Q-Diffusion: Quantizing Diffusion Models

Xiuyu Li\textsuperscript{1}  Yijiang Liu\textsuperscript{2}  Long Lian\textsuperscript{1}  Huanrui Yang\textsuperscript{1}  Zhen Dong\textsuperscript{1}  Daniel Kang\textsuperscript{3}  Shanghang Zhang\textsuperscript{4}  Kurt Keutzer\textsuperscript{1}
\textsuperscript{1}UC Berkeley  \textsuperscript{2}Nanjing University  \textsuperscript{3}University of Illinois Urbana-Champaign \textsuperscript{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 \cite{DiffusionModels2021,Mixmatch2021,DDIM2020,DDPM2020,DDPMPlus2020,DDPMNeRF2020,DDPMNeRF2021}. Recent work \cite{DDPMWeb2021,DDPMWeb2022} 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 \cite{DiffusionModels2021,DDPMNeRF2020}, inpainting \cite{DDPMInpainting2021}, shape generation \cite{DiffusionModels2021}, graph generation \cite{DiffusionModels2021}, image-to-image translation \cite{DiffusionModels2021}, and molecular conformation generation \cite{DiffusionModels2021}.

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 \cite{DDPMWeb2021,DDPMWeb2022} 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 \cite{DDPMNeRF2020,DDPMNeRF2021,DDPMNeRF2022,DDPMNeRF2023,DDPMNeRF2024,DDPMNeRF2025}, 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 \cite{DiffusionModels2021,DDPMWeb2021,DDPMWeb2022,DDPMWeb2023,DDPMWeb2024} 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 \cite{DiffusionModels2021,DDPMWeb2021,DDPMWeb2022,DDPMWeb2023,DDPMWeb2024,DDPMWeb2025}, 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 \cite{DiffusionModels2021} 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 \cite{DiffusionModels2021}, 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-
We further tailor the design of the calibration objective and with diffusion models quantized to only 4 bits. We propose Q-Diffusion, a data-free PTQ solution to compress the cumbersome noise estimation network in diffusion models in a data-free manner. Traditional PTQ inference only needs to go through the quantized \(\theta\) model one time, while Q-Diffusion needs to address the accumulated quantization errors in the multi-time step inference.

### 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 \(x_0 \sim q(x)\) for \(T\) times, resulting in noisy samples \(x_1, \ldots, x_T\):  

\[
q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \tag{1}
\]

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

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

\[
p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \tilde{\mu}_{\theta,t}(x_t), \tilde{\beta}_t I). \tag{2}
\]

With the reparameterization trick in \cite{13}, the mean \(\tilde{\mu}_{\theta,t}(x_t)\) and variance \(\tilde{\beta}_t\) could be derived as follows:

\[
\tilde{\mu}_{\theta,t}(x_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_{\theta,t}) \tag{3}
\]

\[
\tilde{\beta}_t = \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \cdot \beta_t \tag{4}
\]

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 \cite{34} for text-guided synthesis.
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:

\[
\hat{w} = s \cdot \text{clip}(\text{round}\left(\frac{w}{s}\right), c_{\text{min}}, c_{\text{max}}),
\]

where \( s \) denotes the quantization scale parameters, \( c_{\text{min}} \) and \( c_{\text{max}} \) are the lower and upper bounds for the clipping function \( \text{clip}(\cdot) \). These parameters can be calibrated with the weight and activation distribution estimated in the PTQ process. The operator \( \text{round}(\cdot) \) 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 \( c_{\text{min}} \) and \( c_{\text{max}} \) 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 \( 512 \times 512 \).

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 \( x_t \)) is derived by \( x_{t+1} \), 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-
MSE differences between the full-precision model and the model quantized to INT8, INT5, and INT4 at each time step. The errors accumulate quickly through iterative denoising. To reduce the quantization errors at each time step, 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.
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** 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 $[\hat{W}_\theta, W_\theta]$

**Require:** Empty calibration dataset $D$

**Require:** Number of denoising sampling steps $T$

**Require:** Calibration sampling interval $c$, amount of calibration data per sampling step $n$

```plaintext
for $t = 1, \ldots, T$ time step do
  if $t \% c = 0$ then
    Sample $n$ intermediate inputs $x_t^{(1)}, \ldots, x_t^{(n)}$ randomly at $t$ from $W_\theta$ and add them to $D$
  end if
end for
for all $i = 1, \ldots, N$ blocks do
  Update the weight quantizers of the $i$-th block in $\hat{W}_\theta$ with $D$ and $W_\theta$
end for
if do activation quantization then
  for all $i = 1, \ldots, N$ blocks do
    Update the activation quantizers step sizes of the $i$-th block with $\hat{W}_\theta, W_\theta, D$
  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. ⊕ 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:

$$Q_X(X) = Q_{X_1}(X_1) \oplus Q_{X_2}(X_2)$$  \hspace{1cm} (6)

$$Q_W(W) = Q_{W_1}(W_1) \oplus Q_{W_2}(W_2)$$  \hspace{1cm} (7)

$$Q_X(X)Q_W(W) = Q_{X_1}(X_1)Q_{W_1}(W_1) + Q_{X_2}(X_2)Q_{W_2}(W_2)$$  \hspace{1cm} (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. $Q(\cdot)$ 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
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...
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
would like to thank Sehoon Kim, Muyang Li, and Minkai Xu for their valuable feedback.

References


