

# Sampling-Aware Quantization for Diffusion Models

Qian Zeng<sup>1</sup>, Jie Song<sup>1\*</sup>, Yuanyu Wan<sup>1</sup>, Huiqiong Wang<sup>2</sup>, Mingli Song<sup>1,3,4</sup>

<sup>1</sup>School of Software Technology, Zhejiang University

<sup>2</sup>Ningbo Innovation Center, Zhejiang University

<sup>3</sup>State Key Laboratory of Blockchain and Security, Zhejiang University

<sup>4</sup>Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security

{qianz, sjie, wanyy, huiqiong-wang, brooksong}@zju.edu.cn

## Abstract

Diffusion models have recently emerged as the dominant approach in visual generation tasks. However, the lengthy denoising chains and the computationally intensive noise estimation networks hinder their applicability in low-latency and resource-limited environments. Previous research has endeavored to address these limitations in a decoupled manner, utilizing either advanced samplers or efficient model quantization techniques. In this study, we uncover that quantization-induced noise disrupts directional estimation at each sampling step, further distorting the precise directional estimations of higher-order samplers when solving the sampling equations through discretized numerical methods, thereby altering the optimal sampling trajectory. To attain dual acceleration with high fidelity, we propose a sampling-aware quantization strategy, wherein a Mixed-Order Trajectory Alignment technique is devised to impose a more stringent constraint on the error bounds at each sampling step, facilitating a more linear probability flow. Extensive experiments on sparse-step fast sampling across multiple datasets demonstrate that our approach preserves the rapid convergence characteristics of high-speed samplers while maintaining superior generation quality. Code is publicly available at: <https://github.com/TaylorJocelyn/Sampling-aware-Quantization>.

## 1. Introduction

Diffusion models [12, 35, 37, 38] have demonstrated remarkable competitiveness in mainstream generative tasks [3, 17, 27, 33, 39, 43]. By harnessing the power of intricate posterior probability modeling and stable training regimes, these models effectively circumvent mode collapse, attain-

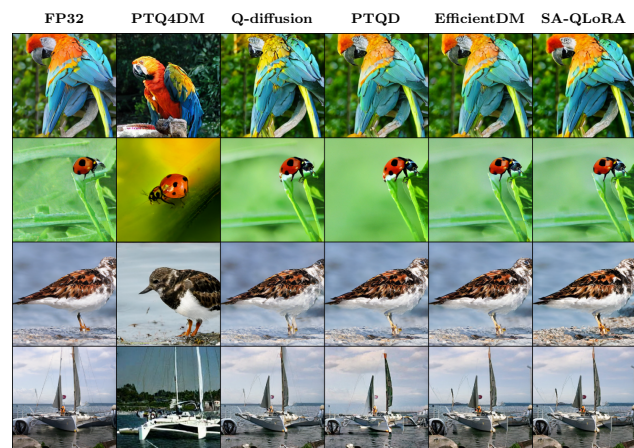


Figure 1. Comparison of generated samples on the ImageNet 256x256 dataset between full-precision LDM-4 and its quantized versions using PTQ4DM, Q-diffusion, PTQD, EfficientDM and our proposed SA-QLoRA.

ing superior generation fidelity and diversity when compared to GANs [1] and VAEs [15]. However, two primary computational bottlenecks impede the scalability and real-world applicability of diffusion models: the prolonged denoising chains [12], and the resource-intensive noise estimation networks. To address the former, advanced samplers [25, 26, 36, 45] have been developed to achieve efficient sampling trajectories with accurate approximations for stochastic differential equations (SDEs) [8, 22] and ordinary differential equations (ODEs) [24]. Regarding the latter, techniques such as quantization [10, 18, 32] have been employed to compress the noise estimation network, reducing both model size and the time and memory costs per iteration. While these two categories of approaches are generally regarded as separate components for accelerating diffusion, quantization errors disrupt the sampler’s directional evaluation, degrading high-speed sampling performance. This study, therefore, seeks to develop a sampling-

\*Corresponding author

aware quantization strategy aimed at achieving high-fidelity dual acceleration in diffusion models.

Specifically, quantization facilitates efficient model compression and inference acceleration by converting a pre-trained FP32 network into fixed-point networks with lower bit-width representations for weights and activations. This numerical transformation truncates the fractional components, inducing a distribution shift in both weights and activations. Consequently, in quantized diffusion models, quantization noise perturbs each directional estimation step, introducing directional bias. This issue is particularly pronounced in high-order samplers, where, within each interval  $(t_{i-1}, t_i)$  of the time schedule  $\{t_i\}_{i=1}^N$ , multiple directional estimations are required along the intermediate step sequence  $\{s_j \mid s_j \in (t_{i-1}, t_i), j = 1, \dots, n\}$  to establish a collectively estimated direction. Quantization noise perturbs each estimation at intermediate step  $s_j$ , substantially distorting the jointly estimated direction (refer to Sec. 3 for a detailed analysis). The multi-intermediate-step joint directional estimation, intrinsic to high-order samplers, aims to minimize truncation errors arising from the numerical solution of the continuous reverse diffusion equation, thus enabling efficient sparse-step trajectory sampling. However, quantization noise not only hinders the sampler’s rapid convergence potential but may also transform the stable probability flow ODE, designed for acceleration, into a variance-exploding SDE, inducing trajectory diffusion.

To mitigate the disruption caused by quantization algorithms on sampling acceleration and to optimally harness the advantages of both acceleration strategies, we propose a sampling-aware quantization technique. This method employs a *Mixed-Order Trajectory Alignment* strategy, thereby fostering a more linear probability flow through the quantization process. In essence, this approach imposes a more stringent constraint on the error bounds at each sampling step (see Sec. 3.3), effectively curbing error accumulation and averting sampling diffusion. Subsequently, we integrate the proposed sampling-aware quantization method atop the PTQ and QLoRA baselines, allowing for adaptation to various quantization bit-width requirements. Experiments on rapid sampling with sparse steps across diverse datasets reveal that our approach preserves the swift convergence capabilities of high-order samplers while upholding exceptional generative quality. In conclusion, our main contributions in this work are summarized as follows:

- We introduce a pioneering Sampling-Aware Quantization framework for diffusion models, examining the influence of quantization errors on rapid sampling through the lens of sampling acceleration principles, and presenting the Mixed-Order Trajectory Alignment strategy to foster a more linear probability flow during quantization.
- To accommodate diverse quantization bit-width requirements, we tailor sampling-aware quantization to post-

training quantization and QLoRA, culminating in the development of SA-PTQ and SA-QLoRA variants.

- Extensive experiments with sparse-step fast sampling across multiple datasets demonstrate that our method preserves the rapid convergence properties of high-speed samplers while maintaining superior generation quality.

## 2. Related Work

### 2.1. Efficient Diffusion Models

Diffusion models achieve impressive generation quality and diversity but are constrained by generation speed. Existing research accelerates diffusion from two main perspectives: optimizing generation trajectories for greater efficiency and compressing the noise estimation network to reduce the computational cost per iteration. In the first category, some works [15, 16] use learning-based methods to optimize  $\sigma(t)$  for efficient generation trajectories. By adjusting the signal-to-noise ratio distribution, they minimize the variance of the variational lower bound (VLB), enabling the model to approximate the target distribution more stably and efficiently, reducing unnecessary noise accumulation. Other works focus on learning-free samplers [25, 26, 41, 45], achieving high-precision numerical approximations of the sampling SDE and ODE, allowing larger sampling step sizes while controlling discretization truncation errors. Further related work on sampling acceleration can be found in Appendix B. The second category applies model lightweighting paradigms such as distillation [31, 44], pruning [4, 9], and quantization [10, 18, 32]. Distillation and pruning require extensive parameter training, whereas quantization, widely adopted for deployment, can be implemented with minimal or no additional training by adjusting only a few quantization parameters. In this paper, we focus on the joint optimization of learning-free sampling and quantization.

### 2.2. Model Quantization

Quantization is a mainstream technique for model compression and computational acceleration, converting FP32 weights and activations to low-bit fixed-point counterparts. Two primary approaches exist: post-training quantization (PTQ) [7, 29], which determines quantization parameters by minimizing the MSE or cross-entropy [28] between pre- and post-quantization tensors, and quantization-aware training (QAT) [20, 23], which trains the network to learn optimal parameters by modeling quantization noise. Common asymmetric quantization involves three parameters: scale factor  $s$ , zero point  $z$ , and quantization bit-width  $b$ . A floating-point value  $x$  is quantized to a fixed-point value  $x_{int}$  through the preceding parameters:

$$x_{int} = \text{clamp} \left( \left\lfloor \frac{x}{s} \right\rfloor + z, 0, 2^b \right), \quad (1)$$

where  $\lfloor \cdot \rfloor$  denotes rounding, with clamp as truncation.

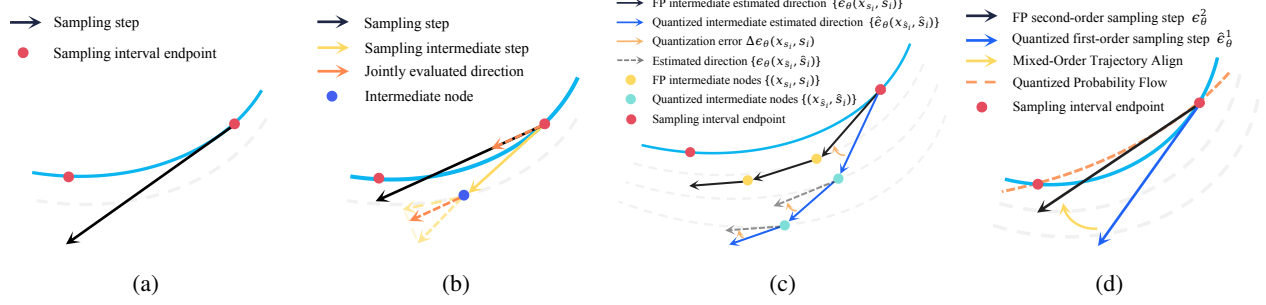


Figure 2. Direction estimation in reverse diffusion sampling. (a) The first-order sampler performs a single direction estimation at the beginning of the sampling interval. (b) The second-order sampler refines the direction estimation by evaluating additional intermediate steps within the interval. (c) Quantization errors lead to deviations in direction estimation, causing the intermediate steps in high-order samplers to drift over time, ultimately impacting the final direction estimation. (d) Our proposed Mixed-Order Trajectory Alignment achieves a more linearized probability flow.

## 2.3. Diffusion Model Quantization

Current research on diffusion model quantization remains relatively sparse. PTQ4DM [32] introduces a normal-distribution time-step calibration method, but in their work the experiments are only conducted on low-resolution datasets. Q-Diffusion [18] presents a time-step-aware calibration and shortcut-splitting quantization for U-Net. PTQD [11] applies a PTQ error correction method that requires additional statistical parameters during inference. TDQ [34] proposes a time-dynamic quantization strategy with a trained auxiliary network to estimate quantization parameters across time steps. EfficientDM [10] develops QALoRA for low-bit quantization, though additional training is required. These studies primarily focus on adapting traditional quantization techniques to the multi-time-step framework of diffusion models, while overlooking the inevitable impact of quantization noise on high-speed sampling. In contrast, our work integrates sampling acceleration to formulate a high-fidelity dual-acceleration scheme.

## 3. Preliminaries

### 3.1. Diffusion Models

Diffusion models progressively adds isotropic Gaussian noise to real data  $x_0$  and learn the denoising process by approximating the posterior probability distribution  $\{p(x_{t-1}|x_t)\}_{t=T}^1$ . The forward process can be modeled as an SDE:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t) dt + g(t) d\mathbf{w}, \quad (2)$$

where  $\mathbf{w}$  is the standard Wiener process (*a.k.a.*, Brownian motion),  $\mathbf{f}(\cdot, t) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a vector-valued function called the drift coefficient of  $\mathbf{x}(t)$ , and  $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  is a scalar function known as the diffusion coefficient of  $\mathbf{x}(t)$ .

The predominant sampling methodologies are categorized into deterministic and stochastic sampling. Stochastic

sampling follows Anderson’s reverse-time SDE:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t) d\bar{\mathbf{w}}, \quad (3)$$

where  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  is the score function [2],  $\bar{\mathbf{w}}$  is a standard Wiener process when time flows backwards from  $T$  to  $0$ , and  $dt$  is an infinitesimal negative timestep. Deterministic sampling follows the *probability flow* ODE:

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt, \quad (4)$$

which shares the same marginal probability as the reverse-time SDE. By eliminating the need for random noise sampling during the generation process, it achieves a more stable and smoother trajectory, facilitating integration with efficient numerical solvers.

To estimate the score  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  in Eqn. (23) and Eqn. (24), it is common to train a time-independent score-based model  $\mathbf{s}_\theta(\mathbf{x}, t)$ , which is linearly related to the noise estimation network  $\epsilon_\theta$ :

$$\mathbf{s}_\theta(\mathbf{x}, t) \triangleq -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t}, \quad (5)$$

where  $\sigma_t$  is the standard deviation of  $p(\mathbf{x}_t|\mathbf{x}_0)$ , referred to as the noise schedule.

### 3.2. High-Speed Sampling

The inverse sampling process is equivalent to substituting Eqn. (25) into Eqn. (24) for integration, i.e., given an initial sample  $\mathbf{x}_s$  at time  $s > 0$ , the solution  $\mathbf{x}_t$  at each  $t < s$  satisfies:

$$\mathbf{x}_t = \mathbf{x}_s + \int_s^t \left( \mathbf{f}(\mathbf{x}_\tau, \tau) + \frac{g(\tau)^2 \epsilon_\theta(\mathbf{x}_\tau, \tau)}{2\sigma_\tau} \right) d\tau \quad (6)$$

It is evident that  $\epsilon_\theta(\mathbf{x}_\tau, \tau)$  directly affecting the sampling direction. Following the setup of DDPM [12], where

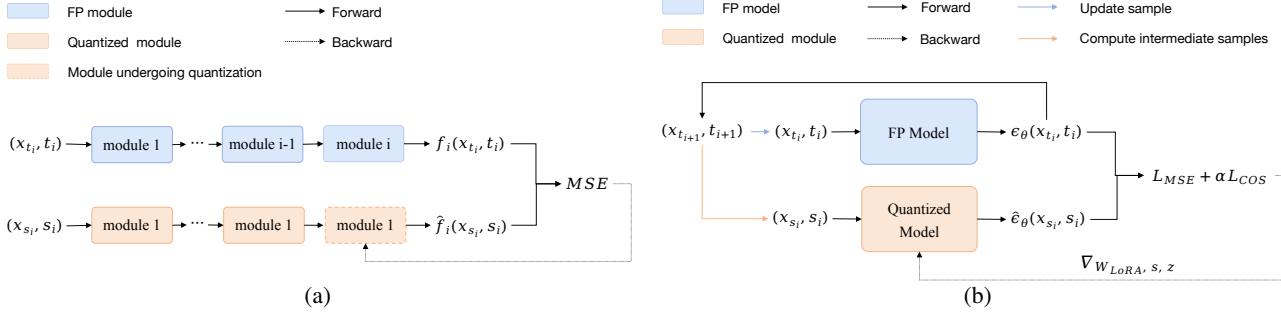


Figure 3. Sampling-aware quantization workflow. (a) Module-level reconstruction process employed in SA-PTQ, where  $\hat{f}_i(\cdot)$  denotes the module undergoing quantization and reconstruction. (b) Basic fine-tuning workflow in SA-QLoRA, where LoRA weights  $W_{LoRA}$  and quantization parameters  $s, z$  are iteratively updated after each sampling step.

$p(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$ , derive the expressions for  $\mathbf{f}(\mathbf{x}, t)$  and  $\mathbf{g}(t)$ . Then, define  $\lambda = \log(\frac{\alpha_t}{\sigma_t})$ , and further change the subscripts of  $\mathbf{x}$  and  $\epsilon_\theta$  from  $t$  to  $\lambda$ , where  $\mathbf{x}_\lambda$  denotes  $\mathbf{x}_{t_\lambda(\lambda)}$ , resulting in the following integration [25]:

$$\mathbf{x}_t = \frac{\alpha_t}{\alpha_s} \mathbf{x}_s + \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \epsilon_\theta(\mathbf{x}_\lambda, \lambda) d\lambda \quad (7)$$

Due to the intractability of nonlinear integration in nonlinear networks  $\epsilon_\theta(\mathbf{x}_\lambda, \lambda)$ , numerical approximation of the sampling direction in continuous equations is achieved by performing a high-order expansion of  $\epsilon_\theta(\mathbf{x}_\lambda, \lambda)$  at  $\lambda_s$  [25]:

$$\begin{aligned} \mathbf{x}_t &= \frac{\alpha_t}{\alpha_s} \mathbf{x}_s - \alpha_t \sum_{n=0}^{k-1} \epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \\ &+ \mathcal{O}((\lambda_t - \lambda_s)^{k+1}), \end{aligned} \quad (8)$$

where  $\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) = \frac{d^n \epsilon_\theta(\mathbf{x}_{\lambda_s}, \lambda_s)}{d\lambda^n}$  is the  $n$ -th order total derivative of w.r.t.  $\lambda$ . In practice, the  $k$ -th order expansion involves selecting  $k - 1$  intermediate points  $\{\lambda_i\}_{i=0}^{k-2}$  within  $(\lambda_s, \lambda_t)$ , and using the corresponding  $\epsilon_\theta(\mathbf{x}_{\lambda_i}, \lambda_i)$  to achieve a more accurate approximation of the sampling direction (more details in Appendix A.1), as shown in Fig. 2b. Under the condition of limited truncation error, acceleration is achieved by increasing the sampling step size.

### 3.3. Pre-analysis: Quantization Error Interference in High-Speed Sampling

The quantized network  $\hat{\epsilon}_\theta$  inevitably introduces quantization noise  $\Delta\epsilon_\theta$ , resulting in a deviation in the numerical integration in Eqn. (28) as follows:

$$\begin{aligned} \mathbf{x}_t &= \frac{\alpha_t}{\alpha_s} \mathbf{x}_s - \alpha_t \sum_{n=0}^{k-1} \hat{\epsilon}_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \\ &+ \mathcal{O}((\lambda_t - \lambda_s)^{k+1}) \\ &= \frac{\alpha_t}{\alpha_s} \mathbf{x}_s - \alpha_t \sum_{n=0}^{k-1} \epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \\ &+ \sum_{n=0}^{k-1} \Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \\ &+ \mathcal{O}((\lambda_t - \lambda_s)^{k+1}) \end{aligned} \quad (9)$$

where the deviation of the derivative sequence  $\{\Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s)\}_{n=0}^{k-1}$  essentially implies a positional shift of the sampling intermediate points (as described in Sec. 3.2), along with a directional estimation shift at these intermediate points, as shown in Fig. 2c. We denote  $\Delta_{quant} = \sum_{n=0}^{k-1} \Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda$  as the quantization cumulative error term, and  $\Delta_{disc}$  as the discretization truncation error term, with  $\Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s)$  simplified to  $\delta$ . As analyzed in Appendix A.2, the total error upper bound for the numerical integration in Eqn. (29), utilizing the  $(k - 1)$ -th order expansion term, can be expressed as:

$$\begin{aligned} \mathcal{L}_\Delta &= \mathcal{L}_{\Delta_{quant}} + \mathcal{L}_{\Delta_{disc}} \\ &= \mathcal{O}(\delta \cdot e^{-\lambda_s} \cdot (\lambda_t - \lambda_s)) + \mathcal{O}((\lambda_t - \lambda_s)^{k+1}) \end{aligned} \quad (10)$$

It is evident that the quantization cumulative error, controlled directly by  $\delta$ , dominates the total error, significantly impacting the rapid convergence of ODE sampling. This cumulative effect is exacerbated by the model's nonlinearity, which amplifies the propagation of quantization errors through higher-order terms. To address this issue, we redesign the quantization scheme to learn a more linear probability flow, bringing  $\delta$  closer to the order of  $\mathcal{O}(\lambda_t - \lambda_s)$  to further constrain the error bound and promote convergence.

## 4. Methodology

In this section, we introduce an innovative Sampling-Aware Quantization framework. We begin by presenting the core component—the Mixed-Order Trajectory Alignment strategy—in Sec. 4.1, using the DPM-Solver sampler as a case study. To facilitate 8-bit quantization, we propose a sampling-aware post-training quantization method in Sec. 4.2. Furthermore, to accommodate lower-bit quantization requirements (e.g., W4A4), this approach is extended in Sec. 4.3 with a sampling-aware Quantized Low-Rank Adaptation (QLoRA) method. Finally, the adaptation scheme for more generalized samplers is presented in

Sec. 4.4, boosting the framework’s versatility.

#### 4.1. Mixed-Order Trajectory Alignment

High-fidelity quantization for diffusion models is typically achieved by aligning the sampling trajectory of the full-precision model with that of the quantized counterpart.

**Sampling Trajectory.** In the reverse diffusion process, the intermediate sample set  $\{\mathbf{x}_t\}_{t=T}^0$ , obtained by numerically integrating the sampling equation from an initial point  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$  along a time schedule  $\{t_i\}_{i=T}^0$ , is conventionally referred to as the sampling trajectory. Thus, the sampling trajectory is governed by three primary factors: the sampler, which defines the numerical solution method; the initial sampling point  $\mathbf{x}_T$ ; and the time schedule  $\{t_i\}_{i=T}^0$ , which can be mapped to a noise schedule  $\{\sigma_i\}_{i=T}^0$ . As detailed in Sec. 3.2,  $\epsilon_\theta(\mathbf{x}_t, t)$  dictates the sampling direction at each step  $t$ . Consequently, once the aforementioned conditions are specified, the sampling trajectory becomes fully determined, with each sampling direction at each step uniquely defined. This establishes a one-to-one correspondence between the direction sequence  $\{\epsilon_\theta(\mathbf{x}_t, t)\}_{t=T}^0$  and the sample trajectory sequence  $\{\mathbf{x}_t\}_{t=T}^0$ . Therefore, we clarify that the trajectory aligned in this work is, in essence, the direction sequence  $\{\epsilon_\theta(\mathbf{x}_t, t)\}_{t=T}^0$ . Accordingly, the core quantization objective can be formulated as:

$$\hat{\epsilon}_\theta = \mathcal{Q}(\epsilon_\theta, s, z)$$

$$\arg \min_{s, z} \mathbb{E}_{(\mathbf{x}_t, t) \sim \mathcal{D}} \|\epsilon_\theta(\mathbf{x}_t, t) - \hat{\epsilon}_\theta(\mathbf{x}_t, t)\|^2$$

Here,  $\mathcal{Q}$  represents the quantization function,  $s$  and  $z$  denote the quantization parameters, scale and zero-point respectively, and  $\mathcal{D}$  refers to the sampling distribution of the calibration dataset.

As analyzed in Sec. 3.2, the quantization noise  $\Delta\epsilon_\theta$  impinges upon the directional estimation of high-speed samplers, particularly exerting cumulative effects on higher-order samplers, which necessitate multiple evaluations of higher-order derivatives, ultimately resulting in trajectory deviation. To counteract the swift accumulation of errors, we employ mixed-order trajectory quantization to foster a more linear probability flow. As detailed in Appendix A.1, for sampling within the interval  $(\lambda_{t_{i-1}}, \lambda_{t_i})$ , a first-order sampler directly evaluates  $\epsilon_\theta(\mathbf{x}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}})$  to determine the sampling direction. In contrast, a  $k$ -order sampler generates  $k - 1$  intermediate points  $\{s_i\}_{i=0}^{k-2}$  and evaluates  $\epsilon_\theta(\mathbf{x}_{\lambda_{s_i}}, \lambda_{s_i})$  at each point  $s_i$ , iteratively refining the direction based on  $\epsilon_\theta(\mathbf{x}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}})$ . Each  $\epsilon_\theta(\mathbf{x}_{\lambda_{s_i}}, \lambda_{s_i})$  encodes higher-order derivative information, thereby enhancing the directional precision of the sampling process. Drawing inspiration from this, we achieve mixed-order trajectory quantization by aligning the quantized first-

order sampling direction trajectory  $\{\hat{\epsilon}_\theta(\mathbf{x}_{\lambda_t}, \lambda_t)\}_{t=T}^0$  with the full-precision higher-order sampling direction trajectory  $\{\epsilon_\theta(\mathbf{x}_{\lambda_s}, \lambda_s)\}_{s \in \mathcal{S}}$  at the intermediate nodes. The quantization objective is defined as:

$$\hat{\epsilon}_\theta = \mathcal{Q}(\epsilon_\theta, s, z) \quad (11)$$

$$\arg \min_{s, z} \mathbb{E}_{(\mathbf{x}_t, t) \sim \mathcal{D}, (\mathbf{x}_s, s) \sim \mathcal{S}} \|\hat{\epsilon}_\theta(\mathbf{x}_{\lambda_s}, \lambda_s) - \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t)\|^2 \quad (12)$$

For example, taking DPM-Solver sampling as a case study. Given an initial value  $\mathbf{x}_T$  and  $M + 1$  time steps  $\{t_i\}_{i=0}^M$  decreasing from  $t_0 = T$  to  $t_M = 0$ . Starting with  $\tilde{\mathbf{x}}_{t_0} = \mathbf{x}_T$ , the DPM-Solver-1 sampling sequence  $\{\tilde{\mathbf{x}}_{t_i}\}_{i=1}^M$  is computed iteratively as follows:

$$\tilde{\mathbf{x}}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{t_i} (e^{h_i} - 1) \epsilon_\theta(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1}), \quad (13)$$

where  $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$ . The corresponding sampling formula for DPM-Solver-2 is given in Alg. 1.

---

#### Algorithm 1 DPM-Solver-2

---

**Require:** Initial value  $\mathbf{x}_T$ , time steps  $\{t_i\}_{i=0}^M$ , model  $\epsilon_\theta$

- 1:  $\tilde{\mathbf{x}}_{t_0} \leftarrow \mathbf{x}_T$
- 2: **for**  $i \leftarrow 1$  to  $M$  **do**
- 3:    $s_i \leftarrow t_\lambda \left( \frac{\lambda_{t_{i-1}} + \lambda_{t_i}}{2} \right)$
- 4:    $\mathbf{u}_i \leftarrow \frac{\alpha_{s_i}}{\alpha_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{s_i} \left( e^{\frac{h_i}{2}} - 1 \right) \epsilon_\theta(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})$
- 5:    $\tilde{\mathbf{x}}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\mathbf{x}}_{t_{i-1}} - \sigma_{t_i} (e^{h_i} - 1) \epsilon_\theta(\mathbf{u}_i, s_i)$
- 6: **end for**
- 7: **return**  $\tilde{\mathbf{x}}_{t_M}$

---

By aligning the quantized directional term  $\hat{\epsilon}_\theta(\tilde{\mathbf{x}}_{t_{i-1}}, t_{i-1})$  with the full-precision directional term  $\epsilon_\theta(\mathbf{u}_i, s_i)$ , we can effectively linearize higher-order sampling trajectories, as illustrated in Fig. 2d.

#### 4.2. Sampling-Aware Post-Training Quantization

For 8-bit quantization, we propose a Sampling-Aware Post-Training Quantization scheme (SA-PTQ), with the process illustrated in Fig. 3a. In alignment with prior work [11, 18, 32], we adopt the widely utilized BRECQ [19] as the baseline quantization algorithm, utilizing Adaround for weight quantization while training only the minimal parameter  $\alpha$ . Building upon the method outlined in Sec. 4.1, we develop a novel dual-order trajectory calibration strategy to guide module reconstruction. This strategy comprises two key components: Dual-order Trajectory Sampling and Mixed-Order Trajectory Alignment Calibration.

**Dual-Order Trajectory Sampling.** Given a predefined seed  $seed$  and time schedule  $\{t_i\}_{i=N}^0$ , we initiate sampling from an initial sample  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Utilizing a first-order sampler, we collect input samples  $\{t_i\}_{i=N}^0$  at each noise

estimation iteration to establish the first-order trajectory, where *cond* encapsulates the conditional information. Subsequently, we input the same  $\mathbf{x}_T$  into a second-order sampler, obtaining input samples  $\{(\mathbf{x}_{s_i}, s_i, \text{cond})\}_{s_i \in (t_{i-1}, t_i)}$  at each intermediate point  $s_i$  within the interval  $(t_{i-1}, t_i)$ , thus forming the second-order trajectory.

**Mixed-Order Trajectory Alignment Calibration.** Let  $f_i(\cdot)$  represent the  $i$ -th module within the noise estimation network requiring reconstruction, and  $\hat{f}_i(\cdot)$  denote its quantized counterpart. The SA-PTQ approach attains high-fidelity quantization by reconstructing each module independently. The objective for reconstructing each module is formulated as:

$$\hat{f}_i = \text{Adaround}(f_i, \alpha) \quad (14)$$

$$\arg \min_{\alpha} \mathbb{E}_{(t_j, s_j)} \|f_i(\mathbf{x}_{t_j}, t_j, \text{cond}) - \hat{f}_i(\mathbf{x}_{s_j}, s_j, \text{cond})\|^2 \quad (15)$$

### 4.3. Sampling-Quantization Dual-Aware LoRA

To meet the demands of low-bit quantization, we integrate Mixed-Order Trajectory Alignment with QLoRA, introducing the Sampling-Quantization Dual-Aware LoRA (SA-QLoRA) framework. The workflow is depicted in Fig. 3b.

For the basic QLoRA, training is supervised by aligning  $\hat{\epsilon}_{\theta}(\mathbf{x}_{t_i}, t_i)$  and  $\epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)$  at each time step  $t_i$ , guiding the optimization of both the LoRA weights  $w$  and the quantization parameters  $s$  and  $z$ . The quantization objective can be formulated as:

$$\hat{\epsilon}_{\theta} = \text{QLoRA}(\epsilon_{\theta}, w, s, z) \quad (16)$$

$$\arg \min_{w, s, z} \mathbb{E}_{(\mathbf{x}_{t_i}, t_i) \sim \mathcal{D}} \|\hat{\epsilon}_{\theta}(\mathbf{x}_{t_i}, t_i) - \epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)\|^2 \quad (17)$$

In light of the analysis in Sec. 3.2,  $\epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)$  directly determines the sampling direction, we introduce an additional directional constraint  $\mathcal{L}_{COS}$  to enhance the Mixed-Order Trajectory Alignment constraint  $\mathcal{L}_{MOTA}$  in supervising QLoRA training, with the resulting training objective formulated as:

$$\mathcal{L}_{COS} = 1 - \frac{\langle \epsilon_{\theta}(\mathbf{x}_{t_i}, t_i), \hat{\epsilon}_{\theta}(\mathbf{x}_{s_i}, s_i) \rangle}{\|\epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)\| \|\hat{\epsilon}_{\theta}(\mathbf{x}_{s_i}, s_i)\|} \quad (18)$$

$$\mathcal{L}_{MOTA} = \mathbb{E}_{(t_i, s_i)} \|\hat{\epsilon}_{\theta}(\mathbf{x}_{s_i}, s_i) - \epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)\|^2 \quad (19)$$

$$\arg \min_{w, s, z} \mathcal{L}_{COS} + \mathcal{L}_{MOTA}, \quad (20)$$

where the set  $\{t_i\}_{i=N}^0$  represents the collection of first-order sampling points, while  $\{s_i\}_{i=N}^0$  denotes the additional intermediate evaluation points for second-order sampling.

### 4.4. Adaptation to Generalized Samplers

As the classical numerical method-based generalized solver for diffusion ODEs, DPM-Solver’s [25] sampling-aware quantization adaptation scheme is discussed in Sec. 4.1. Additionally, DDIM [36] is proven to exhibit identical updates to DPM-Solver-1 [25], and can thus be regarded as a specific instance of DPM-Solver-1 under a particular noise schedule for sampling lower-order trajectories.

**PLMS sampler.** PNDM [22] decomposes the numerical sampling equation into a gradient part and a transfer part, and defines the pseudo-numerical sampling equation by introducing nonlinear transfer parts  $\phi(\cdot)$ , as follows:

$$\phi(\mathbf{x}_t, \epsilon_{\theta}^{(t)}, t, t - \delta) = \frac{\sqrt{\bar{\alpha}_{t-\delta}}}{\sqrt{\bar{\alpha}_t}} \mathbf{x}_t - \frac{(\bar{\alpha}_{t-\delta} - \bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t} \left( \sqrt{(1 - \bar{\alpha}_{t-\delta})\bar{\alpha}_t} + \sqrt{(1 - \bar{\alpha}_t)\bar{\alpha}_{t-\delta}} \right)} \epsilon_{\theta}^{(t)}, \quad (21)$$

where  $\bar{\alpha}_i$  is a parameter related to the noise schedule, and  $\epsilon_{\theta}^{(t)}$  is the gradient part that determines the sampling direction. Various well-established numerical methods can be employed to estimate  $\epsilon_{\theta}^{(t)}$  (e.g., the linear multi-step method applied in **Pseudo-Linear Multi-Step samplers**), with different orders of numerical methods corresponding to trajectories of different orders. Therefore, we achieve Mixed-Order Trajectory Alignment by aligning  $\epsilon_{\theta}^{(t)}$  derived from numerical methods of varying orders.

## 5. Experiments

### 5.1. Settings

**Benchmarks and Metrics.** We evaluated the proposed SA-PTQ and SA-QLoRA across multiple benchmarks: using LDM [30] for class-conditional image generation on ImageNet  $256 \times 256$ ; LDM for unconditional image generation on LSUN-Churches  $256 \times 256$  and LSUN-Bedroom  $256 \times 256$  [42]; and SD-v1.4 for text-guided image generation on MS-COCO  $512 \times 512$  [21]. For the first three benchmarks, we employ metrics such as Fréchet Inception Distance (FID), Sliding Fréchet Inception Distance (sFID), Inception Score (IS), precision, and recall to comprehensively evaluate algorithm performance. For each evaluation, we generate 50k samples and calculate these metrics using the OpenAI’s evaluator [6], with BOPs (Bit Operations) as the efficiency metric. For the text-to-image benchmark, we further incorporate CLIP-Score to evaluate text-image consistency, generating 30k samples per evaluation round.

**Model and Sampling settings.** For both class-conditional and unconditional image generation, we adopt DPM-Solver-1 and DPM-Solver-2 as the low-order and high-order samplers, respectively, in the SA-PTQ and SA-QLoRA frameworks to achieve mixed-order trajectory alignment. Our primary focus is on two parameters of

Table 1. Performance evaluation of class-conditioned image generation on the ImageNet  $256 \times 256$  dataset using LDM-4 with 20 sampling steps of DPM-Solver-2.

Model	Method	Bits (W/A)	Size (MB)	BOPs (T)	IS $\uparrow$	FID $\downarrow$	sFID $\downarrow$	Precision $\uparrow$	Recall $\uparrow$
LDM-4 ( <i>scale</i> = 1.5, <i>steps</i> = 20)	FP	32/32	1742.72	102.20	174.33	9.45	8.08	77.22%	52.25%
	PTQ4DM	8/8	436.79	8.76	115.06	11.43	12.19	60.21%	51.26%
	Q-diffusion	8/8	436.79	8.76	120.14	10.98	11.34	63.97%	54.34%
	PTQD	8/8	436.79	8.76	<b>122.46</b>	10.76	10.58	62.08%	56.16%
	SA-PTQ (ours)	8/8	436.79	8.76	120.71	<b>10.16</b>	<b>9.89</b>	<b>65.39%</b>	<b>56.97%</b>
	PTQ4DM	4/8	219.12	4.38	122.75	10.14	12.73	69.86%	51.03%
	Q-diffusion	4/8	219.12	4.38	130.69	9.76	10.92	68.45%	51.97%
	PTQD	4/8	219.12	4.38	127.41	9.16	9.72	72.80%	50.41%
	EfficientDM	4/8	219.12	4.38	132.70	9.91	8.76	71.05%	53.62%
	SA-QLoRA (ours)	4/8	219.12	4.38	<b>140.56</b>	<b>8.55</b>	<b>8.51</b>	<b>73.20%</b>	<b>54.49%</b>
	PTQ4DM	4/4	219.12	2.19	-	-	-	-	-
	Q-diffusion	4/4	219.12	2.19	-	-	-	-	-
	PTQD	4/4	219.12	2.19	-	-	-	-	-
	EfficientDM	4/4	219.12	2.19	225.20	17.28	13.78	<b>60.33%</b>	52.82%
	SA-QLoRA (ours)	4/4	219.12	2.19	<b>242.03</b>	<b>13.73</b>	<b>12.45</b>	58.90%	<b>55.38%</b>

the generative sampler within LDM: the classifier-free guidance scale *scale* and the number of sampling steps *steps*. For class-conditional generation, we set *steps*=20, *scale*=1.5, while for unconditional generation, we set *steps*=50. For text-to-image tasks, we align the native PLMS sampling trajectory with its one-order-reduced counterpart, setting *steps*=50, *scale*=7.5.

**Quantization Settings.** We denote quantization of weights to  $x$ -bits and activations to  $y$ -bits as  $WxAy$ . For further details on the quantization settings and SA-QLoRA fine-tuning, please refer to Appendix.

## 5.2. Main Results

### 5.2.1. Class-conditional Generation

We first compare our proposed SA-PTQ and SA-QLoRA with previous approaches on class-conditioned image generation task. Specifically, we conduct evaluations on ImageNet  $256 \times 256$  using a pre-trained LDM-4 model with DPM-Solver-2 over 20 sampling steps. The results are presented in Table 1. In terms of efficiency, configurations  $W8A8$ ,  $W4A8$ , and  $W4A4$  achieve bit compression rates of 3.99x, 7.95x, and 7.95x, respectively, along with bit-operation acceleration rates of 11.47x, 23.33x, and 46.67x. In terms of generation quality, under  $W8A8$  and  $W4A8$  configurations, our proposed SA-PTQ and SA-QLoRA consistently demonstrate superior performance across all metrics, achieving the lowest FID and sFID scores of 8.55 and 8.51, respectively. Notably, the FID score is even 0.9 lower than that of the full-precision model. Under the  $W4A4$  configuration, previous work [11, 18, 32] introduce excessive quantization noise, transforming the originally deterministic probability flow ODE into a variance-exploding SDE, ultimately resulting in generation failure. In contrast, our SA-

LoRA demonstrates excellent convergence, with an sFID only 4.37 points higher than the full-precision model.

The consistently strong metrics across various quantization settings confirm that our mixed-order trajectory alignment strategy has, to a certain extent, achieved a more linear probability flow through quantization. Consequently, this approach effectively mitigates the rapid accumulation of high-order sampler errors induced by quantization, thereby preserving outstanding generative performance under fast sampling with sparse trajectories.

### 5.2.2. Unconditional Generation

We then thoroughly evaluate SA-PTQ and SA-QLoRA on unconditional generation tasks, employing the LDM-4 and LDM-8 models across the LSUN-Bedroom and LSUN-Church datasets, respectively. Tab. 2 and Tab.5(in Appendix) indicate that our approach narrows the gap with the full-precision model. Specifically, on the LSUN-

Table 2. Performance comparisons of unconditional image generation on LSUN-Bedroom  $256 \times 256$ .

LDM-4 (steps = 50)					
Method	W/A	FID $\downarrow$	sFID $\downarrow$	Prec. $\uparrow$	Rec. $\uparrow$
FP	32/32	6.17	13.92	64.30%	51.46%
PTQD	8/8	10.31	15.89	56.57%	54.15%
SA-PTQ	8/8	<b>9.83</b>	<b>13.85</b>	<b>56.86%</b>	<b>54.54%</b>
PTQD	4/8	10.96	15.42	47.83%	52.80%
EfficientDM	4/8	9.32	14.48	48.62%	<b>53.02%</b>
SA-QLoRA	4/8	<b>8.79</b>	<b>12.41</b>	<b>50.16%</b>	52.31%
PTQD	4/4	-	-	-	-
EfficientDM	4/4	19.30	22.63	<b>43.77%</b>	40.09%
SA-QLoRA	4/4	<b>14.72</b>	<b>18.19</b>	41.84%	<b>45.60%</b>

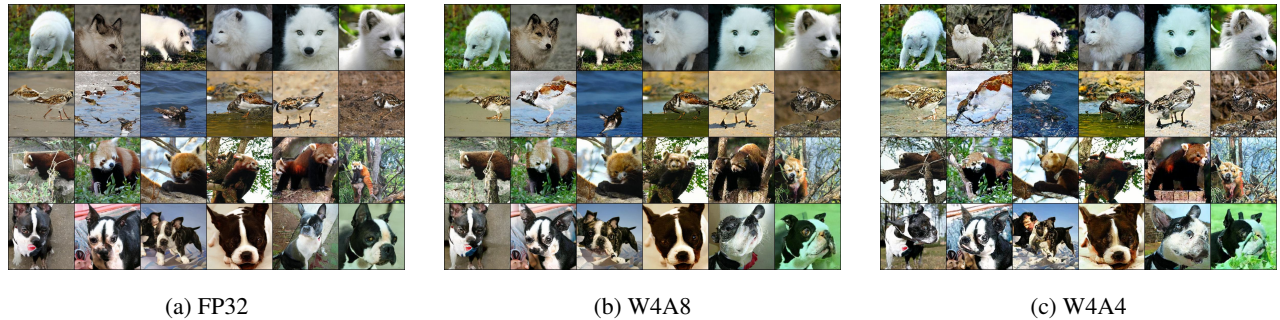


Figure 4. Visualization of the generative performance of our SA-QLoRA under  $W4A8$  and  $W4A4$  quantization settings.

Bedroom dataset, SA-PTQ under  $W8A8$  quantization reduces FID and sFID by 0.48 and 2.04, respectively, compared to PTQD. Furthermore, SA-QLoRA under  $W4A8$  achieves additional reductions of 1.04 and 1.44. Even under  $W4A4$  quantization, SA-QLoRA effectively controls quantization-induced errors, preventing variance explosion and achieving FID and sFID reductions of 4.58 and 4.44, respectively, relative to EfficientDM. On the LSUN-Church dataset, SA-PTQ achieves FID and sFID reductions of 1.22 and 0.46, respectively, over PTQD, while SA-QLoRA further improves these metrics with average reductions of 3.66 in FID and 0.82 in sFID. Experiments on the LSUN dataset further demonstrate that our sampling-aware quantization approach effectively preserves the superior generative performance of high-order samplers under sparse-step sampling, achieving high-fidelity quantization.

### 5.3. Text-guided Image Generation

We evaluate the text-to-image generation task using SD-v1.4 on MS-COCO  $512 \times 512$ , with the results presented in Tab. 3. Under  $W8A8$  quantization, SA-PTQ achieved consistently optimal metrics, particularly outperforming PTQD by 0.62 in sFID. In the  $W4A4$  setting, SA-QLoRA demonstrated the best performance in terms of FID and CLIP-Score. Further visual results are provided in Appendix.

Table 3. Performance evaluation of text-guided image generation on MS-COCO  $512 \times 512$ .

Method	W/A	TBOPs	FID ↓	sFID ↓	CLIP-Score. ↑
Q-Diffusion	8/8	51.8	13.28	20.65	0.2904
PTQD	8/8	51.8	13.65	20.14	0.3029
SA-PTQ	8/8	51.8	<b>13.10</b>	<b>19.52</b>	<b>0.3036</b>
Q-Diffusion	4/8	25.9	14.40	21.09	0.2875
EfficientDM	4/8	25.9	13.31	<b>19.92</b>	0.3002
SA-QLoRA	4/8	25.9	<b>13.27</b>	20.26	<b>0.3017</b>

### 5.4. Ablation Study

As shown in Tab. 4, we conduct ablation studies on the ImageNet  $256 \times 256$  dataset to validate the effectiveness of the

proposed sampling-aware quantization components. Here, MOTAC refers to the Mixed-Order Trajectory Alignment Calibration described in Sec. 4.2, while  $\mathcal{L}_{MOTA}$  and  $\mathcal{L}_{COS}$  represent the two constraints discussed in Sec. 4.3. On the PTQ baseline BRECQ, MOTA reduces FID and sFID by 4.1 and 3.83, respectively. Furthermore, MOTA achieves additional reductions of 0.95 in FID and 0.31 in sFID over QLoRA with applied direction alignment constraints. These results demonstrate that our proposed Mixed-Order Trajectory Alignment strategy effectively mitigates sampler performance degradation caused by quantization errors, achieving high-fidelity quantization.

Table 4. Ablation study of the sampling-aware quantization components using LDM-4 ( $scale = 1.5, step = 20$ ) on the ImageNet  $256 \times 256$ .

Method	W/A	IS ↑	FID ↓	sFID ↓
FP	32/32	174.33	9.45	8.08
BRECQ	8/8	112.80	14.26	13.72
+ MOTAC (SA-PTQ)	8/8	120.71	10.16	9.89
QLoRA	4/8	132.70	9.91	8.76
+ $\mathcal{L}_{COS}$	4/8	134.61	9.50	8.82
+ $\mathcal{L}_{MOTA}$ (SA-QLoRA)	4/8	140.56	8.55	8.51

## 6. Conclusion

In this paper, we present a sampling-aware quantization method for diffusion models, designed to achieve high-fidelity dual acceleration. We begin by analyzing the impact of quantization errors on sampling through the lens of sampling acceleration principles, and subsequently introduce a Mixed-Order Trajectory Alignment strategy to quantize a more linear probability flow, thereby mitigating the rapid accumulation of errors in high-speed samplers. Furthermore, we propose two variants, SA-PTQ and SA-QLoRA, to cater to diverse quantization bit-width requirements. Experimental results substantiate that our approach effectively curtails error accumulation during fast sampling, facilitating high-fidelity quantization.

**Acknowledgements.** This work is funded by National Natural Science Foundation of China (62576305) and Ningbo Youth Science and Technology Innovation Talent Project (2025QL052).

## References

- [1] Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1(1):100004, 2021. 1
- [2] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022. 3
- [3] Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with process diffusion. 2022. 1
- [4] Thibault Castells, Hyoung-Kyu Song, Bo-Kyeong Kim, and Shinkook Choi. Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. *arXiv preprint arXiv:2404.11936*, 2024. 2
- [5] Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. In *International Conference on Learning Representations*, 2026. 3
- [6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 6
- [7] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM international conference on multimedia*, pages 5380–5388, 2022. 2
- [8] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021. 1
- [9] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. *Advances in neural information processing systems*, 36, 2024. 2
- [10] Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Efficientdm: Efficient quantization-aware fine-tuning of low-bit diffusion models. *arXiv preprint arXiv:2310.03270*, 2023. 1, 2, 3
- [11] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 5, 7
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3
- [13] Marlis Hochbruck and Alexander Ostermann. Explicit exponential runge–kutta methods for semilinear parabolic problems. *SIAM Journal on Numerical Analysis*, 43(3):1069–1090, 2005. 1
- [14] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. 1
- [15] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 1, 2
- [16] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 2
- [17] Jin Sub Lee, Jisun Kim, and Philip M Kim. Proteinsgm: Score-based generative modeling for de novo protein design. *bioRxiv*, pages 2022–07, 2022. 1
- [18] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023. 1, 2, 3, 5, 7
- [19] 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. *arXiv preprint arXiv:2102.05426*, 2021. 5
- [20] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024. 2
- [21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 6
- [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, 6, 2
- [23] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023. 2
- [24] C Lu, Y Zhou, F Bao, J Chen, and C Li. A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Proc. Adv. Neural Inf. Process. Syst., New Orleans, United States*, pages 1–31, 2022. 1
- [25] 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 1, 2, 4, 6
- [26] 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 preprint arXiv:2211.01095*, 2022. 1, 2
- [27] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 1
- [28] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort.

- A white paper on neural network quantization. arxiv 2021. *arXiv preprint arXiv:2106.08295*. 2
- [29] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020. 2
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 6
- [31] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2
- [32] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1972–1981, 2023. 1, 2, 3, 5, 7
- [33] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1
- [34] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1
- [36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 6, 2
- [37] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1
- [38] 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
- [39] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: Diffusion-based image super-resolution in a single step. *arXiv preprint arXiv:2311.14760*, 2023. 1
- [40] Zeqing Wang, Gongfan Fang, Xinyin Ma, Xingyi Yang, and Xinchao Wang. Sparsed: Sparse attention for diffusion language models. In *International Conference on Learning Representations*, 2026. 3
- [41] Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *Advances in Neural Information Processing Systems*, 36:76806–76838, 2023. 2
- [42] 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
- [43] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 1
- [44] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*, 2024. 2
- [45] Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7777–7786, 2024. 1, 2