TFMQ-DM: Temporal Feature Maintenance Quantization for Diffusion Models

Yushi Huang[1,2]*, Ruihao Gong[1,2]*, Jing Liu[2,3], Tianlong Chen[4,5,6], Xianglong Liu[1]†
[1]Beihang University [2]SenseTime Research [3]Monash University [4]MIT
[5]Harvard University [6]The University of North Carolina at Chapel Hill
{huangyushi,gongruihao,liujing5}@sensetime.com,tianlong@mit.edu,xlliu@buaa.edu.cn

Abstract

The Diffusion model, a prevalent framework for image generation, encounters significant challenges in terms of broad applicability due to its extended inference times and substantial memory requirements. Efficient Post-training Quantization (PTQ) is pivotal for addressing these issues in traditional models. Different from traditional models, diffusion models heavily depend on the time-step t *to achieve satisfactory multi-round denoising. Usually,* t *from the finite set* $\{1, \ldots, T\}$ *is encoded to a temporal feature by a few modules totally irrespective of the sampling data. However, existing PTQ methods do not optimize these modules separately. They adopt inappropriate reconstruction targets and complex calibration methods, resulting in a severe disturbance of the temporal feature and denoising trajectory, as well as a low compression efficiency. To solve these, we propose a Temporal Feature Maintenance Quantization (TFMQ) framework building upon a Temporal Information Block which is just related to the time-step* t *and unrelated to the sampling data. Powered by the pioneering block design, we devise temporal information aware reconstruction (TIAR) and finite set calibration (FSC) to align the full-precision temporal features in a limited time. Equipped with the framework, we can maintain the most temporal information and ensure the end-to-end generation quality. Extensive experiments on various datasets and diffusion models prove our state-of-the-art results. Remarkably, our quantization approach, for the first time, achieves model performance nearly on par with the full-precision model under 4-bit weight quantization. Additionally, our method incurs almost no extra computational cost and accelerates quantization time by* $2.0\times$ *on LSUN-Bedrooms* 256×256 *compared to previous works.*

1. Introduction

Generative modeling plays a crucial role in machine learning, particularly in applications like image [13, 14, 17, 39, 45], voice [38, 43], and text synthesis [2, 52]. Diffusion models have showcased impressive capabilities in producing high-quality samples across diverse domains. In comparison to generative adversarial networks (GANs) [8] and variational autoencoders (VAEs) [20], diffusion models successfully sidestep issues such as model collapse and posterior collapse, resulting in a more stable training process. However, the substantial computational cost poses a critical bottleneck hampering the widespread adoption of diffusion models. Furthermore, the computational cost for diffusion models can be attributed to two primary factors. First, these models typically require hundreds of denoising steps to generate images, rendering the procedure considerably slower than that of GANs. Prior efforts [21, 27, 29, 45] have addressed this challenge by seeking shorter and more efficient sampling trajectories, thereby reducing the number of denoising steps. Second, the substantial network architecture of diffusion models demands considerable time and memory resources, particularly for foundational models pre-trained on large-scale datasets, e.g., LDM [39] and Stable Diffusion. Our work aims to tackle the latter challenge, focusing on the compression of diffusion models.

Quantization is currently the most widely used method [1, 5, 7, 32, 33] for compressing models by mapping high-precision floating-point numbers to low-precision numbers. Among different quantization methods, Post-training quantization (PTQ) [15, 32, 48] incurs lower overhead and is more user-friendly without the need for retraining or fine-tuning. While PTQ on conventional models has undergone extensive study [7, 24, 32, 48], its application to diffusion models has shown huge performance degradation, especially under low-bit settings. For instance, Q-Diffusion [23] exhibits severe accuracy drop on some datasets [49] under 4-bit quantization. We believe the reason they fail to achieve better results is that they all overlook the sampling data independence and uniqueness of temporal features, which are generated from time-step t through a few modules, used to control the denoising trajectory in diffusion models. Therefore, we observe that temporal feature disturbance significantly impacts model performance in the

*Equal Contribution
†Corresponding Author

aforementioned methods.

To tackle temporal feature disturbance, we first find that the modules generating temporal features are independent of the sampling data and define the whole modules as the **Temporal Information Block**. All existing methods do not separately optimize this block during the quantization process, causing temporal features to overfit to limited calibration data. On the other hand, since the maximum time-step for denoising is a finite positive integer, the temporal feature and the activations during its generation form a finite set. The optimal approach is to optimize each element in this set individually. Based on these observations and analyses, we propose a novel quantization reconstruction approach, **temporal information aware reconstruction (TIAR)**, specifically optimizing for temporal features. It aims to reduce temporal feature loss as the optimization objective while isolating the network's components related to sampled data from the generation of temporal features during calibration. Furthermore, we also introduce a calibration strategy, **finite set calibration (FSC)**, for the finite set of temporal features and activations during its generation. This strategy employs different quantization parameters for activations corresponding to different time-steps. Moreover, the calibration speed of this method is faster than existing mainstream methods [1, 5], for example, we speedup quantization time by 2.0× on LSUN-Bedrooms 256×256 dataset, yet the strategy incurs negligible inference and storage overhead. The overview of our framework can be seen in Fig. 1. In summary, our contributions are as follows:

- We discover that existing quantization methods suffer from temporal feature disturbance, disrupting the denoising trajectory of diffusion models and significantly affecting the quality of generated images.
- We reveal that the disturbance comes from two aspects: inappropriate reconstruction target and unaware of finite activations. Both inducements ignore the special characteristics of time information-related modules.
- A framework of temporal feature maintenance quantization (TFMQ) is proposed, consisting of temporal information aware reconstruction (TIAR) for weight quantization and finite set calibration (FSC) for activation quantization. Both are based on a Temporal Information Block specially devised for diffusion models.
- Extensive experiments on various datasets show that our novel framework achieves a new state-of-the-art result in PTQ of diffusion models, especially under 4-bit weight quantization, and significantly accelerates quantization time. For some hard tasks, e.g., CelebA-HQ 256×256, our method reduces the FID score by 6.71 (images in appendix).

2. Related Work

2.1. Efficient Diffusion Models

There are diverse perspectives to accelerate the inference of diffusion model, e.g., retraining-based model design [3, 6, 31, 53] and retraining-free sampler strategy [21, 45, 51]. However, the retraining-based method proves resource-intensive and time-consuming. The efficient samplers can reduce sampling iterations but still suffer from diffusion models' extensive parameters and computational complexity. In this paper, we focus on diminishing the time and memory overhead of the single-step denoising process using low-bit quantization in a training-free manner, a method orthogonal to previous speedup techniques.

2.2. Model Quantization

Quantization is a predominant technique for minimizing storage and computational costs. It can be categorized into quantization-aware training (QAT) [7, 16, 28, 50, 54] and post-training quantization (PTQ) [15, 24, 26, 32, 48]. QAT requires intensive model training with substantial data and computational demands. Correspondingly, PTQ compresses models without re-training, making it a preferred method due to its minimal data requirements and easy deployment on real hardware. In PTQ, high-precision values are mapped into discrete levels using uniform quantization expressed as:

$$\hat{x} = \Phi(\lfloor \frac{x}{s} \rceil + z, 0, 2^b - 1), \tag{1}$$

where x represents floating-point data, $\hat{x}$ is the quantized value, s is the quantization step size, z is the zero offset, and b is the target bit-width. The clamp function $\Phi(\cdot)$ clips the rounded value $\lfloor \frac{x}{s} \rceil + z$ within the range of $[0, 2^b - 1]$. However, naive quantization may lead to accuracy degradation, especially for low-bit quantization. Recent studies [15, 24, 32, 47, 48] have explored innovative strategies based on reconstruction to preserve model performance after low-bit quantization.

In contrast, the iterative denoising process in diffusion models presents new challenges for PTQ in comparison to traditional models. PTQ4DM [42] represents the initial attempt to quantize diffusion models to 8-bit, albeit with limited experiments and lower resolutions. Conversely, Q-Diffusion [23] achieves enhanced performance and is evaluated on a broader dataset range. Moreover, PTQD [10] eliminates quantization noise through correlated and residual noise correction. Notably, traditional single-time-step PTQ calibration methods are unsuitable for diffusion models due to significant activation distribution changes with each time-step [23, 42, 44, 46]. ADP-DM [46] proposes group-wise quantization across time-steps for diffusion models, and TDQ [44] introduces distinct quantization parameters for different time-steps. However, all of the above works overlook the specificity of temporal features. To address temporal feature disturbance in the aforementioned works, our study delves into the inducements of the phenomenon and introduces a novel reconstruction and calibration framework, significantly enhancing quantized diffusion model performance.

3. Preliminaries

Diffusion models. Diffusion models [14, 45] iteratively add Gaussian noise with a variance schedule $\beta_1, \ldots, \beta_T \in (0, 1)$ to data $\mathbf{x}_0 \sim q(\mathbf{x})$ for T times as sampling process, resulting in a sequence of noisy samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$. In DDPMs [14], the former sampling process is a Markov chain, taking the form:

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \end{aligned} \tag{2}$$

where $\alpha_t = 1 - \beta_t$. Conversely, the denoising process removes noise from a sample from Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to gradually generate high-fidelity images. Nevertheless, due to the unavailability of the true reverse conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, diffusion models approximate it via variational inference by learning a Gaussian distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$, the $\boldsymbol{\mu}_\theta$ can be derived by reparameterization trick as follows:

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

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\boldsymbol{\epsilon}_\theta(\cdot)$ is a noise estimation model. The variance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ can be either learned [34] or fixed to a constant schedule [14] σ_t. When employing the latter method, $\mathbf{x}_{t-1}$ can be expressed as:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z}, \tag{4}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Reconstruction on diffusion models. UNet [40], the predominant model employed as $\boldsymbol{\epsilon}_\theta(\cdot)$ in Eq. 4 to predict Gaussian noise, can be divided into blocks that incorporate residual connections (such as Residual Bottleneck Blocks or Transformer Blocks [36]) and the remaining layers. Numerous PTQ approaches for diffusion models are grounded in layer/block-wise reconstruction [10, 23, 42, 44] to obtain optimal quantization parameters. For example, in the Residual Bottleneck Block, this approach typically minimizes the following loss function as its optimization objective:

$$\mathcal{L}_i = \|f_i(\cdot) - \widehat{f}_i(\cdot)\|_F^2, \tag{5}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm. The function $f_i(\cdot)$ represents the i^{th} Residual Bottleneck Block, and $\widehat{f}_i(\cdot)$ is its quantized counterpart. Furthermore, in the ensuing sections, we use n to denote the total number of Residual Bottleneck Blocks in a single diffusion model.

Temporal feature in diffusion models. Seeing from Fig. 1 (a), time-step t is encoded with `time embed`[1] and then passes through the `embedding layer`[2] in each Residual Bottleneck Block, resulting in a series of unique activations. In this paper, we denote these activations as temporal features. Notably, temporal features are independent of $\mathbf{x}_t$ and unrelated to other temporal features from different time-steps. To enhance clarity, we simplify our notation as follows: we represent `time embed` as $h(\cdot)$, the `embedding layer` in the i^{th} Residual Bottleneck Block as $g_i(\cdot)$, and denote the i^{th} temporal feature at time-step t as $\mathbf{emb}_{t,i}$. Moreover, as illustrated in Fig. 1 (a), the relationship is explicitly expressed by the equation:

$$\text{temporal feature} : \mathbf{emb}_{t,i} = g_i(h(t)) \tag{6}$$

Additionally, we have found that temporal features play a crucial role in the context of the diffusion model, holding unique and substantial physical implications. These features encompass temporal information that signifies the current image's temporal position along the denoising trajectory. Within the UNet structure, each time-step transforms into these temporal features, thereby controlling the denoising trajectory by applying them to the features of images generated at each iteration.

4. TFMQ for Diffusion Models

We present our novel PTQ framework in this section. We first observe temporal disturbance in previous methods in Sec. 4.1 and then analyze the inducements in Sec. 4.2. Finally, we propose our solutions in Sec. 4.3.

4.1. Temporal Feature Disturbance

Based on Sec. 3, we investigate the impact of previous PTQ works on temporal features, and we identify the phenomenon of temporal feature disturbance, which significantly deteriorates the quality of generated images.

Temporal feature error. We thoroughly analyze temporal feature variations before and after the quantization of `embedding layers` and `time embed` in the Stable Diffusion model ($T = 50, i = 11$). Prior to this analysis, we introduce the temporal feature error as defined by:

$$\cos(\mathbf{emb}_{t,i}, \widehat{\mathbf{emb}_{t,i}}), \tag{7}$$

[1] PyTorch time embed implementation in diffusion models.

[2] PyTorch embedding layer implementation in diffusion models.

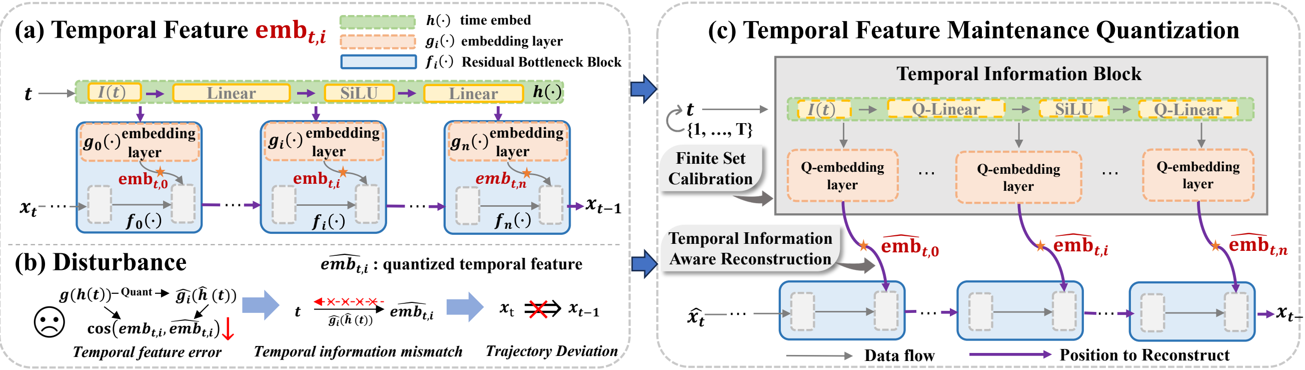

(a) Temporal Feature $\mathbf{emb}_{t,i}$
$h(\cdot)$ time embed
$g_i(\cdot)$ embedding layer
$f_i(\cdot)$ Residual Bottleneck Block
t
$I(t)$
Linear
SiLU
Linear
$h(\cdot)$
$g_0(\cdot)$ embedding layer
$\mathbf{emb}_{t,0}$
$f_0(\cdot)$
$g_i(\cdot)$ embedding layer
$\mathbf{emb}_{t,i}$
$f_i(\cdot)$
$g_n(\cdot)$ embedding layer
$\mathbf{emb}_{t,n}$
$f_n(\cdot)$
x_t
x_{t-1}
(b) Disturbance
$\widehat{emb}_{t,i}$: quantized temporal feature
$g(h(t)) \xrightarrow{\text{Quant}} \widehat{g_i}(\hat{h}(t))$
$\cos(emb_{t,i}, \widehat{emb}_{t,i})$
Temporal feature error
$t \xrightarrow[\widehat{g_i}(\hat{h}(t))]{} \widehat{emb}_{t,i}$
Temporal information mismatch
$x_t \Rightarrow x_{t-1}$
Trajectory Deviation
(c) Temporal Feature Maintenance Quantization
Temporal Information Block
t
{1, …, T}
$I(t)$
Q-Linear
SiLU
Q-Linear
Finite Set Calibration
Q-embedding layer
…
Q-embedding layer
…
Q-embedding layer
Temporal Information Aware Reconstruction
$\widehat{\mathbf{emb}}_{t,0}$
$\widehat{\mathbf{emb}}_{t,i}$
$\widehat{\mathbf{emb}}_{t,n}$
$\hat{x}_t$
x_{t-1}
Data flow
Position to Reconstruct

Figure 1. Overview of the proposed Temporal Feature Maintenance Quantization. (a) Temporal Feature $\mathbf{emb}_{t,i}$, belonging to a finite set representing temporal information, has been overlooked in previous works due to inappropriate reconstruction targets (box with a solid line). (b) This oversight leads to a severe disturbance for $\mathbf{emb}_{t,i}$ and results in the mismatch of crucial temporal information for the diffusion model's generation, causing a deviation in the denoising trajectory and a significant drop in accuracy. (c) Based on these analyses, we introduce a Temporal Information Block that exclusively correlates with the time-step t. Leveraging this $\mathbf{x}_t$-unrelated block, we enable Temporal Information Aware Reconstruction and Finite Set Calibration (utilizing the finite number of t). This approach achieves the maintenance of temporal features and yields state-of-the-art results.

where $\cos(\cdot)$ denotes cosine similarity, and $\widehat{\mathbf{emb}}_{t,i}$ signifies the temporal feature corresponding to $\mathbf{emb}_{t,i}$ in the quantized model. As illustrated in Fig. 2, quantization induces notable temporal feature errors. We term this phenomenon, characterized by substantial temporal feature errors within diffusion models, as temporal feature disturbance.

Temporal information mismatch. Temporal feature disturbance alters the original embedded temporal information. Specifically, $\mathbf{emb}_{t,i}$ is intended to correspond to time-step t. However, due to significant errors, the quantized model's $\widehat{\mathbf{emb}}_{t,i}$ is no longer accurately associated with t, resulting in what we term as temporal information mismatch:

$$t \leftarrow \mathbf{emb}_{t,i}, \quad t \nleftarrow \widehat{\mathbf{emb}}_{t,i}. \tag{8}$$

Furthermore, as depicted in Fig. 3, we even observe a pronounced temporal information mismatch. Specifically, the temporal feature generated by the quantized model at time-step t exhibits a divergence from that of the full-precision model at the corresponding time-step, Instead, it tends to align more closely with the temporal feature corresponding to $t+\delta_t$, importing wrong temporal information from $t+\delta_t$.

Trajectory deviation. Temporal information mismatch delivers wrong temporal information, therefore, causing a deviation in the corresponding temporal position of the image within the denoising trajectory, ultimately leading to:

$$\mathbf{x}_t \nRightarrow \mathbf{x}_{t-1}, \tag{9}$$

where we apply disrupted temporal features to the model. Evidently, as the deviation in the denoising trajectory accumulates with the increase in the number of denoising iterations, the final generated image struggles to align with $\mathbf{x}_0$. This evolution is illustrated in Fig. 4, where we maintain UNet excluding `embedding layers` and `time embed` in full precision.

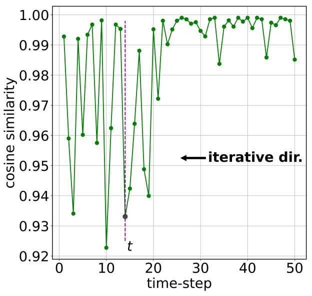


Figure 2. Temporal feature disturbance. The inflection points serve as indicators of temporal feature errors at different time-steps, and they highlight the significant phenomenon of temporal feature disturbance.

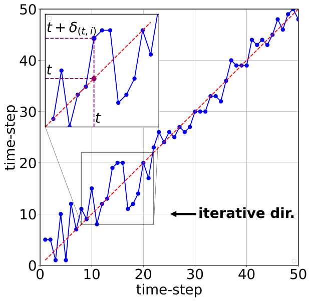


Figure 3. Temporal information mismatch. The coordinates of the inflection points on the blue curve can denoted as $(t, t+\delta_{t,i})$. It indicates $\mathbf{emb}_{t+\delta_{t,i},i}$ exhibits the highest similarity with $\widehat{\mathbf{emb}}_{t,i}$.

4.2. Inducement Analyses

In this section, we explore the two inducements of temporal feature disturbance. For the purpose of clarity, in the subsequent sections, "reconstruction" specifically points to slight weight adjustment for minimal quantization error, while "calibration" specifically refers to activation calibration.

Inappropriate reconstruction target. Previous PTQ works [10, 23, 44] have achieved remarkable progress on diffusion models. However, these existing methods

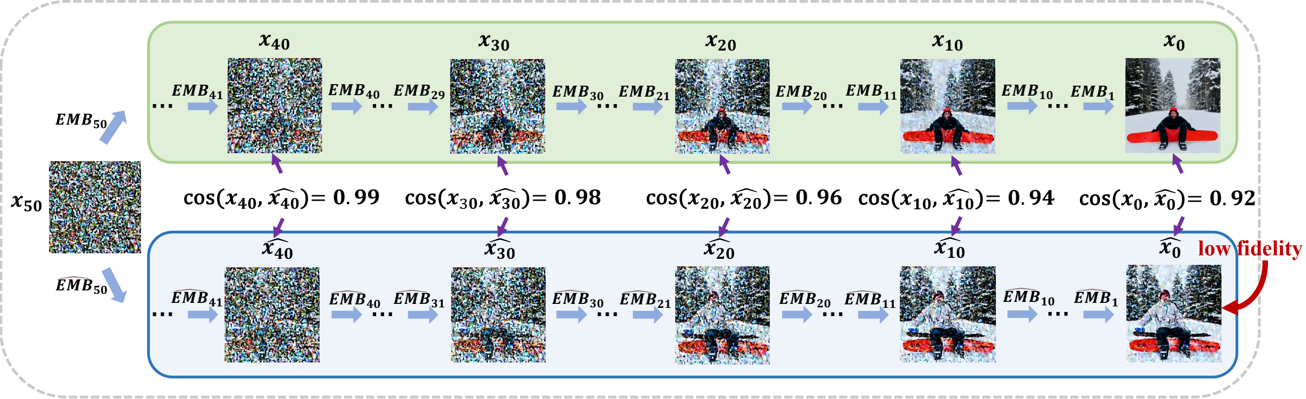


Figure 4. Denoising process of full-precision **(Upper)** and w4a8 quantized **(Lower)** Stable-Diffusion ($T = 50$) under the same experiment settings and prompt: *A man in the snow on a snow board*. We represent $\{\mathbf{emb}_{t,i}\}_{i=0,\ldots,n}$ and $\{\widehat{\mathbf{emb}}_{t,i}\}_{i=0,\ldots,n}$ as $\mathbf{EMB}_t$ and $\widehat{\mathbf{EMB}}_t$, respectively. Additionally, we denote $\widehat{\mathbf{x}}_t$ as $\mathbf{x}_t$ in the context of the quantized diffusion model. It is noteworthy that, in the quantized model employed here, **to showcase the impact of temporal features, only the layers in Temporal Information Block are quantized and the components unrelated to the generation of temporal features are maintained in full precision.**

overlook the temporal feature's independence and its distinctive physical significance. During their reconstruction processes, there was a lack of optimization for the `embedding layers`; instead, a Residual Bottleneck Block of coarser granularity was selected as the reconstruction target. This method involves two potential factors causing temporal feature disturbance:

- Optimize the objective as expressed in Eq. 5 to decrease the reconstruction loss of Residual Bottleneck Block, as opposed to direct reduction of temporal feature loss.
- During backpropagation of the reconstruction process, `embedding layers` independent from $\mathbf{x}_t$ are affected by $\mathbf{x}_t$, resulting in an overfitting scenario on limited calibration data.

To further validate our analyses, we respectively evaluate the FID [12] and sFID [41] for this reconstruction method, *e.g.*, BRECQ [24] and the approach where we freeze the parameters of the `embedding layers` during the reconstruction of the Residual Bottleneck Block, initializing the `embedding layers` solely through Min-max [33] for comparison. As shown in Tab. 1, the Freeze strategy exhibits better results, which verify that `embedding layers` serve as their own optimization objective and maintain their independence of $\mathbf{x}_t$ can significance mitigate temporal feature disturbance, especially at low-bit.

Table 1. FID and sFID on LSUN-Bedrooms 256 × 256 [49] for LDM-4 with 50000 sampling images. Prev represents BRECQ. Freeze denotes our trial.

Methods	Bits (W/A)	FID↓	sFID↓
Full Prec.	32/32	2.98	7.09
Prev	8/8	7.51	12.54
Freeze	8/8	**5.76** (-1.75)	**8.42** (-4.12)
Prev	4/8	9.36	22.73
Freeze	4/8	**7.08** (-2.28)	**16.82** (-5.91)

Unaware of finite activations within $h(\cdot)$ and $g_i(\cdot)$. We observe that, given T as a finite positive integer, the set of all possible activation values for `embedding layers` and `time embed` is finite and strictly dependent on time-steps. Within this set, activations corresponding to the same layer display notable range variations across different time-steps (refer to the appendix). Previous methods [44, 46] mainly focus on finding the optimal calibration method for $\widehat{\mathbf{x}}_t$-related network components. Moreover, akin to the first inducement, their calibration is directly towards the Residual Bottleneck Block, which proves suboptimal (refer to the appendix). However, based on the finite activations, we can employ calibration methods, especially for these time information-related activations, to better adapt to their range variations.

4.3. Temporal Feature Maintenance

To address the problem of temporal feature disturbance, we design a novel Temporal Information Block to maintain the temporal features. Based on the block, Temporal Information Aware Reconstruction and Finite Set Calibration are proposed to solve the two inducements analyzed above.

Temporal Information Block. Based on the inducements, it is crucial to meticulously separate the reconstruction and calibration process for each `embedding layer` and Residual Bottleneck Block to enhance quantized model performance. Considering the unique structure of the UNet, we consolidate all `embedding layers` and `time embed` into a unified Temporal Information Block, which can be denoted as $\{g_i(h(\cdot))\}_{i=0,\ldots,n}$ (see Fig. 1 (c)).

Temporal information aware reconstruction. Based on the Temporal Information Block, we propose Temporal information aware reconstruction (TIAR) to tackle the first inducement. The optimization objective for the block during the reconstruction process is as follows:

$$\mathcal{L}_{TIB} = \sum_{i=0}^{n} \|g_i(h(t)) - \widehat{g}_i(\widehat{h}(t))\|_F^2, \tag{10}$$

where $\widehat{h}(\cdot)$ and $\widehat{g}_i(\cdot)$ are quantized versions of $h(\cdot)$ and $g(\cdot)$, respectively. With this reconstruction, weights are adjusted to pursue a minimal disturbance for temporal features.

Finite set calibration. To address the challenge posed by the wide span of activations within a finite set for the second inducement, we propose Finite Set Calibration (FSC) for activation quantization. This strategy employs T sets of quantization parameters for every activation within all the `embedding layers` and `time embed`, such as $\{(s_T, z_T), \dots, (s_1, z_1)\}$ for activation $\mathbf{x}$. In time-step t, the quantization function for the $\mathbf{x}$ can be expressed as:

$$\hat{\mathbf{x}} = \Phi(\left\lfloor \frac{\mathbf{x}}{s_t} \right\rceil + z_t, 0, 2^b - 1). \tag{11}$$

To be noted, the calibration target is also aligned with the output of the Temporal Information Block. For a more specific estimation method of activation ranges, since the solution space within the finite set is limited, we find the Minmax [33] method can achieve satisfactory results with high efficiency (more evidence in Sec. 5.3).

5. Experiments

5.1. Implementation details

Models and datasets. In this section, we conduct image generation experiments to evaluate the proposed TFMQ-DM framework on various diffusion models: pixel-space diffusion model DDPM [14] for unconditional image generation, latent-space diffusion model LDM [39] for unconditional image generation and class-conditional image generation. We also apply our work to Stable Diffusion-v1-4 for text-guided image generation. In our experiments, We use seven standard benchmarks: CIFAR-10 32 × 32 [22], LSUN-Bedrooms 256 × 256 [49], LSUN-Churches 256 × 256 [49], CelebA-HQ 256 × 256 [18], ImageNet 256 × 256 [4], FFHQ 256 × 256 [19] and MS-COCO [25].

Quantization settings. We use channel-wise quantization for weights and layer-wise quantization for activations, as it is a common practice. In our experimental setup, we employ BRECQ [24] and AdaRound [32]. Drawing from empirical insights derived from conventional model quantization practices [9, 37], we maintain the input and output layers of the model in full precision. Calibration sets, integral to our methodology, are generated through full-precision diffusion models, mirroring the approach outlined in Q-Diffusion [23]. Moreover, in weight quantization, we reconstruct quantized weights for 20k iterations with a mini-batch size of 32 for DDPM and LDM, and 8 for Stable Diffusion. In activation quantization, we utilize EMA [16] to estimate the ranges of activations with a mini-batch size of 16 on all models. More details can be found in the appendix.

Evaluation metrics. For each experiment, we evaluate the performance of diffusion models with Fréchet Inception Distance (FID) [12]. In the case of LDM and Stable Diffusion experiments, we also include sFID [41], which better captures spatial relationships than FID. For ImageNet and CIFAR-10 experiments, we additionally provide Inception Score (IS) [41] as a reference metric. Further, in the context of Stable Diffusion experiments, we extend our evaluation to include the compatibility of image-caption pairs, employing the CLIP score [11]. The ViT-B/32 is used as the backbone when computing the CLIP score. To ensure consistency in the reported outcomes, all results are derived from our own implementation or from other papers, where experiments are conducted under conditions consistent with ours. More specifically, in the evaluation process of each experiment, we sample 50k images from DDPM or LDM, or 30k images from Stable-Diffusion. All experiments are conducted utilizing one H800 GPU and implemented with the PyTorch framework [35].

5.2. Main Results

Unconditional image generation. In the experiments conducted on the LDM, we maintain the same experimental settings as presented in [39], including the number of steps, variance schedule, and classifier-free guidance scale (denoted by eta and cfg in the following, respectively). As shown in Tab. 2, the FID performance differences relative to the full precision (FP) model are all within 0.7 for all settings. Specifically, on the CelebA-HQ 256 × 256 dataset, our method exhibits a FID reduction of 6.71 and a sFID reduction of 6.60 in the w4a8 setting compared to the current state-of-the-art (SOTA). It is noticeable that existing methods, whether in 4-bit or 8-bit, show significant performance degradation when compared to the FP model on face datasets like CelebA-HQ 256 × 256 and FFHQ 256×256, whereas our TFMQ-DM shows almost no performance degradation compared to the FP model. Importantly, our method achieves significant performance improvement on the LSUN-Bedrooms 256 × 256 in the w4a8 setting, with FID and sFID reductions of 2.26 and 7.51 compared to PTQD [10], respectively. Regarding LDM-8 on LSUN-Churches 256 × 256, we attribute the moderate improvement, compared to other datasets. We believe that the use of the LDM-8 model with a downsampling factor of 8 may

Table 2. Quantization results for unconditional image generation with LDM-4 on LSUN-Bedrooms 256, FFHQ 256 and CelebA-HQ 256 × 256, LDM-8 on LSUN-Churches 256 × 256. * represents our implementation according to open-source codes and †means directly rerunning open-source codes.

Methods	Bits (W/A)	**LSUN-Bedrooms** 256 × 256		**LSUN-Churches** 256 × 256		**CelebA-HQ** 256 × 256		**FFHQ** 256 × 256	
		FID↓	sFID↓	FID↓	sFID↓	FID↓	sFID↓	FID↓	sFID↓
Full Prec.	32/32	2.98	7.09	4.12	10.89	8.74	10.16	9.36	8.67
PTQ4DM* [42]	4/32	4.83	7.94	4.92	13.94	13.67	14.72	11.74	12.18
Q-Diffusion† [23]	4/32	4.20	7.66	4.55	11.90	11.09	12.00	11.60	10.30
PTQD* [10]	4/32	4.42	7.88	4.67	13.68	11.06	12.21	12.01	11.12
TFMQ-DM (Ours)	4/32	**3.60** **(-0.60)**	**7.61** **(-0.05)**	**4.07** **(-0.48)**	**11.41** **(-0.49)**	**8.74** **(-2.32)**	**10.18** **(-1.82)**	**9.89** **(-1.71)**	**9.06** **(-1.24)**
PTQ4DM* [42]	8/8	4.75	9.59	4.80	13.48	14.42	15.06	10.73	11.65
Q-Diffusion† [23]	8/8	4.51	8.17	4.41	12.23	12.85	14.16	10.87	10.01
PTQD [10]	8/8	3.75	9.89	4.89*	14.89*	12.76*	13.54*	10.69*	10.97*
TFMQ-DM (Ours)	8/8	**3.14** **(-0.61)**	**7.26** **(-0.91)**	**4.01** **(-0.40)**	**10.98** **(-1.25)**	**8.71** **(-4.05)**	**10.20** **(-3.34)**	**9.46** **(-1.23)**	**8.73** **(-1.28)**
PTQ4DM [42]	4/8	20.72	54.30	4.97*	14.87*	17.08*	17.48*	11.83*	12.91*
Q-Diffusion† [23]	4/8	6.40	17.93	4.66	13.94	15.55	16.86	11.45	11.15
PTQD [10]	4/8	5.94	15.16	5.10*	13.23*	15.47*	17.38*	11.42*	11.43*
TFMQ-DM (Ours)	4/8	**3.68** **(-2.26)**	**7.65** **(-7.51)**	**4.14** **(-0.52)**	**11.46** **(-1.77)**	**8.76** **(-6.71)**	**10.26** **(-6.60)**	**9.97** **(-1.45)**	**9.14** **(-2.01)**

be more quantization-friendly. Existing methods have already achieved satisfactory results on this dataset. Nonetheless, our method still approaches the performance of the FP model more closely compared to existing methods.

Table 3. Quantization results for unconditional image generation with DDIM on CIFAR-10 32 × 32.

Methods	Bits (W/A)	**CIFAR-10** 32 × 32	
		IS↑	FID↓
Full Prec.	32/32	9.04	4.23
PTQ4DM* [42]	4/32	9.02	5.65
Q-Diffusion† [23]	4/32	8.78	5.08
TDQ [44]	4/32	-	-
TFMQ-DM (Ours)	4/32	**9.14** **(+0.12)**	**4.73** **(-0.35)**
PTQ4DM [42]	8/8	9.02	19.59
Q-Diffusion† [23]	8/8	8.89	4.78
TDQ [44]	8/8	8.85	5.99
TFMQ-DM (Ours)	8/8	**9.07** **(+0.05)**	**4.24** **(-0.54)**
PTQ4DM* [42]	4/8	8.93	5.14
Q-Diffusion† [23]	4/8	9.12	4.98
TDQ [44]	4/8	-	-
TFMQ-DM (Ours)	4/8	**9.13** **(+0.01)**	**4.78** **(-0.20)**

We also conduct experiments with DDPM on CIFAR-10 32 × 32. Due to the lower resolution and simplicity of the images in this dataset, existing methods show minimal performance degradation. As seen in Tab. 3, while the results we obtain may not be as pronounced, we still achieve comprehensive improvements in terms of IS and FID compared to the existing SOTA.

Table 4. Quantization results for unconditional image generation with class-conditional image generation with LDM-8 on ImageNet 256 × 256.

Methods	Bits (W/A)	**ImageNet** 256 × 256		
		IS↑	FID↓	sFID↓
Full Prec.	32/32	235.64	10.91	7.67
PTQ4DM [42]	4/32	-	-	-
Q-Diffusion* [23]	4/32	213.56	11.87	8.76
PTQD† [10]	4/32	201.78	11.65	9.06
TFMQ-DM (Ours)	4/32	**223.81** **(+10.25)**	**10.50** **(-1.15)**	**7.98** **(-0.78)**
PTQ4DM [42]	8/8	161.75	12.59	-
Q-Diffusion* [23]	8/8	187.65	12.80	9.87
PTQD [10]	8/8	153.92	11.94	8.03
TFMQ-DM (Ours)	8/8	**198.86** **(+11.21)**	**10.79** **(-1.15)**	**7.65** **(-0.38)**
PTQ4DM [42]	4/8	-	-	-
Q-Diffusion* [23]	4/8	212.51	10.68	14.85
PTQD [10]	4/8	214.73	10.40	12.63
TFMQ-DM (Ours)	4/8	**221.82** **(+7.09)**	**10.29** **(-0.11)**	**7.35** **(-5.28)**

Table 5. Quantization results for text-guided image generation with Stable-Diffusion on MS-COCO captions.

Methods	Bits (W/A)	**MS-COCO**		
		FID↓	sFID↓	CLIP↑
Full Prec.	32/32	13.15	19.31	0.3146
Q-Diffusion† [23]	4/32	13.58	19.50	0.3143
TFMQ-DM (Ours)	4/32	**13.21** **(-0.37)**	**19.03** **(-0.47)**	**0.3144** **(+0.0001)**
Q-Diffusion† [23]	8/8	13.31	20.54	0.3134
TFMQ-DM (Ours)	8/8	**13.09** **(-0.22)**	**19.91** **(-0.63)**	**0.3134** **(+0.0000)**
Q-Diffusion† [23]	4/8	14.49	20.43	0.3121
TFMQ-DM (Ours)	4/8	**13.36** **(-1.13)**	**20.14** **(-0.29)**	**0.3128** **(+0.0007)**

Class-conditional image generation. On the ImageNet 256 × 256 dataset, we employed a denoising process with 20 iterations, setting eta and cfg to 0.0 and 3.0, respectively. Compared to PTQD, our method achieved a FID reduction of 1.15 on both w4a32 and w8a8. Simultaneously, in the w4a8 setting, sFID decreased by 5.28. Under the same conditions, we observed an improvement of over 7 in IS. Particularly noteworthy is that, across various quantization settings, our method consistently achieved lower FID compared to the FP model.

Text-guided image generation. In this experiment, we sample high-resolution images of 512 × 512 pixels with 50 denoising steps and fix cfg to the default 7.5 in Stable Dif-

Table 6. The effect of different methods proposed in the paper on LSUN-Bedrooms 256 × 256.

Methods	**Bits (W/A)**	FID↓	sFID↓
Full Prec.	32/32	2.98	7.09
BRECQ [24] (Baseline)	4/8	9.36	22.73
+TIAR	4/8	4.84	9.29
+FSC	4/8	6.07	11.31
+TFMQ-DM (TIAR + FSC)	4/8	**3.68** **(-5.68)**	**7.65** **(-15.08)**

fusion as the trade-off between sample quality and diversity. In Tab. 5, compared to the current SOTA Q-Diffusion, our approach achieves an FID reduction of 1.13 on w4a8. Simultaneously, our FID on w8a8 and sFID on w4a32 are even lower than those of the full precision model. However, existing metrics fail to adequately assess the semantic consistency of generated images. Nevertheless, based on the images generated in the appendix, our method produces higher-quality images with more realistic details, better demonstrating semantic information. Furthermore, our generated images closely approximate the effects of FP model.

5.3. Ablation Study

To evaluate the effectiveness of each proposed method, we perform a thorough ablation study on the LSUN-Bedrooms 256 × 256 dataset with w4a8 quantization, utilizing the LDM-4 model with a DDIM sampler, as outlined in Tab. 6. We begin the assessment with a baseline BRECQ [24], which shows ineffective in denoising images when operating with 4-bit quantization. Additional ablation experiments can be found in the appendix.

Effect of TIAR. It can be observed that, compared to the baseline, our TIAR method reduces FID and sFID by 4.52 and 13.44, respectively. Furthermore, As shown in Fig. 5, our method's temporal feature disturbance is significantly far weaker than existing SOTA PTQD. This indicates the effectiveness of our method in maintaining temporal information contained in temporal features. Further details of the effects are presented in the appendix.

Effect of FSC. Our FSC method has also achieved remarkably positive results. Compared to the baseline, it reduces FID and sFID by 3.29 and 11.42, respectively. Compared to PTQD, from Fig. 5, our method's temporal feature error is significantly smaller than that of PTQD. Since we employ layer-wise quantization for activations, this introduces less than one percent of additional parameters. The inference time overhead incurred by switching different step sizes and zero points during multiple time-steps inference is negligible, as detailed in the appendix.

Efficiency of FSC. For FSC, there are various methods to evaluate the range of activations to determine the optimal step size. We try several methods and assess the GPU time consumed during calibration, as detailed in Tab. 7. Since the improvements achieved by these methods in model performance are similar, we opt for the simplest and most efficient Min-max [33] method as our specific calibration strategy, striking a balance between calibration time and effectiveness. Notably, we have found that PTQD and Q-Diffusion cost 4.68 and 5.29 GPU hours in their PTQ methods under w4a8 quantization on LSUN-Bedrooms 256 × 256, respectively. However, our framework only spends 2.32 GPU hours (as shown in Tab. 7).

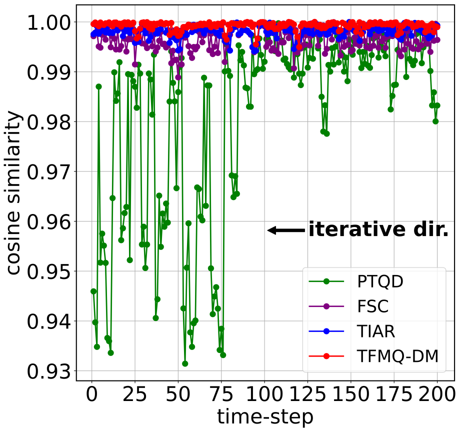


Figure 5. Temporal feature errors across different PTQ Methods.

Table 7. Different calibration methods for FTFC. We note T as calibration GPU hours in the table.

TSC Methods	**Bits (W/A)**	FID↓	sFID↓	T (hours)
FIAR	8/32	2.84	7.08	2.18
+LSQ [5]	8/8	3.17	7.18	2.48
+KL-divergence	8/8	3.27	7.32	19.67
+Percentile	8/8	3.34	7.41	12.00
+MSE	8/8	**3.12**	**7.12**	8.89
+Min-max [33]	8/8	3.14 (+0.02)	7.26 (+0.14)	**0.12** **(-2.36)**
FIAR	4/32	3.04	7.16	2.20
+LSQ [5]	4/8	3.69	7.48	2.57
+KL-divergence	4/8	3.94	**7.42**	19.65
+Percentile	4/8	3.74	8.02	12.04
+MSE	4/8	**3.62**	7.48	8.89
+Min-max [33]	4/8	3.68 (+0.06)	7.65 (+0.23)	**0.12** **(-2.45)**

Furthermore, for the only learning-based method in the table, LSQ [5], which is one of the most commonly used methods in previous works [10, 23, 42], we observe that it did not outperform other methods and, in some cases, performs even worse. We speculate that the main reason might be that, for a fixed time-step, the calibration data used to determine the quantization parameters is relatively limited compared to all calibration data. In such cases, LSQ may struggle to learn an optimal set of parameters.

6. Conclusion

This research explores the application of quantization for accelerating diffusion models. In this work, we identify a novel and significant problem, namely temporal feature disturbance, in the quantization of diffusion models. We conducted a detailed analysis of the root causes of this problem and introduced our TFMQ-DM quantization framework. In 4-bit quantization on extensive datasets and different diffusion models, this framework exhibits minimal performance degradation compared to the FP model and speedup quantization time.

References

[1] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization, 2020. 1, 2

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 1

[3] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, pages 12413–12422, 2022. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[5] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *ICLR*, 2020. 1, 2, 8

[6] Giulio Franzese, Simone Rossi, Lixuan Yang, Alessandro Finamore, Dario Rossi, Maurizio Filippone, and Pietro Michiardi. How much is enough? a study on diffusion times in score-based generative models. *arXiv preprint arXiv:2206.05173*, 2022. 2

[7] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, pages 4852–4861, 2019. 1, 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[9] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 6

[10] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. In *NeurIPS*, 2023. 2, 3, 4, 6, 7, 8, 11, 12

[11] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. 6

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 5, 6

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 1

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 3, 6, 11

[15] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020. 1, 2

[16] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, pages 2704–2713, 2018. 2, 6

[17] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis, 2023. 1

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. 6, 12

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 6

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 1

[21] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 1, 2

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[23] Xiuyu Li, Long Lian, Yijiang Liu, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *ICCV*, 2023. 1, 2, 3, 4, 6, 7, 8, 11, 12

[24] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021. 1, 2, 5, 6, 8

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 6

[26] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. In *IJCAI*, pages 1173–1179, 2022. 2

[27] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022. 1, 11

[28] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. Relaxed quantization for discretized neural networks. In *ICLR*, 2019. 2

[29] 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. In *NeurIPS*, 2022. 1

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. 12

[31] Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524*, 2022. 2

[32] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *ICML*, pages 7197–7206, 2020. 1, 2, 6

[33] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization, 2021. 1, 5, 6, 8, 11

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171, 2021. 3

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 6

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 3

[37] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542, 2016. 6

[38] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech, 2022. 1

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 6

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 3

[41] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. 5, 6

[42] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, 2023. 2, 3, 7, 8

[43] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2018. 1

[44] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. In *NeurIPS*, 2023. 2, 3, 4, 5, 7

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2, 3, 11

[46] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate data-free quantization for diffusion models. *arXiv preprint arXiv:2305.18723*, 2023. 2, 5

[47] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards accurate post-training network quantization via bit-split and stitching. In *ICML*, pages 9847–9856, 2020. 2

[48] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2022. 1, 2

[49] 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, 2016. 1, 5, 6, 11, 12

[50] Luoming Zhang, Yefei He, Zhenyu Lou, Xin Ye, Yuxing Wang, and Hong Zhou. Root quantization: a self-adaptive supplement ste. *Applied Intelligence*, 53(6):6266–6275, 2023. 2

[51] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022. 2

[52] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 1

[53] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models. *stat*, 1050:7, 2022. 2

[54] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *CVPR*, pages 7920–7928, 2018. 2

Appendix

Note: All references to figures or tables identified by Arabic numerals point to the corresponding figures or tables in the main text.

A. More Implementation Details

In our reconstruction and calibration, apart from the Temporal Information Block proposed by us, the partitioning of the remaining network components remains consistent with PTQD [10] and Q-Diffusion [23] (*i.e.*, Residual Bottleneck Blocks, Attention Blocks, and the remaining layers). Specifically, for the reconstruction of the Residual Bottleneck Block, we freeze the quantization parameters of the `embedding layer`, and these parameters are only tuned in the reconstruction within the Temporal Information Block.

Additionally, the quantization settings are kept consistent with Q-Diffusion and PTQD.

B. Activation Range Variations in Finite Set

We analyze activation value ranges across all time steps in sampling data-unrelated components, *e.g.*, `time embed` and `embedding layers` for diffusion models. In Fig. I, it is evident that activation ranges vary notably among different time steps within these components. This observation suggests that the activation ranges within the same layer undergo considerable changes with varying time steps. Fortunately, the activations in the Time Information Block belong to a finite set, providing us the opportunity to conduct an accurate calibration for each time step.

C. Inappropriate Calibration Target

In this part, we further conduct experiments to provide the clues that the inappropriate reconstruction target also results in an inappropriate calibration. In the previous works, they calibrate the `embedding layers` along with the corresponding Residual Bottleneck Blocks. On the contrary, we freeze the quantized parameters of the `embedding layers` during the calibration process with a simple Min-max [33] initialization, which separates the calibration of `embedding layers` as alone. The experimental results in Tab. I demonstrate that without calibrating these layers inside the Residual Bottleneck Block can achieve better results. This confirms that the inappropriate calibration target leads to the suboptimal tuning of the quantization parameters.

D. Additional Effect of TIAR

As shown in Fig. 5, both of our proposed methods for LDM-4 on LSUN-Bedrooms 256×256 significantly reduce temporal feature errors, thereby alleviating temporal feature disturbance to a great extent. In this section, we conduct a detailed analysis of the cosine similarity between the outputs of the i^{th} Residual Bottleneck Blocks before and after quantization. We compare the results obtained with our TIAR and PTQD under w4a8 quantization, where $i = 8$ and $T = 200$ (the same as the settings in Fig. 5). From Fig. II, it can be observed that our approach significantly reduces output errors of the Residual Bottleneck Block compared to PTQD. However, it is essential to note that the error at this point involves the accumulated errors from multiple denoising iterations in diffusion models. Since Fig. 5 is not subject to the impact of accumulated errors, the trends of the lines in the two graphs may exhibit slight differences.

Table I. FID and sFID on LSUN-Bedrooms 256×256 [49] for LDM-4. Prev represents BRECQ, the same as Tab. 1. Freeze denotes our trial here.

Methods	**Bits (W/A)**	FID↓	sFID↓
Full Prec.	32/32	2.98	7.09
Prev	8/8	7.51	12.54
Freeze	8/8	**6.87 (-0.64)**	**10.12 (-2.42)**
Prev	4/8	9.36	22.73
Freeze	4/8	**8.06 (-1.30)**	**18.47 (-4.26)**

E. Inference Cost of TSC

In this section, we assess the inference overhead of our TFMQ-DM method on real hardware, specifically the Intel® Xeon® Gold 6248R Processor. All floating-point and quantized operations are implemented using Intel's OpenVINO toolkit [3]. As illustrated in Table II, in comparison to the UNet quantized with the built-in w8a8 quantization method in the OpenVINO toolkit, our approach results in a memory overhead of less than 0.076%, yielding a 2.38× acceleration compared to the original floating-point model. Moreover, our method introduces less than 0.5% additional latency compared to the built-in w8a8 quantization in the OpenVINO toolkit.

Table II. Inference analysis of Stable Diffusion with 50 denoising time-steps on Intel CPU.

Methods	**Bits (W/A)**	UNet Size (Mb)	Latency (s)	Speedup
Full Prec.	32/32	3278.81	81.01	-
OpenVINO	8/8	821.15	33.93	2.39×
TFMQ-DM	8/8	821.77	34.07	2.38×

F. Study of Sampling with Advanced Samplers

Apart from employing the DDIM sampler [45], we also utilize a variant of DDPM [14] called PLMS [27] on the

[3] OpenVINO toolkit

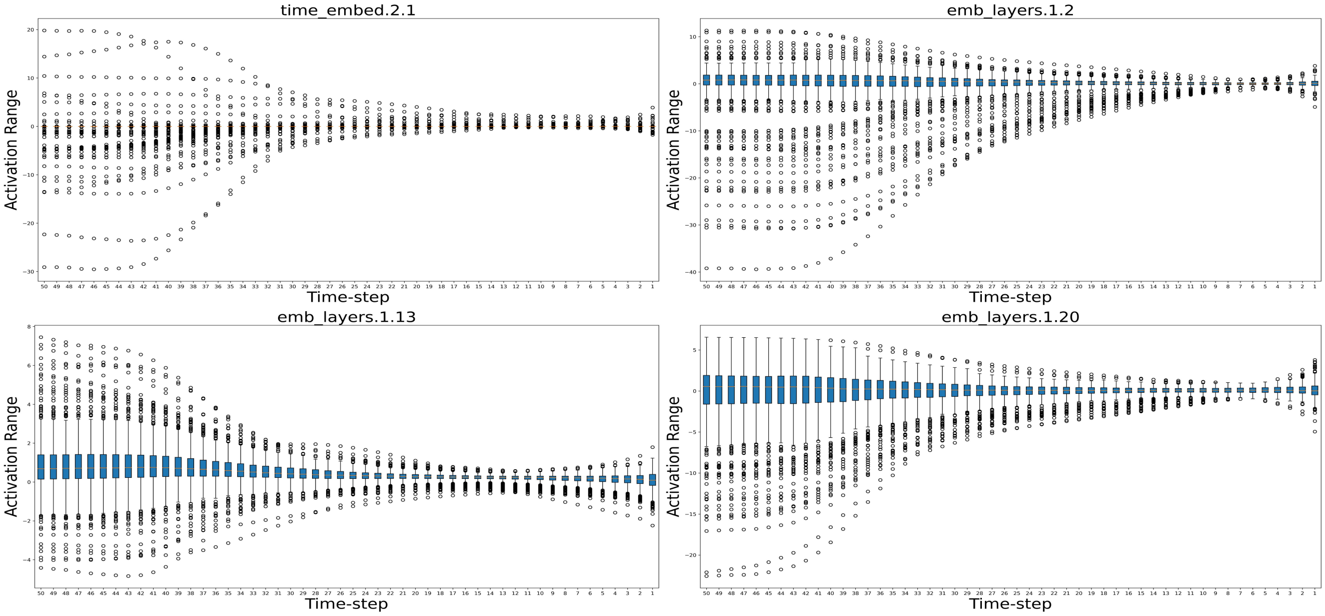


Figure I. Activation ranges within sampling data-unrelated components for LDM-4 on LSUN-Bedrooms 256 × 256 with 50 denoising steps. We randomly select 4 linear or convolutional layers' activations in these components to demonstrate the range variation.

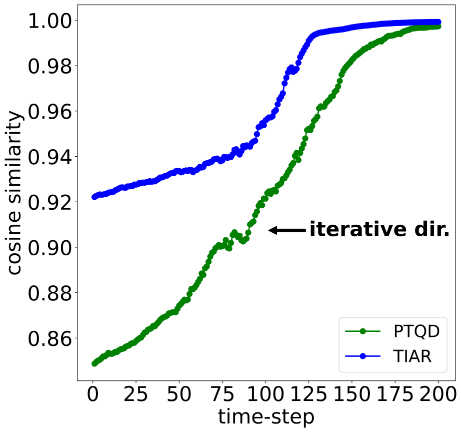


Figure II. Cosine similarity of the Residual Bottleneck's outputs across different PTQ Methods.

CelebA-HQ 256 × 256 dataset [18]. This better demonstrates the superiority of our TFMQ-DM framework compared to previous works. From Tab. III, the introduced TFMQ-DM substantially reduces FID and sFID, surpassing PTQD by margins of 12.40 and 7.09, respectively.

Additionally, we present experiments performed using the DPM++ solver [30] on LSUN-Churches 256 × 256 [49]. As illustrated in Tab. IV, our framework consistently outperforms existing methods in terms of performance on this dataset with the DPM++ solver.

Table III. Quantization results for unconditional image generation with PLMS on CelebA-HQ 256 × 256.

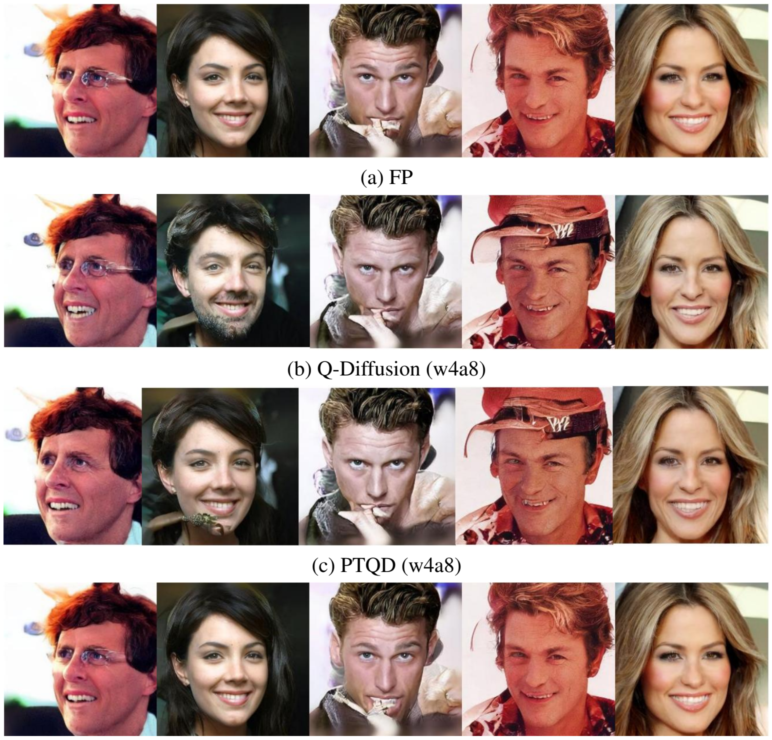
Methods	Bits (W/A)	CelebA-HQ 256 × 256	
		FID↓	sFID↓
Full Prec.	32/32	8.92	10.42
Q-Diffusion [23]	4/8	24.31	22.11
PTQD [10]	4/8	21.08	17.38
TFMQ-DM (Ours)	4/8	**8.68** **(-12.40)**	**10.29** **(-7.09)**

Table IV. Quantization results for unconditional image generation with DPM++ on LSUN-Churches 256 × 256.

Methods	Bits (W/A)	LSUN-Churches 256 × 256	
		FID↓	sFID↓
Full Prec.	32/32	4.12	10.55
Q-Diffusion [23]	4/8	7.80	23.24
PTQD [10]	4/8	7.45	22.74
TFMQ-DM (Ours)	4/8	**5.51** **(-1.94)**	**13.15** **(-9.59)**

G. Comparison of Visualization Results

Within this section, we present random samples derived from full-precision and w4a8 quantized diffusion models with a fixed random seed. These quantized models were created through our TFMQ-DM or previous state-of-the-art methods. The figures below illustrate the obtained results. As shown from Fig. III to Fig. IX, our framework yields results that closely resemble those of the full-precision model, showcasing higher fidelity. Moreover, it excels in finer details, producing superior outcomes in some intricate aspects (zoom in to closely examine the relevant images).



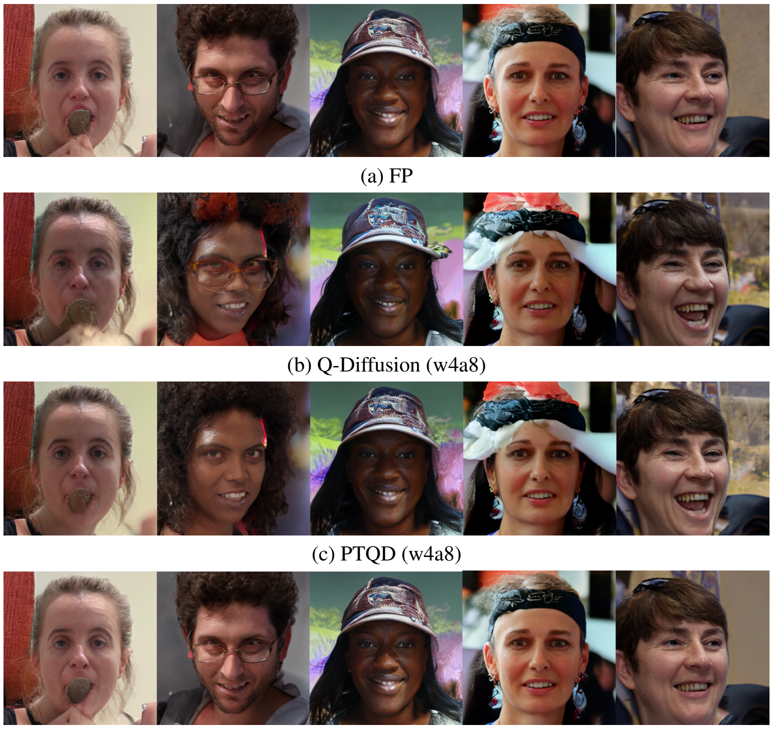
(a) FP

(b) Q-Diffusion (w4a8)

(c) PTQD (w4a8)

(d) TFMQ-DM (w4a8)

Figure III. Random samples from w4a8 quantized and full-precision LDM-4 on CelebA-HQ 256 × 256. The resolution of each sample is 256 × 256.



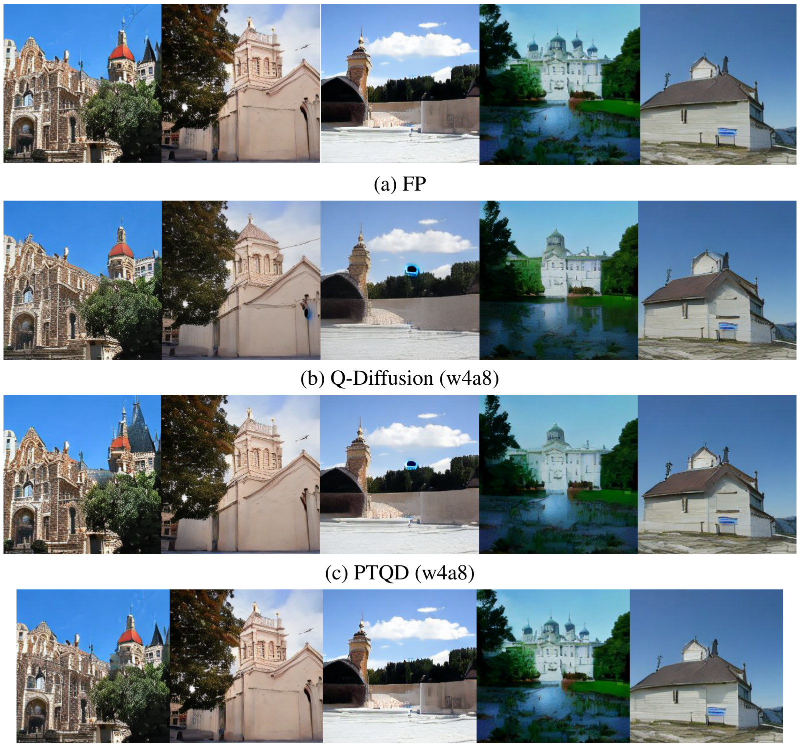
(a) FP

(b) Q-Diffusion (w4a8)

(c) PTQD (w4a8)

(d) TFMQ-DM (w4a8)

Figure IV. Random samples from w4a8 quantized and full-precision LDM-4 on FFHQ 256 × 256. The resolution of each sample is 256 × 256.



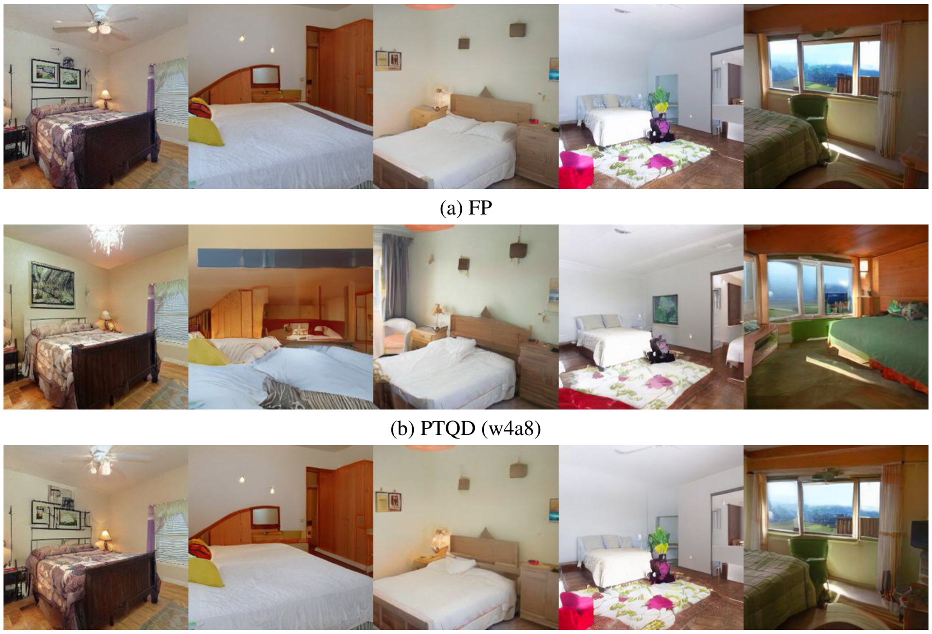
(a) FP

(b) Q-Diffusion (w4a8)

(c) PTQD (w4a8)

(d) TFMQ-DM (w4a8)

Figure V. Random samples from w4a8 quantized and full-precision LDM-8 on LSUN-Churches 256 × 256. The resolution of each sample is 256 × 256.

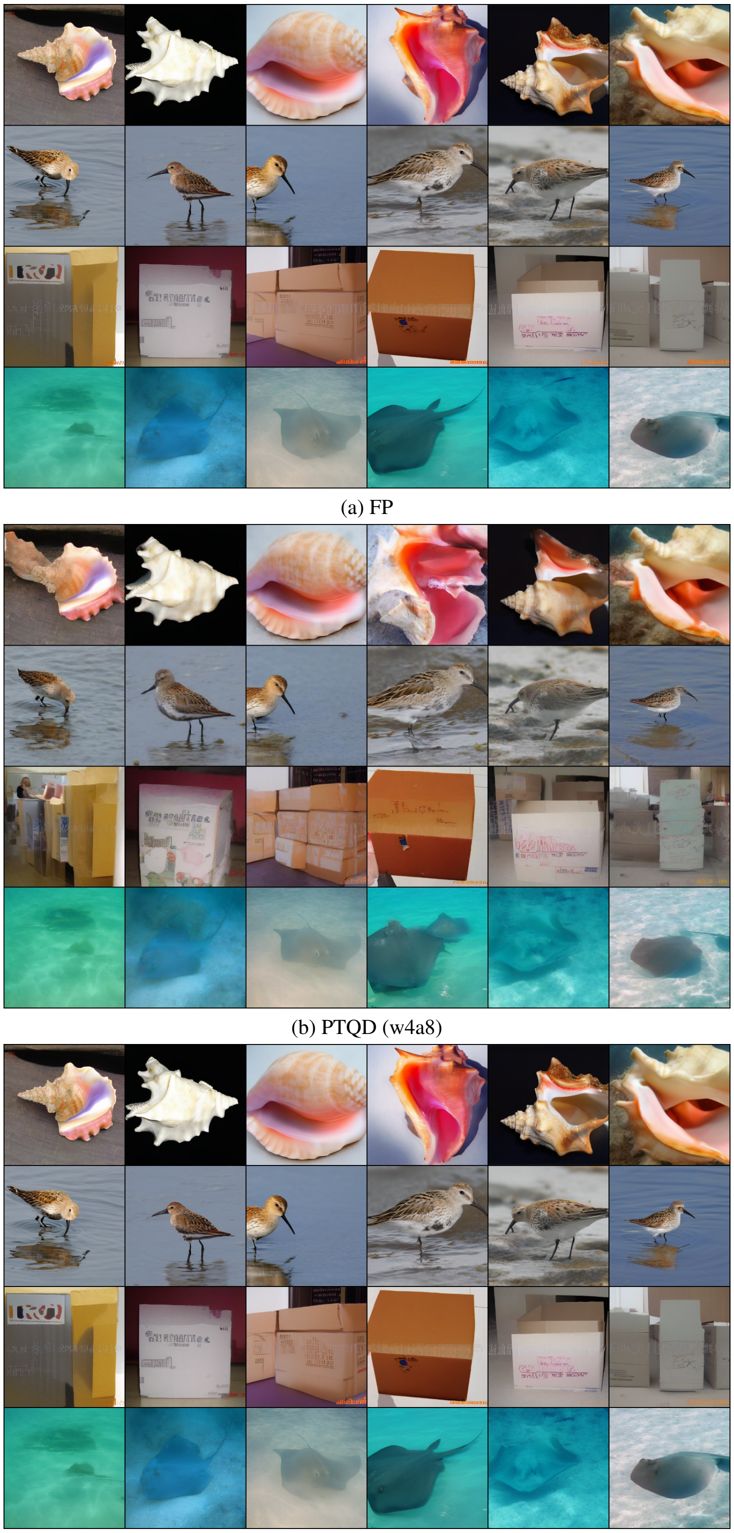


(a) FP

(b) PTQD (w4a8)

(c) TFMQ-DM (w4a8)

Figure VI. Random samples from w4a8 quantized and full-precision LDM-4 on LSUN-Bedrooms 256 × 256. The resolution of each sample is 256 × 256.

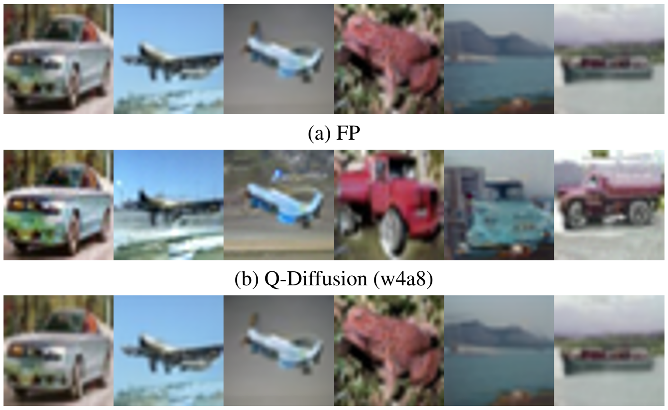


(a) FP

(b) PTQD (w4a8)

(c) TFMQ-DM (w4a8)

Figure VII. Random samples from w4a8 quantized and full-precision LDM-4 on ImageNet 256×256. The resolution of each sample is 256×256.

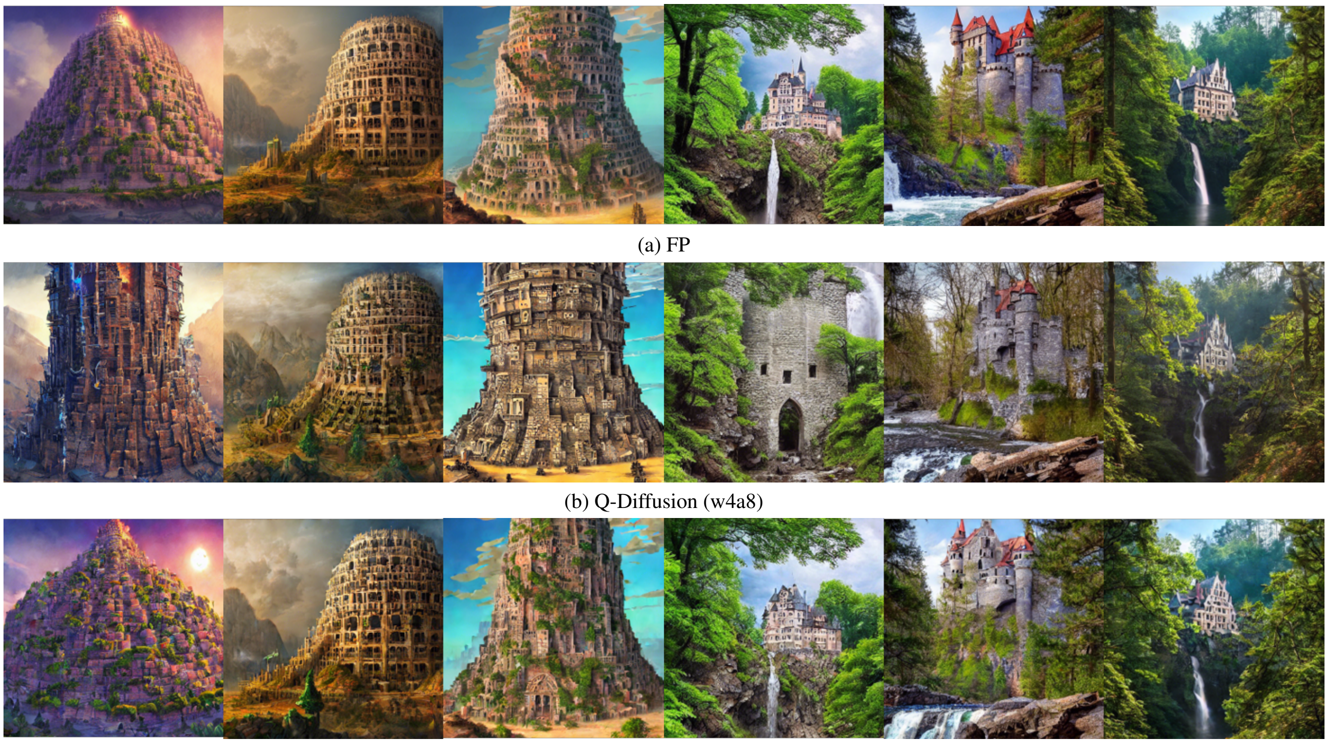


(a) FP

(b) Q-Diffusion (w4a8)

(c) TFMQ-DM (w4a8)

Figure VIII. Random samples from w4a8 quantized and full-precision DDIM on CIFAR-10 32×32. The resolution of each sample is 32×32.



(a) FP

(b) Q-Diffusion (w4a8)

(c) TFMQ-DM (w4a8)

Figure IX. Random samples from w4a8 quantized and full-precision Stable Diffusion. **(Left)** prompt: *A digital illustration of the Babel tower, detailed, trending in artstation, fantasy vivid colors*. **(Right)** prompt: *A beautiful castle beside a waterfall in the woods*. The resolution of each sample is 512×512.