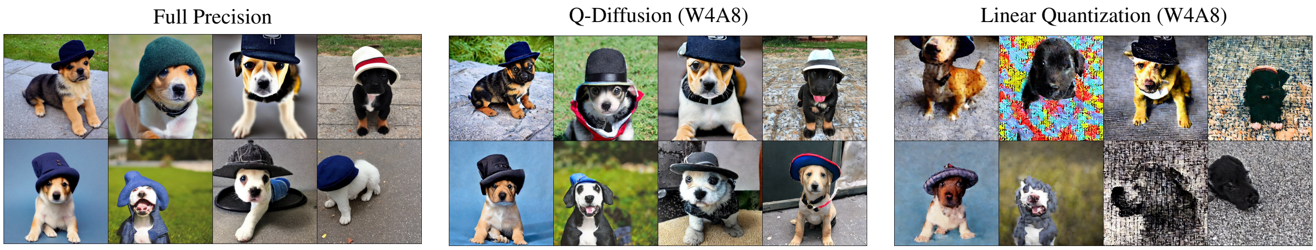

Figure 9: Stable Diffusion 512 × 512 text-guided image synthesis results using Q-Diffusion and Linear Quantization under W4A8 precision with prompt *A puppy wearing a hat*.

the guidance strength to the default 7.5 in Stable Diffusion as the trade-off between sample quality and diversity. Qualitative results are shown in Figure 9. Compared to Linear Quantization, our Q-Diffusion provides higher-quality images with more realistic details and better demonstration of the semantic information. Similar performance gain is also observed in other random samples showcased in Appendix, and quantitatively reported in Appendix. The output of the W4A8 Q-Diffusion model largely resembles the output of the full precision model. Interestingly, we find some diversity in the lower-level semantics between the Q-Diffusion model and the FP models, like the heading of the horse or the shape of the hat. We leave it to future work to understand how quantization contributes to the diversity.

4.4. Ablation Study

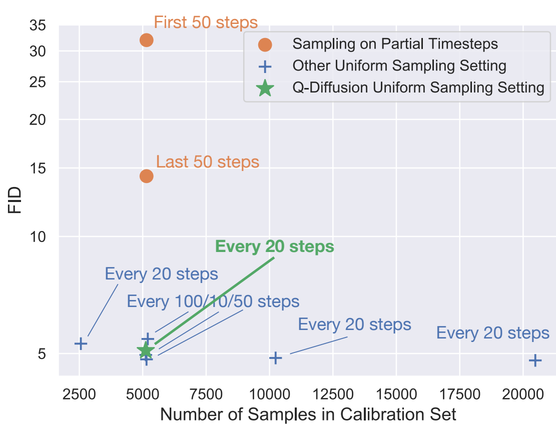

Figure 10: Uniform sampling strategies which cover all time steps are better than strategies that cover only a part of the time steps, as in Fig. 4. Furthermore, adjusting the sampling techniques within uniform sampling, such as tuning the sampling interval and the number of samples, has a marginal effect on the performance of the quantized model.

Effects of Sampling Strategies To analyze the effects of different sampling strategies for calibration in detail, we implemented multiple variants of our method using different sampling strategies. We then evaluated the quality of the models quantized by each variant. We experimented with varying numbers of time steps used for sampling and samples used for calibration. In addition to calibration sets from uniform timestep intervals, we also employed sampling at the first 50 and last 50 steps. As in Figure 10, uniform sampling that spans all time steps results in superior performance compared to sampling from only partial time steps. Furthermore, adjusting the sampling hyperparams, including using more calibration samples, does not significantly improve the performance. Therefore, we simply choose to sample uniformly every 20 steps for a total of 5,120 samples for calibration, resulting in a high-quality quantized model with low computational costs during quantization.

We also conduct ablation experiments to explore the effectiveness of several non-uniform calibration data sampling schemes, such as using Unsupervised Selecting Labeling (USL) [45] to select both representative and diverse calibration samples. We present the results in Appendix.

Effects of Split Previous linear quantization approaches suffer from severe performance degradation as shown in Figure 11, where 4-bit weight quantization achieves a high FID of 141.47 in DDIM CIFAR-10 generation. Employing additional 8-bit activation quantization further degrades the performance (FID: 188.11). By splitting shortcuts in quantization, we significantly improve the generation performance, achieving an FID of 4.93 on W4A8 quantization.

5. Conclusion

This work studies the use of quantization to accelerate and reduce the memory usage of diffusion models. We propose Q-Diffusion, a novel post-training quantization scheme that conducts calibration with multiple time steps in the denoising process and achieves significant improvements in the performance of the quantized model. Q-Diffusion models under 4-bit quantization achieve comparable results to the full precision models.

Acknowledgement

We thank Berkeley Deep Drive, Intel Corporation, Panasonic, and NVIDIA for supporting this research. We also

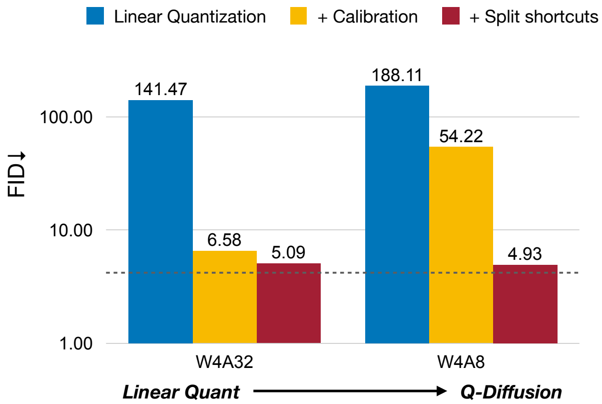

Figure 11: Splitting the shortcut convolution is crucial for both weight and activation quantization. Comparisons on CIFAR-10 show that Q-Diffusion could achieve comparable image generation quality to the model with full precision (dashed line) with shortcut splitting.

Figure 12: Examples of text-to-image generation with a quantized Stable Diffusion model. Naive linear quantization degrades the appearance of teeth, which gets fixed by shortcut splitting. Q-Diffusion further improves the semantic consistency of eyes through calibration.

would like to thank Sehoon Kim, Muyang Li, and Minkai Xu for their valuable feedback.

References

[1] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *ArXiv*, abs/2201.06503, 2022. 1, 3

[2] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2978–2985, 2020. 1

[3] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge J. Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, 2020. 1

[4] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13166–13175, 2020. 1, 2, 3

[5] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022. 3

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[7] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. *Advances in neural information processing systems*, 33:18518–18529, 2020. 1

[8] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019. 1

[9] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020. 5

[10] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022. 3

[11] Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. Squant: On-the-fly data-free quantization via diagonal hessian approximation. *ArXiv*, abs/2202.07471, 2022. 1, 2, 3, 6

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 6

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 4

[15] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 1

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[17] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 4, 6

[18] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhi hai Xu, Qi Li, and Yue ting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2021. 1

[19] Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. Efficient spatially sparse inference for conditional gans and diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[20] 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 *International Conference on Learning Representations*, 2021. 1, 2, 3, 4, 5

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 7

[22] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 1, 3

[23] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. *arXiv preprint arXiv:2211.16056*, 2022. 1

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *ArXiv*, abs/2206.00927, 2022. 1, 3

[25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *ArXiv*, abs/2211.01095, 2022. 3

[26] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 3

[27] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2022. 3

[28] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. *ArXiv*, abs/2004.10568, 2020. 2, 4, 5

[29] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. 1

[30] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1, 3

[31] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, 2020. 1

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3, 4

[33] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 3

[34] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 1, 2, 3, 4, 6, 7

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 3, 4

[37] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021. 1

[38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6

[39] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *ArXiv*, abs/2202.00512, 2022. 1, 3

[40] Hiroshi Sasaki, Chris G. Willcocks, and T. Breckon. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models. *ArXiv*, abs/2104.05358, 2021. 1

[41] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. *CVPR*, 2023. 1, 3

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 3, 4, 6

[43] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[44] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[45] Xudong Wang, Long Lian, and Stella X Yu. Unsupervised selective labeling for more effective semi-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 427–445. Springer, 2022. 8

[46] Di Wu, Qingming Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. Easyquant: Post-training quantization via scale optimization. *ArXiv*, abs/2006.16669, 2020. 3

[47] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: a geometric diffusion model for

molecular conformation generation. *ArXiv*, abs/2203.02923, 2022. 1

[48] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[49] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *International conference on machine learning*, pages 7543–7552. PMLR, 2019. 1

[50] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017. 1