

Anonymous Authors

We organize our supplementary material as follows:

- Intra-QNCD occurs within the noise prediction process of single-step sampling. Inter-QNCD occurs at the end of single-step sampling. Here we add more details on how our QNCD incorporates the sampling process of the diffusion model from algorithm procedures:

First, we obtain the exact embeddings and store them when quantizing the noise prediction network ϵ_θ . At the same time, according to Eq. 7, we compute the static smoothing factor S for each Resblock. Lines 6-10 of the Alg. 1 show the scale separation process. Taking a single Resblock as an example, we first get the accurate emb_l from the tabular reference and add it to the features. Since embedding scales the features channel-by-channel dimensionally, it makes the distribution of the fused features uneven and difficult to quantify, which is smoothed by our smoothing factor S . We then incorporate the smoothing factor into the weights, thus maintaining the mathematical equivalence of the convolution. For the other Resblocks in the network, we do the same above.

Unpublished working draft. Not for distribution.

Input: Floating-point noise prediction network ϵ_θ .

Parameters: emb_t means the embedding at the t steps for the Resblock. W means the convolutional weight of the output layer in Resblock.

-
- Algorithm 2** Inter-QNCD in the sampling process of diffusion models.

Input: Floating-point noise prediction network ϵ_θ . **Parameters:**
Hyperparameters in sampling $\alpha_t, \beta_t, \sigma_t$. total timesteps T ,
Estimation interval n .

Output: Final synthesized samples

- 1: Collecting Calibration Set.
- 2: Converting noise prediction network ϵ_θ to quantized one $\tilde{\epsilon}_\theta$
- 3: Generate a Gaussian Noise \tilde{x}_T as initialization.
- 4: **for** $t = T$ to 1 **do**
- 5: $z \in N(0, I)$ if $t > 1$, else $z = 0$
- 6: $\tilde{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \tilde{\epsilon}_\theta(\tilde{x}_t, t) \right) + \sigma_t z$
- 7: **if** $t \% n == 0$ and $t < T - n$ **then**
- 8: Perform a **diffusion process** on \tilde{x}_{t-1} :
 $\hat{x}_t = \sqrt{\alpha_t} \tilde{x}_{t-1} + \sqrt{1 - \alpha_t} z_1, z_1 \in N(0, I)$
- 9: Perform a **denoising process** on \hat{x}_t :
 $\tilde{\epsilon}_\theta(\hat{x}_t, t) = \epsilon_\theta(\hat{x}_t, t) + q_\theta(\hat{x}_t, t) \approx z_1 + q_\theta(\hat{x}_t, t)$
- 10: $q_\theta(\hat{x}_t, t) \approx \tilde{\epsilon}_\theta(\hat{x}_t, t) - z_1$
- 11: $q_\theta(\tilde{x}_t, t) \approx q_\theta(\hat{x}_t, t)$
- 12: **else**
- 13: $q_\theta(\tilde{x}_t, t) \approx q_\theta(\tilde{x}_{t+1}, t+1)$ if $t < T - n$ **else** 0
- 14: **end if**
- 15: Feed $q_\theta(\tilde{x}_t, t)$ into Eq. (4) to remove inter noise: $\tilde{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} (\tilde{\epsilon}_\theta(\tilde{x}_t, t) - q_\theta(\tilde{x}_t, t)) \right) + \sigma_t z$
- 16: **end for**
- 17: Return final sample \tilde{x}_0

8.2 Inter quantization noise correction

Alg. 2 shows the complete sampling process, where our inter quantization noise correction module (Inter-QNCD) occurs after the end of the single-step sampling. It is worth noting that our method starts working only after the diffusion model has carried out several normal sampling processes. Also, our Inter-QNCD is stage-by-stage and is not performed at every step. First, the line 5 and line 6 in Alg. 2 are the normal sampling process, and they yield the sampling output \tilde{x}_{t-1} :

$$\begin{aligned}\tilde{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} \left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \tilde{\epsilon}_\theta(\tilde{x}_t, t) \right) + \sigma_t z, \quad z \in N(0, I) \\ &= \frac{1}{\sqrt{\alpha_t}} \left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} (\epsilon_\theta(\tilde{x}_t, t) + q_\theta(\tilde{x}_t, t)) \right) + \sigma_t z.\end{aligned}\quad (1)$$

And \hat{x}_t is obtained by adding a well-determined Gaussian noise z_1 to \tilde{x}_{t-1} , which simulates the diffusion process:

Based on \hat{x}_t , the network outputs the desired filtered noise $\tilde{\epsilon}_\theta(\hat{x}_t, t)$, which consists of two parts, first the target noise z_1 , and the quantization noise $q_\theta(\hat{x}_t, t)$.

\hat{x}_t is obtained by a single-step denoising and diffusion process on \tilde{x}_t , thus their distributions remain highly similar as well as the corresponding quantization noise:

$$\begin{aligned}\hat{x}_t &= \underbrace{\sqrt{\alpha_t} \tilde{x}_{t-1} + \sqrt{1-\alpha_t} z_1}_{\text{diffusion process}}, \quad z_1, z \in N(0, I) \\ &= \underbrace{\left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \tilde{\epsilon}_\theta(\tilde{x}_t, t) + \sqrt{\alpha_t} * \sigma_t z + \sqrt{1-\alpha_t} z_1 \right)}_{\text{denoising process}} \\ &= \tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \tilde{\epsilon}_\theta(\tilde{x}_t, t) + (\sqrt{\alpha_t} \sigma_t + \sqrt{1-\alpha_t}) z \\ &\approx \tilde{x}_t\end{aligned}\quad (2)$$

Finally, the quantization noise $q_\theta(\tilde{x}_t, t)$ can be determined, as the Gaussian noise z_1 is manually designed and $\tilde{\epsilon}_\theta(\hat{x}_t, t)$ is the output of the noise predicting network, both of which are ascertainable:

$$q_\theta(\tilde{x}_t, t) \approx q_\theta(\hat{x}_t, t) \approx \tilde{\epsilon}_\theta(\hat{x}_t, t) - z_1. \quad (3)$$

Besides, the quantization noise is obtained through estimation and doesn't align perfectly with the actual noise in terms of pixel dimension, whereas it is identical at the level of the overall distribution. Thus we get the distribution of the quantization noise at stage $t-1$ and correct the output sample in Eq. (4):

$$\begin{aligned}\tilde{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} \left(\tilde{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} (\tilde{\epsilon}_\theta(\tilde{x}_t, t) - q_\theta(\tilde{x}_t, t)) \right) \\ &\quad + \sigma_t z, \quad z \in N(0, I)\end{aligned}\quad (4)$$

9 STATISTICAL ANALYSIS

9.1 Activation Imbalance due to Embedding

As shown in Fig. 10 and Fig. 11, we visualize the data range of the eight channels of feature activation (total channels are 256). The following conclusions can be obtained:

- The addition of embedding makes the distance between the upper and lower endpoints of the activation distribution larger, which enlarges the range of features and **increases the number of outliers**.
- The impact of embedding on features varies across different channels, exhibiting a more pronounced effect on channels 0, 32, 96, and 160 while demonstrating a relatively diminished influence on channel 224. This **discrepancy** arises from the incorporation of embedding, as *embt* scales different channels to varying extents. Consequently, in order to address the non-uniform distribution introduced by embedding, it becomes imperative to tackle this issue on a per-channel basis.
- We present visualizations of the activation range at different timesteps ($t=50$ and $t=1$). It is evident that the phenomenon of uneven distribution of features, resulting from embedding, persists across all time steps. This observation aligns with the findings depicted in Fig. 1, indicating that the impact of embedding permeates throughout the entire sampling process and exhibits periodicity.
- The application of our scaling factor effectively mitigates the impact caused by embedding, thereby reducing the occurrence of outliers and promoting a **more compact data distribution**. With the incorporation of our scaling factor, features can be quantized with greater ease.

9.2 Accumulated Quantization Noise

In Fig. 1, we present a visualization of the LPIPS Distance between the quantized model output and its floating-point counterpart for all 100 time steps, demonstrating that our method consistently produces outputs that are closer in proximity.

As direct visualizing the quantization noise is challenging for extracting information (defined as the absolute difference between the output of the floating-point model and that of the quantization model), we substitute it with an illustrative representation of its distribution

As shown in Fig. 12, the first single-step output samples x_{100} contain almost no quantization noise, since x_{100} is a standard Gaussian noise. With continuous sampling, the mean value of the quantization noise increases, implying the accumulation of quantization noise.

Besides, the increasing interquartile spacing indicates a continuous increase of large quantization noise in the pixel dimension. To address this problem, we discern the distribution of quantization noise via a reversal strategy, enabling its exclusion in subsequent sampling steps. As shown in Fig. 12, the mean value of the quantization noise of our method is much lower than that of the original quantization method, as well as the range of quantization noise.

Finally, for the final synthesized samples, the mean value of quantization noise is reduced from 0.089 to 0.034, which is consistent with the increase in output cosine similarity from 93.4% to 95.1% in Fig. 8.

In the Fig 13, we visualize the changes in output during the sampling process. It can be seen that there is a substantial gap between the synthesized images of the quantized model and the output of the full-precision model, and this gap magnifies as the

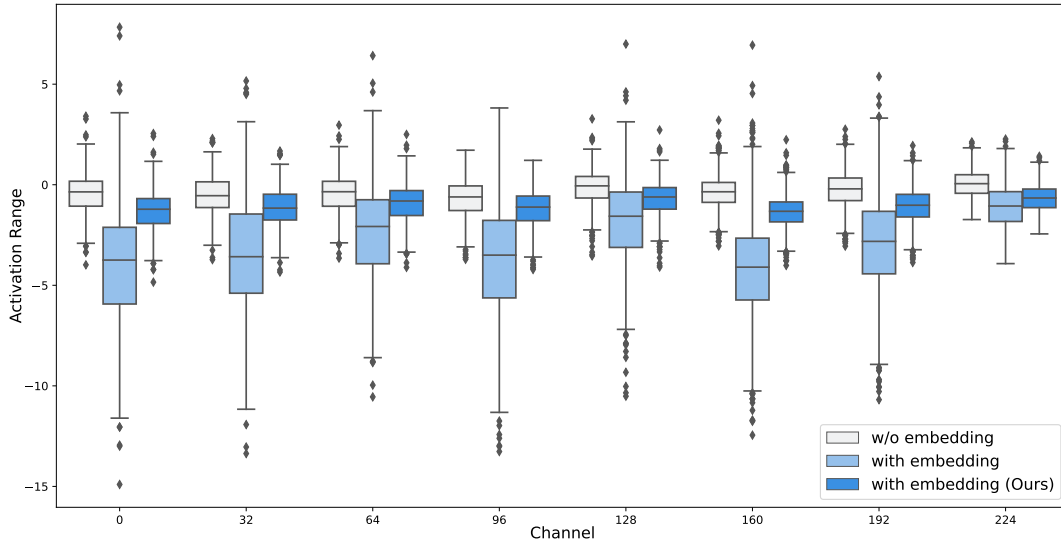


Figure 10: Activation range of feature h_t in Resblock (DDIM model with $t = 50$ on CIFAR). The feature distribution, when combined with the embedding, exhibits pronounced irregularities, which can be efficiently smoothed using our scale separation procedure.

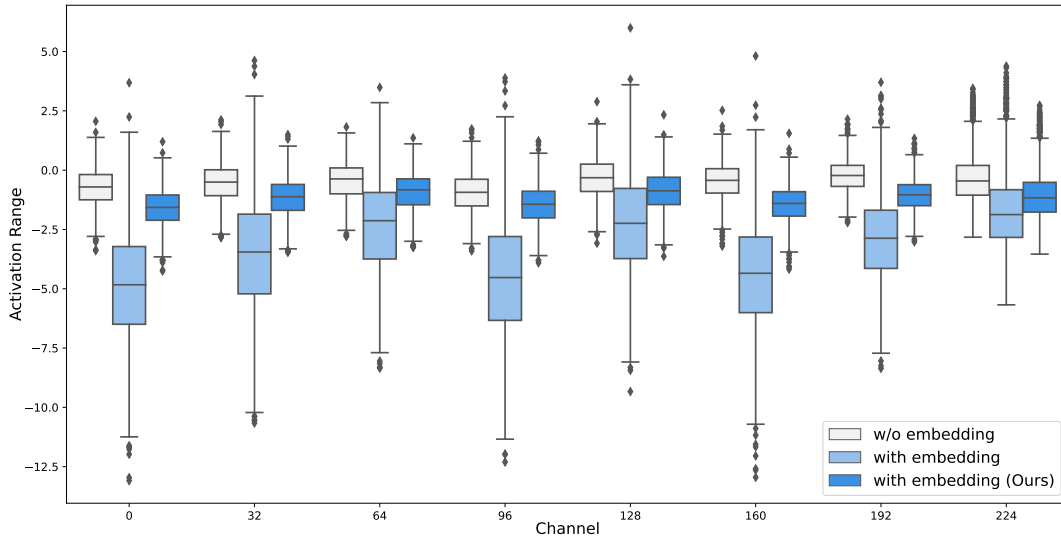


Figure 11: Activation range of feature h_t in Resblock (DDIM model with $t=1$ on CIFAR).

sampling iterations continue. At the same time, our QNCD can effectively correct the quantization noise, making the sampling direction of the synthesized image closer to that of the full-precision model, which demonstrates the effectiveness of our method.

10 ADDITIONAL VISUALIZATION RESULTS

In the body paper, we visualized the results of Stable Diffusion on MS-COCO (512x512) under W4A6 configuration. In this section, we visualize the results on more datasets with more diffusion models. The order of the visualized results is as follows: W4A6 (Fig. 14) and

W4A8 (Fig. 15) for ImageNet, W4A8 (Fig. 16) for MS-COCO, W8A8 for CIFAR (Fig. 17), W8A8 for LSUN (Fig. 18) .

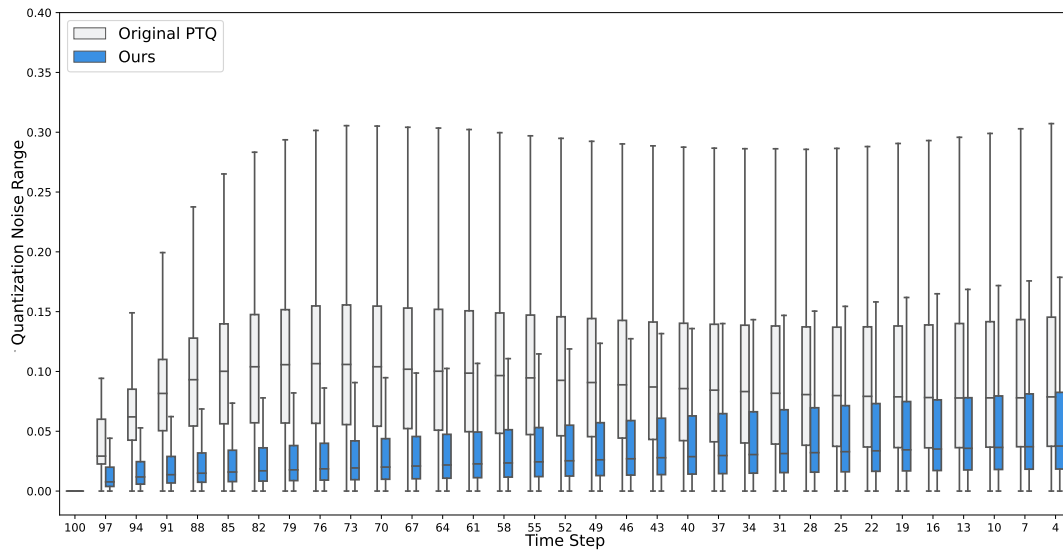


Figure 12: Quantization noise for all 100 time steps (DDIM model on CIFAR). With iterative sampling, quantization noise accumulates, which can be mitigated by our quantization noise estimation module. (W8A8)

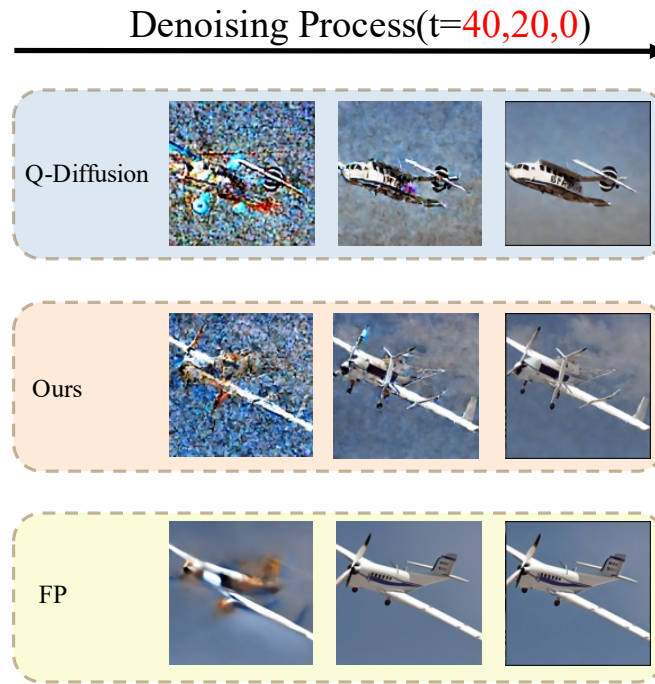
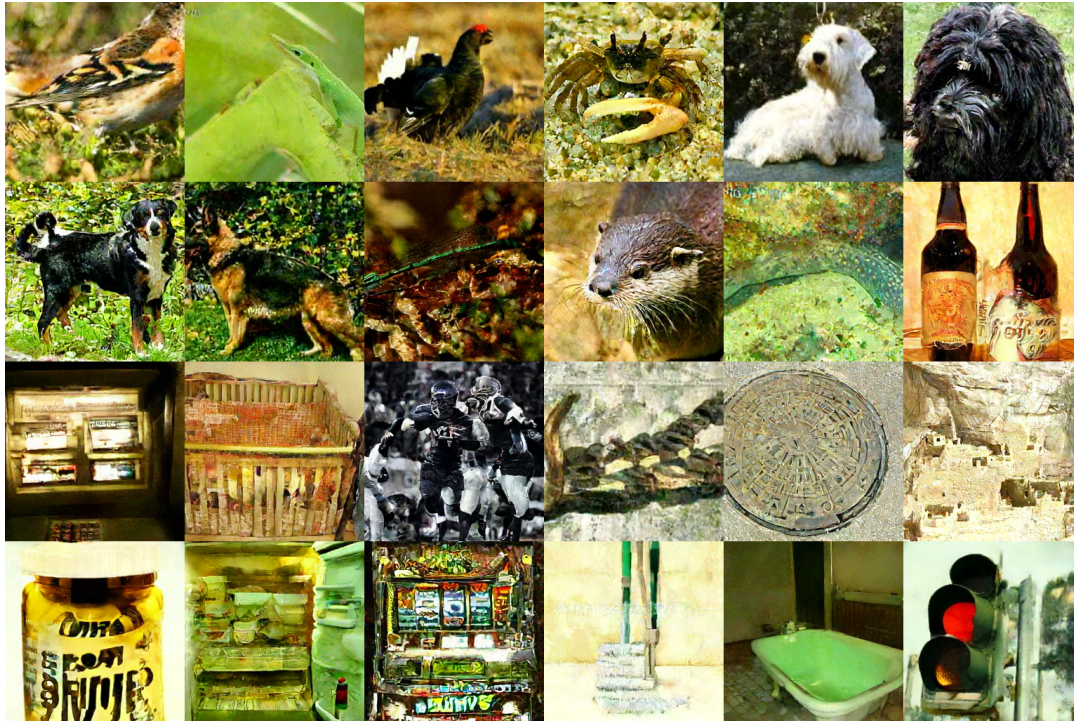


Figure 13: The output images during the denoising process. We conduct experiments on MS-COCO (512×512) using Stable Diffusion (Step=50) in W8A8.

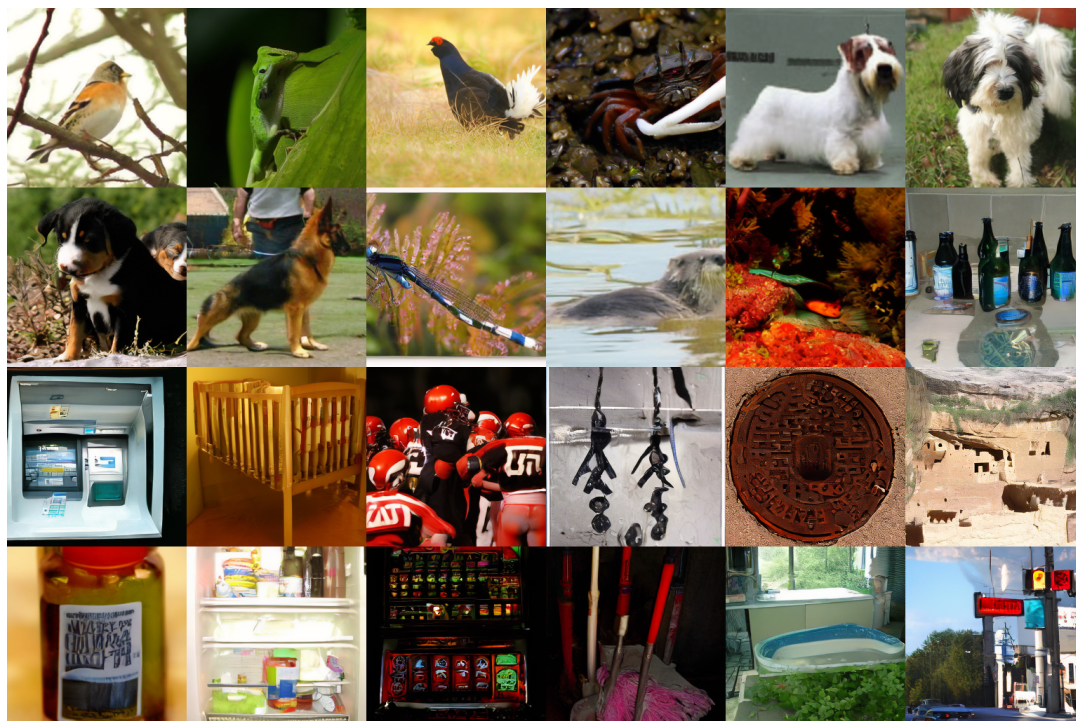


(a) Sample generated with Q-Diffusion (W4A6)

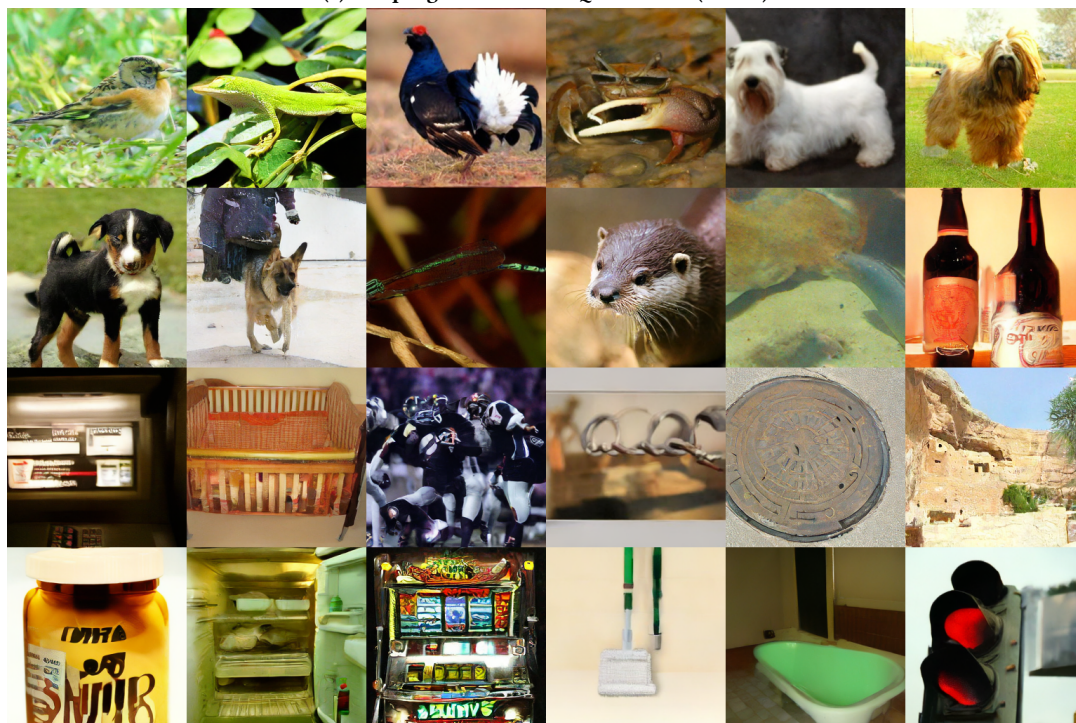


(b) Sample generated with our QNCD (W4A6)

Figure 14: Class-conditional generation on ImageNet 256×256 by LDM-4 (steps=20). The bitwidth for Q-Diffusion and our method is W4A6. The quality of our synthesized samples is much higher than that of Q-Diffusion. Corresponding FID is decreased from 43.00 to 23.24.



(a) Sample generated with Q-Diffusion (W4A8)



(b) Sample generated with our QNCD (W4A8)

Figure 15: Class-conditional generation on ImageNet 256×256 by LDM-4 (steps=20). The bitwidth for Q-Diffusion and our method is W4A8. Our method can generate high-fidelity images in only 20 steps.



(a) Sample generated with Q-Diffusion (W4A8)

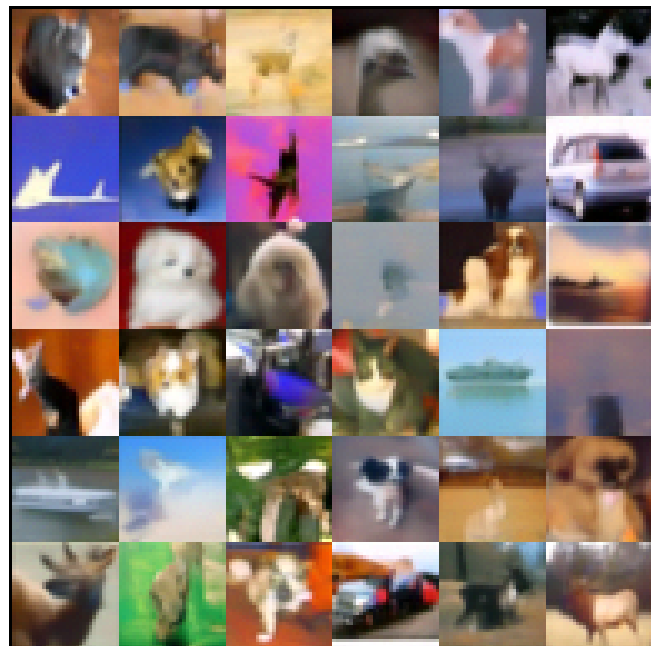


(b) Sample generated with our QNCD (W4A8)

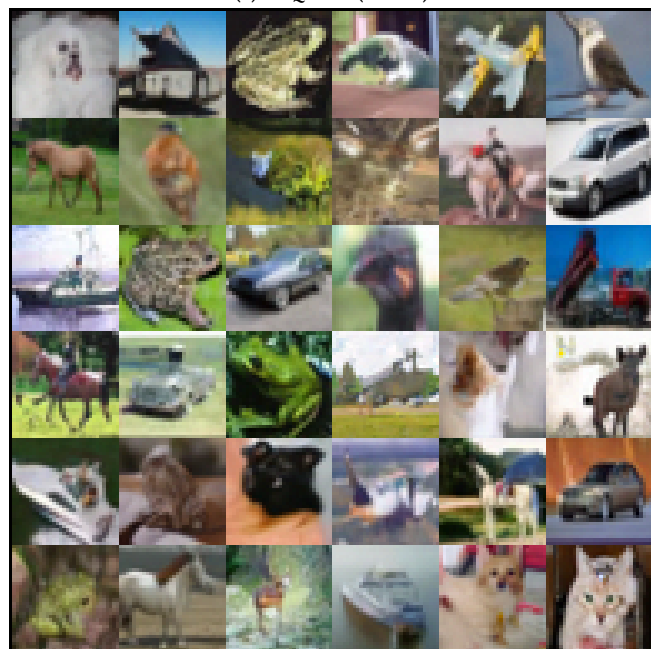


(c) Sample generated with Full Precision(FP)

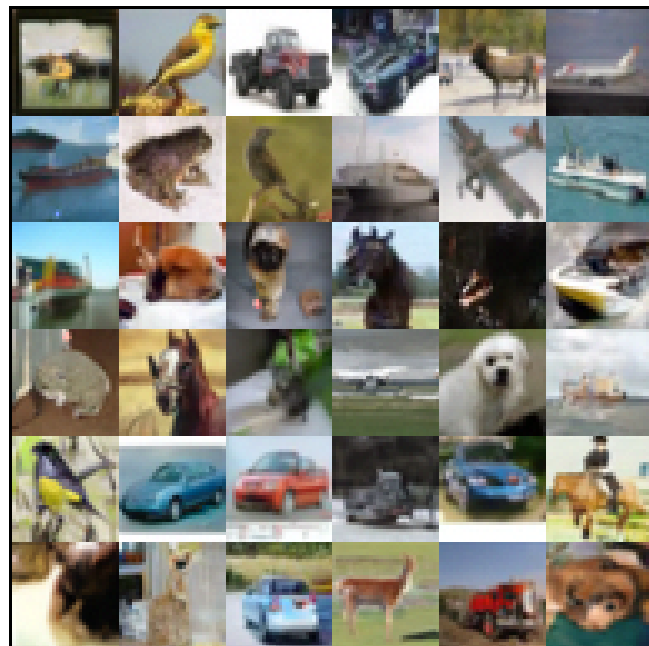
Figure 16: Text-guided image generation on MS-COCO 512×512 by Stable Diffusion (steps=50). The bitwidth for Q-Diffusion and our method is W4A8. Both two method can generate high-fidelity images, while the details of images synthesized by our QNCD are more rational.



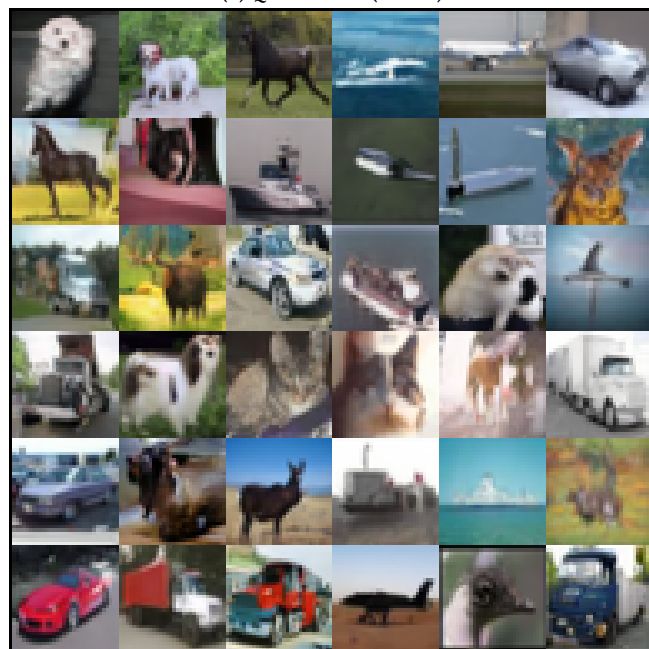
(a) PTQ4DM (W8A8)



(c) our QNCD (W8A8)

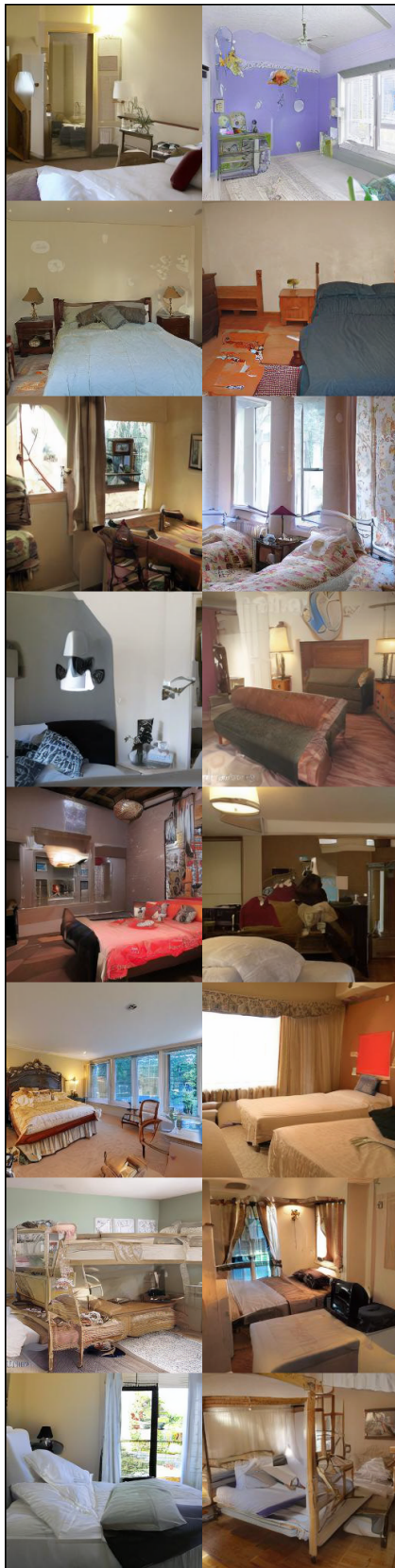


(b) Q-Diffusion (W8A8)

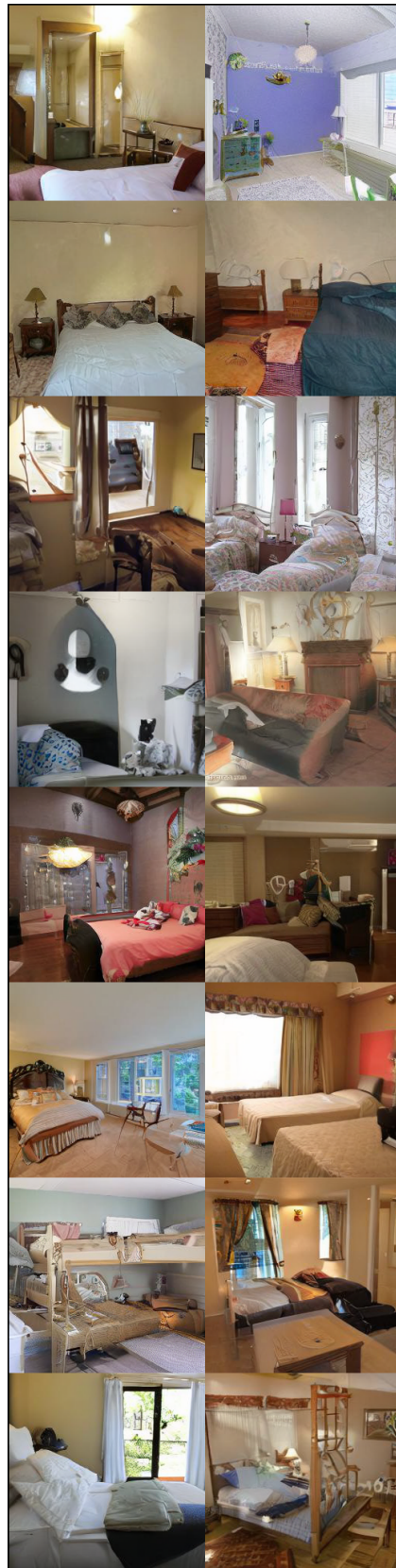


(d) FP (W32A32)

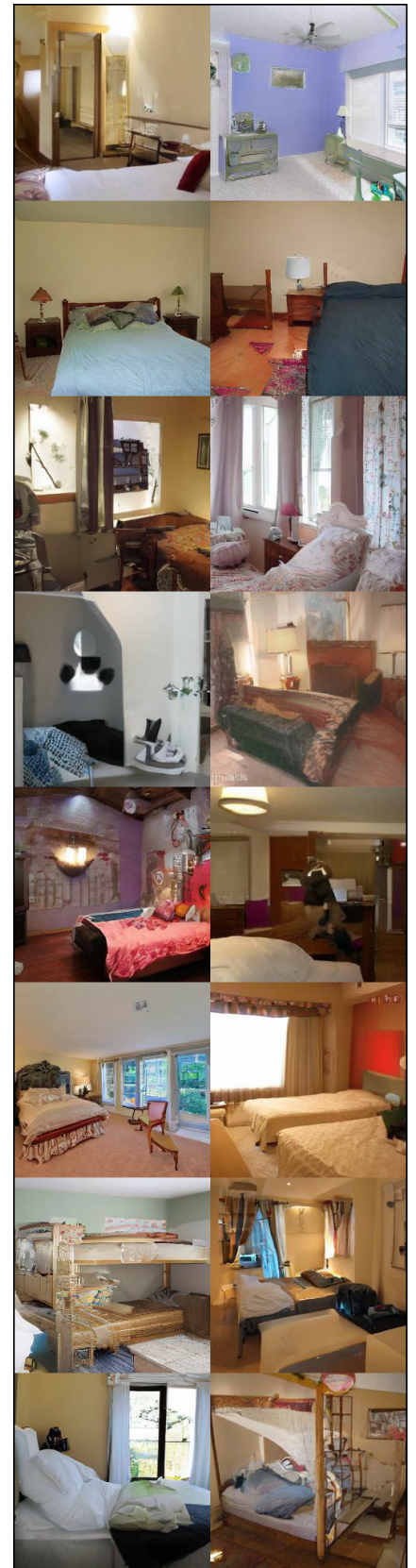
Figure 17: Unconditional generation on CIFAR 32×32 by DDIM (steps=100).



(a) Q-Diffusion (W8A8)



(b) our QNCD (W8A8)



(c) FP

Figure 18: Unconditional generation on LSUN-Bedrooms 256×256 by LDM-4 (steps=200). Compared with Q-Diffusion, samples generated by our QNCD are less affected by quantization noise and exhibit a closer resemblance to the results of the floating-point model.