

# Sampling-Aware Quantization for Diffusion Models

## Supplementary Material

### A. Theoretical Analysis

#### A.1. High-Order Approximation via Intermediate Point Evaluations in Numerical Integration

For the following ODE [25]:

$$\frac{d\mathbf{x}_t}{dt} = \alpha \mathbf{x}_t + N(\mathbf{x}_t, t), \quad (21)$$

where  $\alpha \in \mathbb{R}$  and  $N(\mathbf{x}_t, t) \in \mathbb{R}^D$  is a non-linear function of  $\mathbf{x}_t$ . Given an initial value  $\mathbf{x}_t$  at time  $t$ , for  $h > 0$ , the true solution at time  $t + h$  is:

$$\mathbf{x}_{t+h} = e^{\alpha h} \mathbf{x}_t + e^{\alpha h} \int_0^h e^{-\alpha \tau} N(\mathbf{x}_{t+\tau}, t + \tau) d\tau. \quad (22)$$

The exponential Runge-Kutta methods [13, 14] use some intermediate points to approximate the integral  $\int_0^h e^{-\alpha \tau} N(\mathbf{x}_{t+\tau}, t + \tau) d\tau$ . Accordingly, DPM-Solver adopts this method to compute the analogous integral in Eqn. (23) with  $\alpha = 1$  and  $N = \epsilon_\theta$ :

$$\mathbf{x}_{\lambda_t+h} = \frac{\alpha_{\lambda_t+h}}{\alpha_{\lambda_t}} \mathbf{x}_{\lambda_t} + \alpha_{\lambda_t+h} \int_{\lambda_t}^{\lambda_t+h} e^{-\lambda} \epsilon_\theta(\mathbf{x}_\lambda, \lambda) d\lambda \quad (23)$$

This is equivalent to approximating the continuous integral using a higher-order Taylor expansion of  $\mathbf{x}(\lambda + h)$  at  $\lambda = \lambda_t$ . For an in-depth theoretical foundation of numerical methods, refer to [13, 14]. Here, we present a concise derivation of the expansion corresponding to the second-order Runge-Kutta method.

First, we make the following assumptions to ensure the applicability of the  $k$ -th order Taylor expansion:

**Assumption #1:** The total derivatives of  $\epsilon_\theta(\mathbf{x}_\lambda, \lambda)$ , denoted as  $\frac{\partial^j \epsilon_\theta(\mathbf{x}_\lambda, \lambda)}{\partial \mathbf{x}_\lambda^j}$  and  $\frac{\partial^j \epsilon_\theta(\mathbf{x}_\lambda, \lambda)}{\partial \lambda^j}$ , exist and are continuous for all  $0 \leq j \leq k + 1$ .

**Assumption #2:** The step size  $h = \lambda_t - \lambda_s$  satisfies  $h = \mathcal{O}(\frac{1}{N})$ , where  $N$  is the number of integration steps, ensuring the step size is sufficiently small.

**Analysis.** In denoising diffusion, for the simplified probability flow integral:

$$\mathbf{x}_{\lambda_t+h} = \mathbf{x}_{\lambda_t} + \int_{\lambda_t}^{\lambda_t+h} \epsilon_\theta(\mathbf{x}_\lambda, \lambda) d\lambda, \quad (24)$$

the general form of the second-order Runge-Kutta method is:

$$\begin{cases} k_1 = \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t), \\ k_2 = \epsilon_\theta(\mathbf{x}_{\lambda_t} + bhk_1, \lambda_t + ah), \\ \mathbf{x}_{\lambda_t+h} = \mathbf{x}_{\lambda_t} + h \left[ \left(1 - \frac{1}{2a}\right) k_1 + \frac{1}{2a} k_2 \right]. \end{cases} \quad (25)$$

For the classical midpoint method, taking  $a = b = \frac{1}{2}$ , we have:

$$\begin{cases} k_1 = \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t), \\ k_2 = \epsilon_\theta(\mathbf{x}_{\lambda_t} + \frac{h}{2} k_1, \lambda_t + \frac{h}{2}), \\ \mathbf{x}_{\lambda_t+h} = \mathbf{x}_{\lambda_t} + hk_2. \end{cases} \quad (26)$$

Then, for  $k_2$ , perform a first-order Taylor expansion of  $\epsilon_\theta(\mathbf{x}_\lambda, \lambda)$  at  $(\mathbf{x}_{\lambda_t}, \lambda_t)$ , yielding:

$$\begin{aligned} k_2 &= \epsilon_\theta(\mathbf{x}_{\lambda_t} + \frac{h}{2} k_1, \lambda_t + \frac{h}{2}) \\ &= \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{\partial \epsilon_\theta(\mathbf{x}_\lambda, \lambda)}{\partial \lambda} \Big|_{(\mathbf{x}_{\lambda_t}, \lambda_t)} \cdot \frac{h}{2} + \mathcal{O}(h^2) \\ &\quad + \frac{\partial \epsilon_\theta(\mathbf{x}_\lambda, \lambda)}{\partial \mathbf{x}_\lambda} \Big|_{(\mathbf{x}_{\lambda_t}, \lambda_t)} \cdot \frac{h}{2} k_1 \\ &= \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{h}{2} \frac{\partial \epsilon_\theta}{\partial \lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{h}{2} \frac{\partial \epsilon_\theta}{\partial \mathbf{x}_\lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) \\ &\quad + \mathcal{O}(h^2) \end{aligned} \quad (27)$$

Substituting  $k_2$  into Eqn. (25), we obtain:

$$\begin{aligned} \mathbf{x}_{\lambda_t+h} &= \mathbf{x}_{\lambda_t} + h \left[ \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{h}{2} \frac{\partial \epsilon_\theta}{\partial \lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \right. \\ &\quad \left. + \frac{h}{2} \frac{\partial \epsilon_\theta}{\partial \mathbf{x}_\lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \mathcal{O}(h^2) \right] \\ &= \mathbf{x}_{\lambda_t} + h \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{h^2}{2} \left[ \frac{\partial \epsilon_\theta}{\partial \lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \right. \\ &\quad \left. + \frac{\partial \epsilon_\theta}{\partial \mathbf{x}_\lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) \right] + \mathcal{O}(h^3) \end{aligned} \quad (28)$$

Thus, this is equivalent to the second-order Taylor expansion of  $\mathbf{x}(\lambda + h)$  at  $\lambda = \lambda_t$ :

$$\begin{aligned} \mathbf{x}(\lambda_t + h) &= \mathbf{x}(\lambda_t) + h \mathbf{x}'(\lambda_t) + \frac{h^2}{2} \mathbf{x}''(\lambda_t) + \mathcal{O}(h^3) \\ &= \mathbf{x}(\lambda_t) + h \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) + \frac{h^2}{2} \left[ \frac{\partial \epsilon_\theta}{\partial \lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \right. \\ &\quad \left. + \frac{\partial \epsilon_\theta}{\partial \mathbf{x}_\lambda}(\mathbf{x}_{\lambda_t}, \lambda_t) \epsilon_\theta(\mathbf{x}_{\lambda_t}, \lambda_t) \right] + \mathcal{O}(h^3) \end{aligned} \quad (29)$$

Moreover, from Eqn. (27), it can be observed that the evaluation  $\epsilon_\theta(\mathbf{x}_{\lambda_t} + bhk_1, \lambda_t + ah)$  at the midpoint  $(\mathbf{x}_{\lambda_t} + bhk_1, \lambda_t + ah)$  contributes derivative information  $\epsilon_\theta^{(1)}(\mathbf{x}_{\lambda_t}, \lambda_t)$  to the second-order Taylor expansion in Eqn. (29).

#### A.2. Quantization Error Analysis in Fast Sampling of Quantized Diffusion Models

To compute the numerical integration over the interval  $(\lambda_s, \lambda_t)$  corresponding to Eqn. (23), the sampler approx-

imates the sampling direction of the continuous equation by solving the higher-order expansion of  $\epsilon_\theta(\mathbf{x}_\lambda, \lambda)$  at  $(\mathbf{x}_{\lambda_s}, \lambda_s)$ :

$$\mathbf{x}_t = \frac{\alpha_t}{\alpha_s} \mathbf{x}_s - \alpha_t \sum_{n=0}^{k-1} \epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda + \mathcal{O}((\lambda_t - \lambda_s)^{k+1}), \quad (30)$$

where  $\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) = \frac{d^n \epsilon_\theta(\mathbf{x}_\lambda, \lambda)}{d\lambda^n}$  is the  $n$ -th order total derivative of w.r.t.  $\lambda$ . However, the quantized model  $\hat{\epsilon}_\theta$  introduces quantization errors  $\Delta\epsilon_\theta$ , transforming the integral into:

$$\mathbf{x}_t = \frac{\alpha_t}{\alpha_s} \mathbf{x}_s - \alpha_t \sum_{n=0}^{k-1} \epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda + \sum_{n=0}^{k-1} \Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda + \mathcal{O}((\lambda_t - \lambda_s)^{k+1}) \quad (31)$$

Next, 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,  $\Delta_{disc}$  as the discretization truncation error, and proceed to analyze the upper bound of the quantization cumulative error.

**Analysis.** First, we compute the integral:

$$I = \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \cdot \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \quad (32)$$

Define  $u = \lambda - \lambda_s$ , which implies  $\lambda = u + \lambda_s, d\lambda = du$ . Substituting these into Eqn. (31), we obtain:

$$\begin{aligned} I &= \int_0^{\lambda_t - \lambda_s} e^{-(u + \lambda_s)} \cdot \frac{u^n}{n!} du \\ &= \frac{e^{-\lambda_s}}{n!} \int_0^{\lambda_t - \lambda_s} e^{-u} \cdot u^n du \\ &= \frac{e^{-\lambda_s}}{n!} \cdot \gamma(n+1, \lambda_t - \lambda_s), \end{aligned} \quad (33)$$

where  $\gamma(\cdot, \cdot)$  denotes the lower incomplete Gamma function. According to **Assumption #2**, the step size  $h$  is small, and  $s < t$ , thus  $(\lambda_t - \lambda_s) \rightarrow 0^+$ . Under this condition,  $\gamma(\cdot, \cdot)$  can be approximated as:

$$\gamma(n+1, \lambda_t - \lambda_s) \approx \frac{(\lambda_t - \lambda_s)^{n+1}}{n+1}, \quad (34)$$

thus, the integral  $I$  simplifies to:

$$I = \frac{e^{-\lambda_s} \cdot (\lambda_t - \lambda_s)^{n+1}}{(n+1)!} \quad (35)$$

Considering the convergence of the quantization algorithm, we assume that the quantization error is bounded:

$$|\Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s)| \leq \delta_n \quad (36)$$

$$\delta = \max_{i \in \mathcal{I}} \delta_i, \quad \mathcal{I} = \{1, 2, \dots, n\} \quad (37)$$

thus, according to the triangle inequality, we have:

$$\begin{aligned} |\Delta_{quant}| &= \left| \sum_{n=0}^{k-1} \Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s) \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \right| \\ &\leq \sum_{n=0}^{k-1} |\Delta\epsilon_\theta^{(n)}(\mathbf{x}_{\lambda_s}, \lambda_s)| \cdot \left| \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \frac{(\lambda - \lambda_s)^n}{n!} d\lambda \right| \\ &\leq \sum_{n=0}^{k-1} \delta \cdot e^{-\lambda_s} \cdot \frac{(\lambda_t - \lambda_s)^{n+1}}{(n+1)!} \end{aligned} \quad (38)$$

Define the  $n$ -th order derivative error  $a_n = \delta \cdot e^{-\lambda_s} \cdot \frac{(\lambda_t - \lambda_s)^{n+1}}{(n+1)!}$ , then, the cumulative quantization error satisfies  $|\Delta_{quant}| \leq \sum_{n=0}^{k-1} a_n$ , where the ratio of successive terms is given by:

$$\begin{aligned} \frac{a_n}{a_{n-1}} &= \frac{\delta \cdot e^{-\lambda_s} \cdot \frac{(\lambda_t - \lambda_s)^{n+1}}{(n+1)!}}{\delta \cdot e^{-\lambda_s} \cdot \frac{(\lambda_t - \lambda_s)^n}{n!}} \\ &= \frac{\lambda_t - \lambda_s}{n+1} \end{aligned} \quad (39)$$

According to the previous assumption that  $\lambda_t - \lambda_s \ll 1$ , it follows that  $a_n \ll a_{n-1}$ , indicating that  $a_n$  decreases rapidly as the order  $n$  increases. Consequently, the error upper bound is estimated as:

$$\mathcal{L}_{\Delta_{quant}} = \mathcal{O}(\delta \cdot e^{-\lambda_s} (\lambda_t - \lambda_s)) \quad (40)$$

$$\begin{aligned} \mathcal{L}_\Delta &= \mathcal{L}_{\Delta_{quant}} + \mathcal{L}_{\Delta_{disc}} \\ &= \mathcal{O}(\delta \cdot e^{-\lambda_s} \cdot (\lambda_t - \lambda_s)) + \mathcal{O}((\lambda_t - \lambda_s)^{k+1}) \end{aligned} \quad (41)$$

## B. Related Work on Sampling Acceleration

Advanced accelerated sampling algorithms approximate the continuous sampling equations using high-precision numerical integration methods, minimizing truncation errors introduced by discretization. This enables larger sampling step sizes, thus reducing the number of required sampling steps while maintaining accuracy. DDIM [36] achieves non-Markovian skip-step sampling by aligning marginal probability distributions, essentially leveraging a first-order Euler discretization to approximate the solution of the neural ODE. DPM-solver [25, 26] performs a high-order expansion of the noise estimation network at discrete steps to approximate the sampling direction of the corresponding analytical integral. AMED-Solver [45] utilizes an embedded network to estimate the direction and step size of the subsequent step, incurring a minor increase in computational overhead during inference. PNMD [22] introduces a pseudo-numerical solving approach, further enhancing the accuracy of traditional numerical solvers. Complementing these trajectory-level acceleration strategies, parallel

and sparse decoding techniques offer orthogonal efficiency gains: dParallel [5] unlocks the inherent parallelism of diffusion language models for fast sampling, while SparseD [40] accelerates inference by introducing sparse attention.

## C. Experimental Details and Results

### C.1. Quantization Settings.

To comprehensively evaluate the proposed sampling-aware quantization framework, we conduct experiments under three quantization configurations:  $W8A8$ ,  $W4A8$ , and  $W4A4$ . For the  $W8A8$  setting, we assess the performance of the proposed SA-PTQ, while for  $W4A8$  and  $W4A4$  configurations, we employ SA-QLoRA for evaluation. Consistent with prior work, the first and last layers are quantized to 8 bits, with all other layers quantized to the target bit-width. Regarding data calibration, SA-PTQ utilizes the proposed dual-order trajectory sampling to gather the calibration dataset, whereas SA-QLoRA first collects the full-precision first-order trajectory to initialize the quantization parameters.

### C.2. SA-QLoRA Finetuning Details

In SA-QLoRA fine-tuning, we set the batch size to 4, the adapter rank to 32, and the number of training epochs to 160. To further enhance the alignment of sparse-step sampling trajectories, we design a mixstep progressive LoRA strategy. The basic QLoRA strategy fixes the sampling steps and aligns the full-precision and quantized outputs at each step of the sampler. In contrast, the mixstep progressive LoRA strategy iterates over a list of sampling steps set to  $\text{steps} = [100, 50, 20]$ . For each cycle, the sampler updates to the current  $\text{steps}[i]$  value, and the alignment is performed at each sampling step between  $\epsilon_{\theta}(\mathbf{x}_{t_i}, t_i)$  and  $\hat{\epsilon}_{\theta}(\mathbf{x}_{s_i}, s_i)$ , where  $(\mathbf{x}_{t_i}, t_i)$  denotes first-order sampling step and  $(\mathbf{x}_{s_i}, s_i)$  denotes intermediate step in second-order sampling.

### C.3. Unconditional Image Generation on LSUN-Churches $256 \times 256$

Table 5. Performance comparisons of unconditional image generation on LSUN-Church  $256 \times 256$  using the LDM-8 model.

Method	W/A	FID ↓	sFID ↓	Prec. ↑	Rec. ↑
FP	32/32	7.26	13.75	61.50%	50.72%
PTQD	8/8	11.87	12.97	56.57%	54.15%
SA-PTQ	8/8	<b>10.65</b>	<b>12.51</b>	<b>56.86%</b>	<b>54.54%</b>
PTQD	4/8	12.96	17.81	50.23%	52.80%
EfficientDM	4/8	11.86	15.64	52.27%	<b>53.78%</b>
SA-QLoRA	4/8	<b>10.07</b>	<b>15.11</b>	<b>54.15%</b>	53.68%
PTQD	4/4	-	-	-	-
EfficientDM	4/4	23.42	20.15	46.02%	<b>45.63%</b>
SA-QLoRA	4/4	<b>17.89</b>	<b>19.04</b>	<b>48.95%</b>	43.07%



## D. Additional Visual Results

### D.1. Visualization of Multi-Order Trajectories

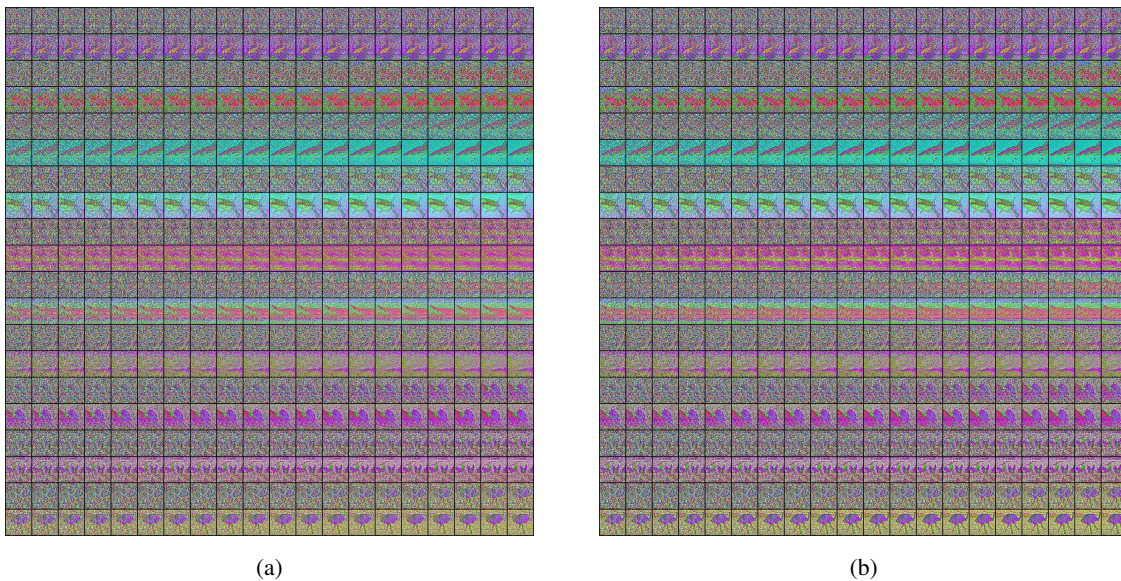


Figure 5. Latent space feature trajectories of LDM4 under 20-step sampling on the ImageNet  $256 \times 256$  dataset. (a) Feature trajectories sampled using DPM-Solver-1. (b) Intermediate-step feature trajectories sampled using DPM-Solver-2.

### D.2. Visual Comparison Across Quantization Algorithms

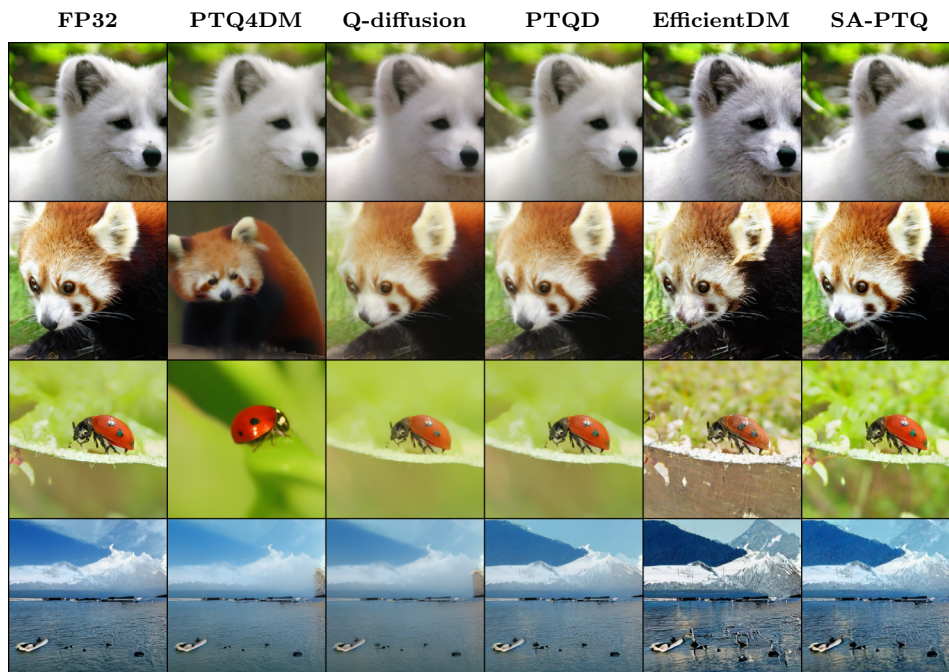


Figure 6. Comparison of generative performance on the ImageNet  $256 \times 256$  dataset with 20-step sampling among the full-precision LDM4 and its W8A8 quantized counterparts using PTQ4DM, Q-diffusion, PTQD, EfficientDM, and our proposed SA-LoRA. (Revised version of the main figure in the main text, supplemented with the names of the applied quantization algorithms.)



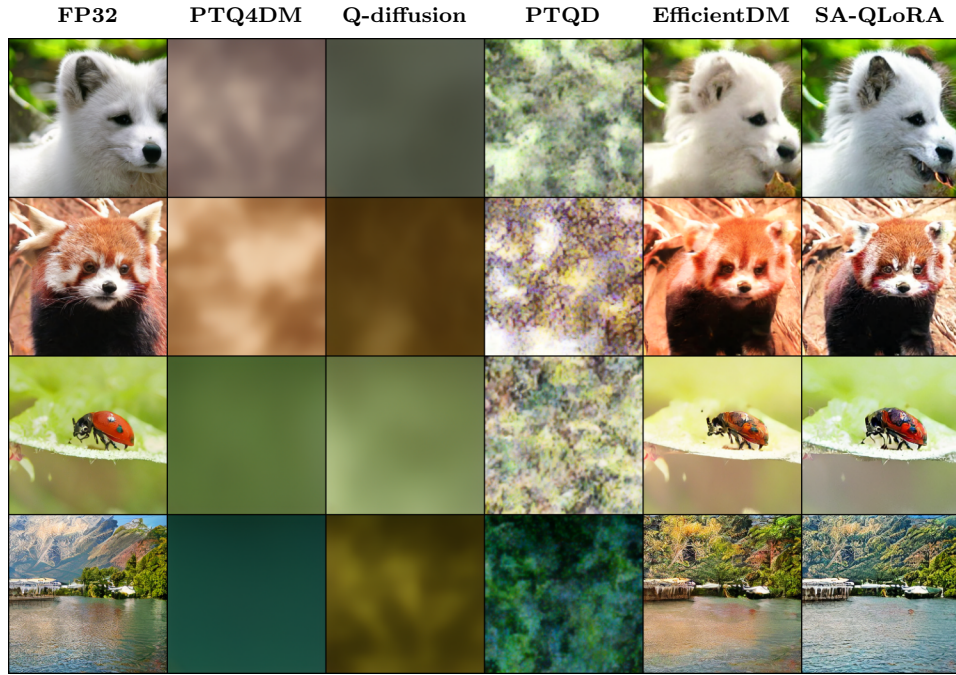
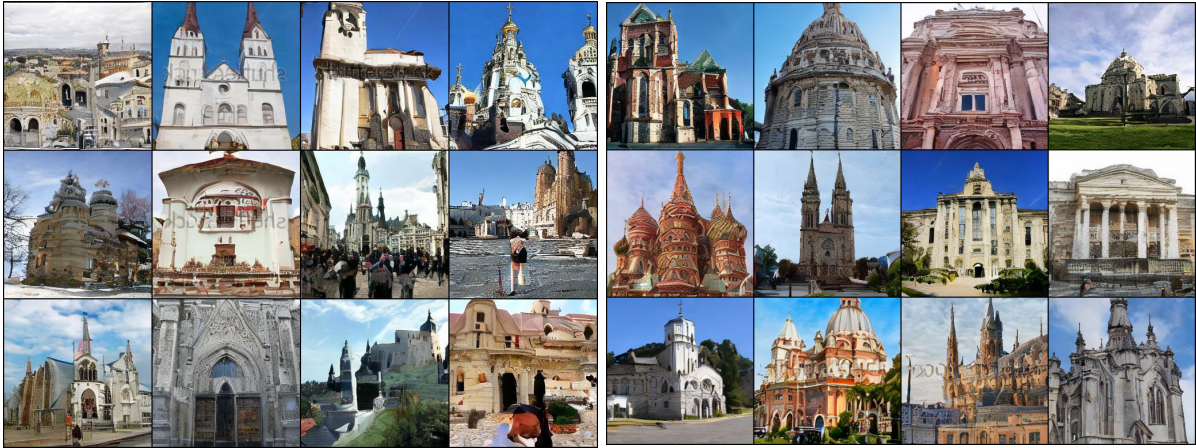


Figure 7. Comparison of generative performance on the ImageNet  $256 \times 256$  dataset with 20-step sampling among the full-precision LDM4 and its W4A4 quantized counterparts using PTQ4DM, Q-diffusion, PTQD, EfficientDM, and our proposed SA-LoRA. (Revised version of the main figure in the main text, supplemented with the names of the applied quantization algorithms.)

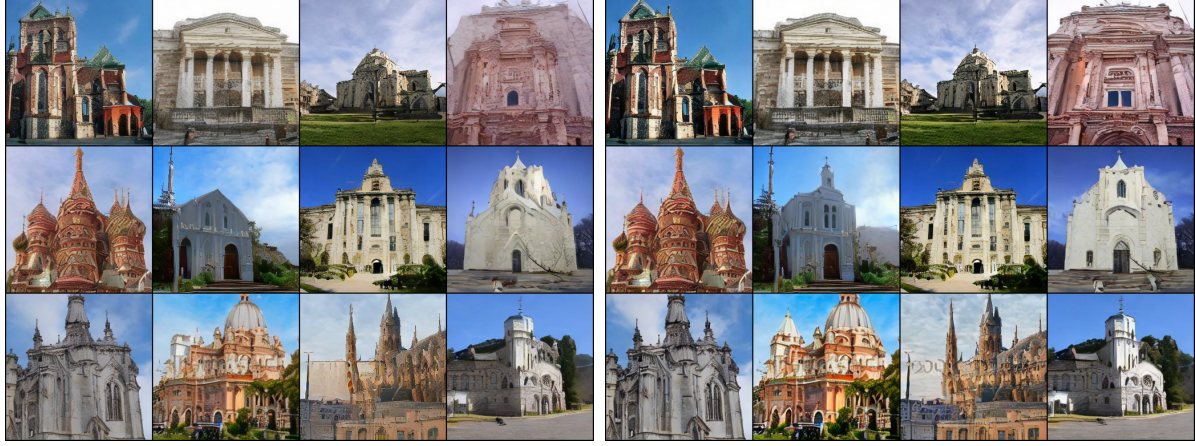


(a) FP32

(b) SA-QLoRA [W4A8]

Figure 8. Comparison of generative performance between the full-precision LDM8 and its W4A8 quantized counterpart, utilizing our proposed SA-QLoRA, on the LSUN-Church  $256 \times 256$  dataset under 50-step sampling.

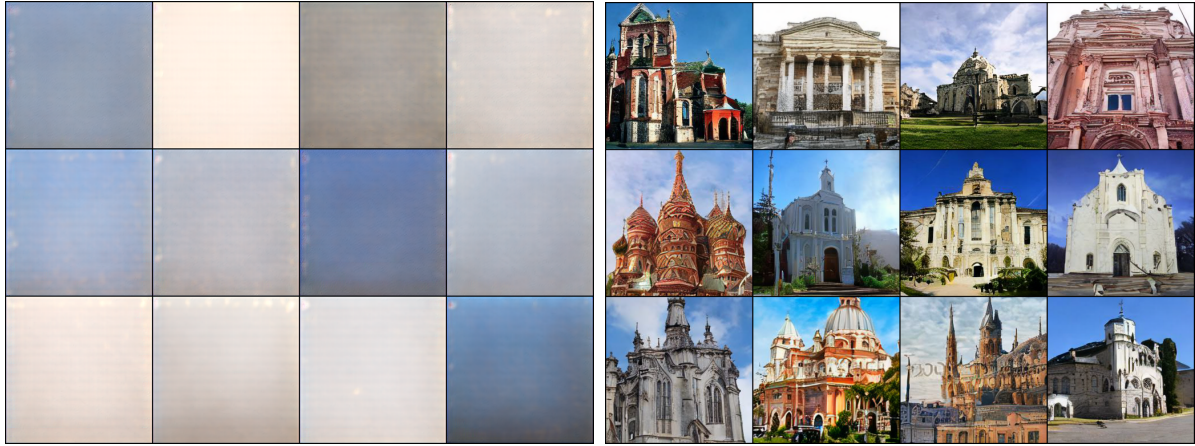




(a) PTQD [W4A8]

(b) SA-QLoRA [W4A8]

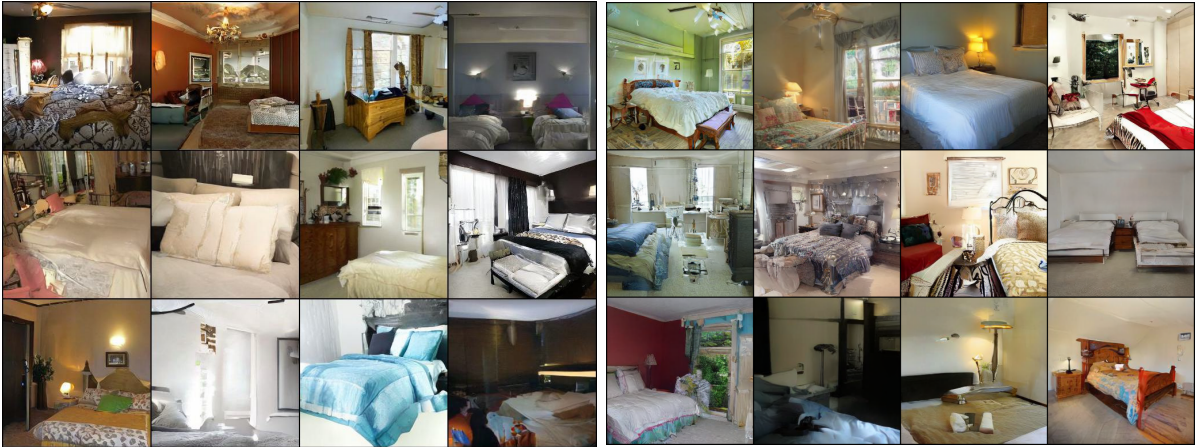
Figure 9. Generative performance comparison of W4A8 quantized LDM8 models, employing PTQD and our proposed SA-QLoRA, on the LSUN-Church  $256 \times 256$  dataset with 50-step sampling.



(a) Q-diffusion [W4A4]

(b) SA-QLoRA [W4A4]

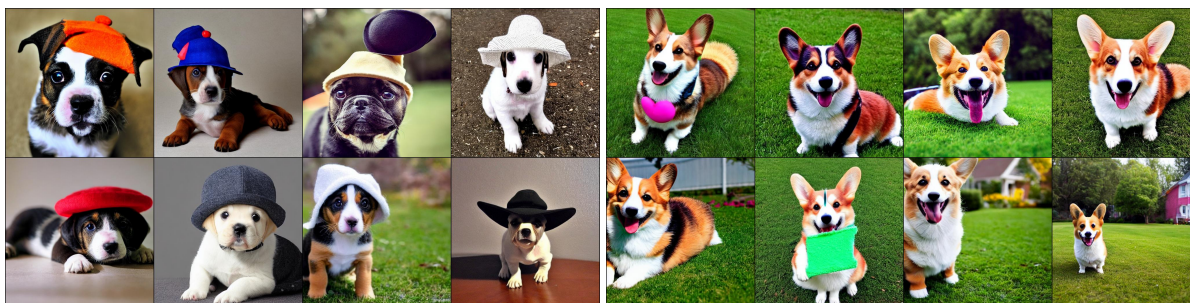
Figure 10. Generative performance comparison of W4A4 quantized LDM8 models, employing Q-diffusion and our proposed SA-QLoRA, on the LSUN-Church  $256 \times 256$  dataset with 50-step sampling.



(a) FP32

(b) SA-QLoRA [W4A8]

Figure 11. Comparison of generative performance between the full-precision LDM4 and its W4A8 quantized counterpart, utilizing our proposed SA-QLoRA, on the LSUN-Bedroom  $256 \times 256$  dataset under 50-step sampling.



(a) Prompt "a puppy wearing a hat"

(b) Prompt "A Corgi lying on a green lawn, smiling happily."

Figure 12. Generation performance of our SA-QLoRA under W8A8 quantization.