DMQ: Dissecting Outliers of Diffusion Models for Post-Training Quantization

Dongyeun Lee Jiwan Hur Hyounguk Shon Jae Young Lee Junmo Kim
KAIST
{ledoye, jiwan.hur, hyounguk.shon, mcneato, junmo.kim}@kaist.ac.kr

Figure 1. Images generated by Stable Diffusion [46] in full precision (top row) and in W4A8-quantization using our method (bottom row).

Abstract

Diffusion models have achieved remarkable success in image generation but come with significant computational costs, posing challenges for deployment in resource-constrained environments. Recent post-training quantization (PTQ) methods have attempted to mitigate this issue by focusing on the iterative nature of diffusion models. However, these approaches often overlook outliers, leading to degraded performance at low bit-widths. In this paper, we propose a DMQ which combines Learned Equivalent Scaling (LES) and channel-wise Power-of-Two Scaling (PTS) to effectively address these challenges. Learned Equivalent Scaling optimizes channel-wise scaling factors to redistribute quantization difficulty between weights and activations, reducing overall quantization error. Recognizing that early denoising steps, despite having small quantization errors, crucially impact the final output due to error accumulation, we incorporate an adaptive timestep weighting scheme to prioritize these critical steps during learning. Furthermore, identifying that layers such as skip connections exhibit high inter-channel variance, we introduce channel-wise Power-of-Two Scaling for activations. To ensure robust selection of PTS factors even with small calibration set, we introduce a voting algorithm that enhances reliability. Extensive experiments demonstrate that our method significantly outperforms existing works, especially at low bit-widths such as W4A6 (4-bit weight, 6-bit activation) and W4A8, maintaining high image generation quality and model stability. The code is available at https://github.com/LeeDongYeun/dmq.

1. Introduction

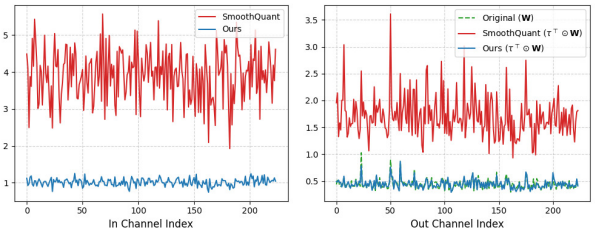
Diffusion models [16, 41, 43, 46] have recently shown remarkable success in synthesizing high-fidelity images, enabling various applications such as image editing [19, 26, 36, 47, 50, 69], 3D generation [31, 44, 47, 68], and video [3, 17, 20, 51] generation. However, despite their impressive generative capabilities, diffusion models face significant computational challenges due to the iterative nature of the denoising process, which can require hundreds or even thousands of steps. Consequently, accelerating this process while preserving output quality is essential for practical applications and has become a key research focus in the field.

Quantization [11, 21, 39] reduces computational and memory demands by lowering the bit-precision of neural networks. However, quantization of diffusion models presents unique challenges due to their iterative nature. The iterative nature results in highly varying activation distributions across multiple timesteps and quantization error accumulation as denoising progresses, hindering accurate quantization. To overcome this, several methods have explored Post-Training Quantization (PTQ) for diffusion models, focusing on composition of calibration data considering multiple timesteps [28, 35, 49, 57], adapting quantization parameters across timesteps [18, 52] or correcting noise within the diffusion process [13, 62]. However, they often overlook channel-wise outliers which stretch the quantization range, hindering effective quantization of non-outlier channels. As a result, existing methods often exhibit large performance degradation in low-bit settings (e.g. W4A6).

In this work, we address the challenge of outliers in dif-

fusion model quantization. A natural approach to mitigating outliers is Equivalent Scaling—originally introduced for LLMs quantization [32, 59, 61]—which reduces quantization difficulty by transferring the challenges of outlier channels in activations to non-outlier channels in weights and vice versa. This involves simultaneously dividing each activation channel and multiplying the corresponding weights by a channel-specific scaling factor. Although this approach has shown promise in LLMs, its naive application often fails for several reasons. First, inaccurate scaling factors can drastically degrade output quality. As shown in Tab. 1, SmoothQuant [61], which determines scaling factor based on the ratio of maximum activation and weight magnitudes, results in poor performance for diffusion models. Since diffusion models typically exhibit much larger activations than weights, SmoothQuant yields large scaling factor (Fig. 2a), which significantly increases quantization range of weight and in turn amplifies weight quantization errors(Fig. 2b). In diffusion models, weights are used at every sampling step, making inaccurate weight quantization severely degrade performance, with errors compounding over iterations. Second, Equivalent Scaling redistributes outliers between activations and weights but does not eliminate outliers. By scaling activations down and compensating with an inverse scaling of weights, this method reduces the overall quantization difficulty. However, it inherently shifts the burden rather than removing it. Consequently, layers exhibiting extremely large outliers remain problematic.

To address these issues, we propose DMQ, a new post-training quantization (PTQ) framework tailored for diffusion models. DMQ unifies two key techniques: **L**earned **E**quivalent **S**caling (**LES**) and channel-wise **P**ower-of-**T**wo **S**caling (**PTS**), which together ensure accurate quantization even under low-bit constraints. First, LES learns accurate scaling factors that balance quantization difficulty between activation and weight by minimizing overall quantization error. We observe that early denoising steps, despite having smaller quantization errors, significantly influence the final output quality. To account for this, we introduce an adaptive timestep weighting loss that prioritizes learning in critical timesteps, enhancing overall quantization performance. Second, to handle extreme outliers that remain unresolved by Equivalent Scaling, we introduce PTS. Unlike LES, which compensates for reduced activation difficulty by increasing weight difficulty and vice versa, PTS directly removes extreme activation outliers by scaling them with power-of-two factors. These scaling factors are efficiently processed during weight loading via bit-shifting operations, ensuring that the quantization difficulty is not merely transferred but instead removed with minimal overhead. To prevent overfitting on small calibration datasets, we introduce an effective voting algorithm which selects optimal scaling factors based on statistical consensus across calibration samples, enhancing robustness and reliability.



(a) Equivalent scaling factor τ (b) Quantization range of $\mathbf{W}$

Figure 2. (a) Smoothquant results large equivalent scaling factor. (b) Large scaling factor expands quantization range (Max - Min).

Method	Weight Quant. Error	FID↓	sFID↓
Baseline [18]	0.0060	36.08	33.06
Baseline + SmoothQuant [61]	0.0694	454.16	254.41
LES (ours)	**0.0058**	**30.37**	**22.72**

Table 1. Comparison on FFHQ 256×256 in W4A8.

LES, which is applied across all layers without introducing extra overhead, finely adjusts outliers using floating-point scaling factors. By contrast, PTS is selectively used only for layers with extremely large outliers (e.g. skip connection), incurring a small overhead in the form of bit-shifting but enabling drastic outlier removal through power-of-two scaling. By combining the strengths of LES and PTS, our approach effectively addresses the challenges of outliers in diffusion model quantization. Extensive experiments demonstrate that DMQ consistently achieves superior performance across various datasets and architectures. Notably, we achieve stable 4-bit weight and 6-bit activation (W4A6) quantization—where previous methods often fail—while preserving high image generation quality.

2. Related work

Quantization. Quantization is a core technique for reducing memory overhead and accelerating inference by converting high-precision floating-point values into lower-bit integer approximations. Of the two primary quantization methods — Quantization-Aware Training (QAT) [5, 10, 22] and Post-Training Quantization (PTQ) [25, 29, 37, 38] — PTQ stands out for its efficiency in terms of time and data, as it bypasses the need for retraining. However, traditional methods often fail to accurately quantize diffusion models, as they require multiple denoising steps with a single model.

Quantization of diffusion models. Recently, various methods have explored quantization for diffusion models, enabling low-bit quantization while preserving performance [12, 13, 18, 28, 30, 35, 49, 52, 57, 62]. In QAT approach, Q-DM [30] proposes timestep-aware smoothing operation for attention. EfficientDM [12] introduces a quantization-aware low-rank adapter (LoRA) that is jointly quantized into low-bit. However, QAT requires intensive computation and data resources, making it less practical than PTQ.

Research in PTQ for diffusion models has largely branched into three directions: calibration data composition

[28, 49, 57], noise correction [13, 62], and timestep-specific quantization parameters [18, 52]. To enhance calibration data, PTQ4DM [49] samples calibration data from a normal distribution across timesteps, Q-Diffusion [28] employs uniform sampling, and EDA-DM [35] enhances alignment of distribution with original samples. Addressing quantization noise, PTQD [13] proposes to correct quantization noise correlated with the denoising process, and TAC-Diffusion [62] presents a timestep-aware correction to mitigate errors throughout the denoising process. Alternatively, TFMQ-DM [18] applies different quantization parameters across timesteps, whereas TDQ [52] proposes to learn these parameters. Despite the advancements of these methods, they often suffer from significant performance degradation under 8-bit activation because they overlook the outliers.

Equivalent channel-wise scaling. In a neural network, activations often have varying distributions across channels, making it challenging to accurately represent them by per-tensor quantization. This issue is particularly pronounced in large language models (LLMs), where extreme outliers in specific channels lead to a wide activation range, hindering accurate quantization [8, 58, 63]. To address this, several methods adopt equivalent channel-wise scaling, which scales the activations channel-wise and compensates for this scaling by inversely scaling the weights [32, 59, 61]. This approach maintains mathematical equivalence while compacting the activation range within each channel, facilitating finer quantization without loss of precision due to outliers.

Building on the success in LLMs, several recent studies have adopted Equivalent Scaling for DiT [43] quantization [60, 71]. Although they have achieved some success in DiT—where adaptation is straightforward due to their similarity to LLMs—they are based on SmoothQuant [61] which results suboptimal performance in Stable Diffusion (Tab. 1). Additionally, ViDiT-Q [71] has practical limitations, as the scaling factors vary with each timestep and therefore cannot be fused into the weights. Unlike these approaches, our method learns these factors to efficiently capture the dynamically changing activations across timesteps.

3. Preliminaries

Diffusion models [16, 53] are a family of generative models that approximate the data distribution $p(\boldsymbol{x}_0)$ by denoising process. Specifically, the forward Markov chain $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ perturbs the data with Gaussian noise and is defined as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}\right), \quad (1)$$

where β_t is a noise schedule that is a strictly decreasing function of time $t \in [0, T]$. Then, the learned reverse process aims to gradually recover the clean data $\boldsymbol{x}_0$ from the noised data $\boldsymbol{x}_t$ to generate novel samples:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)\right), \quad (2)$$

where the μ_θ and Σ_θ are neural networks that predict mean and variance of input $\boldsymbol{x}_t$ given timestep t [41].

To efficiently implement the iterative denoising process, diffusion models share the parameters across timesteps t using the position embedding [56]. This often raises challenges for quantizing diffusion models, as activation varies across timestep conditioning [28, 49]. Furthermore, quantization errors are introduced at each denoising step, which can compound over time [18, 28]. This accumulation of errors throughout the sampling process leads to the model drifting away from the training distribution, resulting in significant degradation of the final output quality [6, 27, 42].

Quantization. Among some quantization methods, we focus on uniform quantization in this work. Given a tensor $\mathbf{X}$, its quantization and dequantization functions are defined as:

$$\tilde{\mathbf{X}} = \text{clamp}\left(\left\lfloor \frac{\mathbf{X}}{s} \right\rceil, l, u\right), \quad \mathbf{X} \approx Q(\mathbf{X}) = s \cdot \tilde{\mathbf{X}}, \quad (3)$$

where s is the scale factor, and $\lfloor \cdot \rceil$ denotes round-to-nearest-integer operator. clamp$(\cdot)$ truncates values to $[l, u]$, yielding a low-bit representation $\tilde{\mathbf{X}}$ of $\mathbf{X}$. Here, the range $[l, u]$ is determined by the bit-width of the quantization. When s is a vector, $\mathbf{X}/s$ implies channel-wise division.

Consider a linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W} \approx (s^{(\mathbf{X})}\tilde{\mathbf{X}})(s^{(\mathbf{W})}\tilde{\mathbf{W}})$, where $\mathbf{X} \in \mathbb{R}^{B \times C_{in}}$ is the activation, $\mathbf{W} \in \mathbb{R}^{C_{in} \times C_{out}}$ is the weight, and $\mathbf{Y} \in \mathbb{R}^{B \times C_{out}}$ is the output. $s^{(\mathbf{X})}$ and $s^{(\mathbf{W})}$ denote the scales for $\mathbf{X}$ and $\mathbf{W}$, respectively. Note that the following discussion applies equally to convolutional layers, but for simplicity, we will illustrate the concepts with a linear layer. To enable efficient integer matrix multiplication, it is necessary to factor out the scale s, which is a high-precision value, from the summation. Under this condition, the matrix multiplication can be formulated as:

$$\mathbf{Y}_{ij} \approx s_i^{(\mathbf{X})} s_j^{(\mathbf{W})} \left(\sum_{k=1}^{C_{in}} \tilde{\mathbf{X}}_{ik} \cdot \tilde{\mathbf{W}}_{kj} \right). \quad (4)$$

This formulation shows that integer matrix multiplication is feasible when activations are quantized with per-tensor or per-sample (row) granularity and weights are quantized in per-tensor or per-channel (column) granularity. Since per-sample quantization requires online computation, per-tensor quantization is generally used for activations.

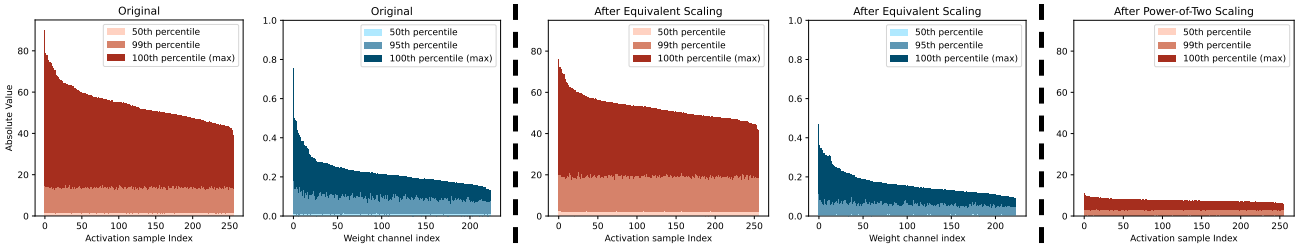
4. Method

4.1. Learned Equivalent Scaling

Channel-wise equivalent scaling. Our goal is to reduce quantization errors by optimizing channel-wise scaling factors $\tau \in \mathbb{R}^{C_{in}}$. We reformulate the matrix multiplication to incorporate these scaling factors:

$$\mathbf{Y} = (\mathbf{X}/\tau)\left(\tau^\top \odot \mathbf{W}\right) = \hat{\mathbf{X}}\hat{\mathbf{W}}, \quad (5)$$

where $/$ and $\odot$ denote channel-wise division and multiplication, respectively. By introducing τ, outliers are bidirectionally redistributed, resulting in scaled versions $\hat{\mathbf{X}}$ and $\hat{\mathbf{W}}$ that are more amenable to quantization.



(a) $\mathbf{X}$ (b) $\mathbf{W}$ (c) $\hat{\mathbf{X}} = [\mathbf{X} \oslash \tau]$ (d) $\hat{\mathbf{W}} = [\tau^\top \odot \mathbf{W}]$ (e) $\hat{\mathbf{X}} \odot \left(1/2^{\boldsymbol{\delta}}\right)$

Figure 3. Visualization of activations and weights in a skip connection layer, before and after applying Equivalent Scaling and Power-of-Two Scaling. Each values are sorted along x-axis by their maximum values. Darker colors indicate higher values, representing these prominent outliers. In the original distributions (a) and (b), both $\mathbf{X}$, $\mathbf{W}$ contain large outliers. After applying Equivalent scaling, large outliers are reduced in both the scaled activation $\hat{\mathbf{X}}$ and weights $\hat{\mathbf{X}}$, as shown in (c) and (d). When Power-of-Two Scaling is applied as in (e), the activation outliers are further diminished, demonstrating the effectiveness of PoT scaling in reducing extreme activation values.

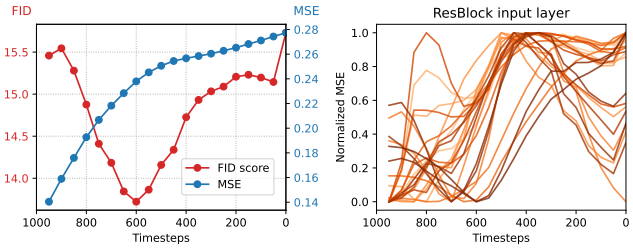


Figure 4. **(Left):** Plot of MSE and FID score across timesteps. Quantization noise increases as t decreases and the same amount of noise has a non-uniform impact on the final output. The MSE indicates qauntization error of activations averaged for all layers. **(Right):** Normalized quantization error for different layers over timesteps. Error trends vary across layers along timesteps.

The optimization objective to find the optimal τ for the i-th sample is defined as:

$$\mathcal{L}_i = \left\| \mathbf{X}_i \mathbf{W} - Q(\hat{\mathbf{X}}_i) Q(\hat{\mathbf{W}}) \right\|^2, \tag{6}$$

where $Q(\cdot)$ denotes standard MinMax quantizer [39]. Minimizing $\mathcal{L}_i$ ensures that the quantized outputs closely approximate the original outputs. As shown in the Fig. 3, the quantization difficulties caused by outliers in weights and activations are bidirectionally redistributed to non-outlier elements, thereby reducing the overall quantization error.

Analysis on quantization error. In a diffusion model, the activation distribution varies dynamically across timesteps, and quantization errors are accumulated over the denoising process [18, 28]. Therefore, before directly optimizing the objective loss across all timesteps, it is crucial to carefully analyze the behavior of quantization error over timesteps. To investigate this, we initially quantize the network using a MinMax quantizer and measure average quantization error across all layers. As illustrated in Fig. 4, the quantization error increases as denoising progresses (i.e., as the timestep t decreases). This observation raises a central question: *does a higher quantization error at a specific timestep correlate with a greater degradation of the final image quality?*

To answer this, we probed the diffusion model by injecting small random noise $\mathcal{U}(0, 0.1)$ at a specific timestep to activations of the quantized model and measuring the FID [15] of the generated images. As shown in Fig. 4, the final output quality does not directly correlate with the quantization error across timesteps. Specifically, while the later timesteps (smaller t) exhibit larger quantization errors and significantly degraded FID scores, the earlier timesteps (large t) also substantially affect the final image quality despite having smaller quantization errors. This is due to the fact that errors at early timesteps accumulate throughout the denoising process, ultimately impacting the final image quality. These findings highlight the necessity of carefully quantizing the model at early timesteps, even if the immediate quantization error appears small.

Adaptive timestep weighting. Applying a uniform loss across timesteps is suboptimal, as later steps typically exhibit larger errors, biasing optimization. Moreover, as shown in the Tab. 6, simple monotonic weighting that prioritizes early steps performs even worse. While the average error increases as t decreases (Fig. 4 left), individual layers exhibit varying trends, with some having minimal error even at later timesteps (Fig. 4 right). Thus, naively prioritizing early steps can overlook critical later steps in certain layers. To address this, we propose an adaptive timestep weighting that dynamically adjusts weights based on timesteps and the observed quantization error. Our adaptive loss function for each batch B is formulated as:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \lambda_{t_i} \mathcal{L}_i, \tag{7}$$

where λ_{t_i} adjusts the contribution of each sample based on its timestep t_i. The weighting factor λ_{t_i} is defined as:

$$\lambda_{t_i} = \left(1 - \frac{\Lambda_{t_i}}{\sum_{t' \in T} \Lambda_{t'}}\right)^{\alpha}, \tag{8}$$

where T is set of timesteps used for calibration and Λ_t is accumulated loss for the timestep t. This design is inspired by focal loss [34], which emphasizes harder-to-classify samples by assigning larger weights. Similarly, our approach prioritizes critical timesteps with lower accumulated errors.

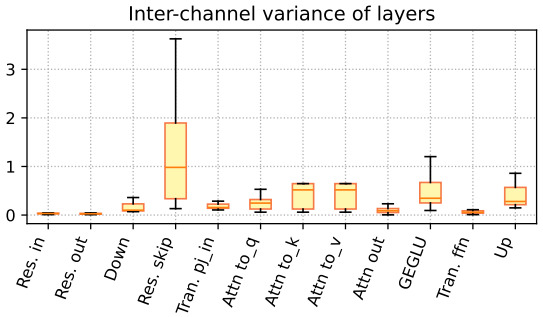


Figure 5. Inter-channel variance of the activations. *Res.* and *Tran.* indicates the residual block and the transformer layer, respectively.

Specifically, a higher α shifts optimization toward early denoising steps (Fig. 4 left), where small quantization errors can still significantly impact final output quality due to error accumulation. This adaptive weighting scheme balances both quantization error magnitude and its influence on the denoising trajectory. The accumulated loss Λ_t is updated iteratively as follows:

$$\Lambda_t \leftarrow \xi\Lambda_t + (1-\xi)\mathbb{E}_{i\in I_t}[\mathcal{L}_i], \tag{9}$$

where $I_t = \{i|t_i = t\}$ is the set of samples at timestep t. Using a moving average with momentum $\xi = 0.95$, we prevent abrupt changes in weights, ensuring stability while capturing shifts in quantization error throughout training.

Fusing τ with static parameters for efficient inference. Applying LES factors τ in Eq. (5) directly during inference introduces additional overhead due to the need for online computation. Whereas τ can be fused into the weight $\mathbf{W}$ since they are static, fusing τ directly with activation $\mathbf{X}$ is not feasible as they change dynamically. Previous works on LLMs [32, 59, 61] address this by fusing τ into previous linear operations (e.g., layer norm, linear layer). However, most diffusion models [16, 46] have a nonlinear operation [45] before matrix multiplication, preventing such fusion. To overcome this, we incorporate τ into the quantization process by using a modified scale for activation:

$$\tilde{\mathbf{X}} = \text{clamp}\left(\left\lfloor \frac{\mathbf{X}}{\{\tau \odot s^{(\mathbf{X})}\}} \right\rceil, l, u\right), \hat{\mathbf{X}} \approx s^{(\mathbf{X})} \cdot \tilde{\mathbf{X}} \tag{10}$$

$$\tilde{\mathbf{W}} = \text{clamp}\left(\left\lfloor \frac{\{\tau^\top \odot \mathbf{W}\}}{s^{(\mathbf{W})}} \right\rceil, l, u\right), \hat{\mathbf{W}} \approx s^{(\mathbf{W})} \cdot \tilde{\mathbf{W}}, \tag{11}$$

where $\{\cdot\}$ indicates pre-computed values. For weights, τ is fused directly into the weight values; in contrast, for activations, τ is integrated into the scale $s^{(\mathbf{X})}$ during quantization. During dequantization, both weights and activations use original scales $s^{(\mathbf{X})}$ and $s^{(\mathbf{W})}$, respectively. This is feasible as our approach employs static quantization, ensuring that $s^{(\mathbf{X})}$ remains fixed. This allows us to apply the scaling factors without additional overhead during inference.

4.2. Power-of-Two Scaling

Challenges with large outliers in activations. Upon examining activation distributions in the diffusion model, we found that some layers—especially skip connections lacking normalization—exhibit significant inter-channel variance due to extreme outliers (Fig. 5). Notably, while previous work [28] focused on bimodal distributions in UNet shortcut layers, our analysis shows that ResBlock skip connections suffer from even more severe outlier-induced variance. This high inter-channel variance poses substantial challenges for quantization. Although LES reduces quantization errors by redistributing difficulty between weights and activations, it alone cannot fully mitigate these extreme cases, which pose substantial challenges for quantization.

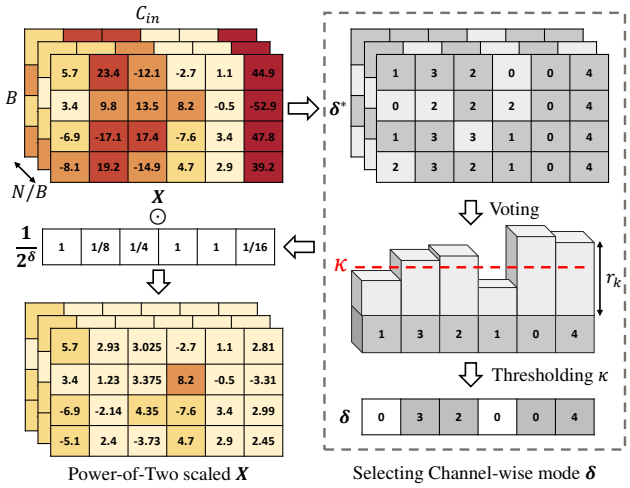


Figure 6. Illustration of voting algorithm for PTS factor δ.

Algorithm 1 Voting Algorithm for PTS Factor

Input: calibration data $\{\mathbf{X}_i\}_{i=1}^N$, number of candidate scales D, agreement threshold κ, per-tensor scale s

▷ *Candidate Selection by Per-Channel Quantization*

for sample index $i \in \{1, ..., N\}$ **do**

  for all channel index $k \in \{1, \ldots, C_{in}\}$ **do**

    $\delta^*_{i,k} \leftarrow \arg\min_{d\in\{0,\ldots,D\}} \left\|\mathbf{X}_{ik} - Q_{(s\cdot 2^d)}(\mathbf{X}_{ik})\right\|^2$

▷ *Robust Voting Mechanism by thresholding*

for all channel index $k \in \{1, \ldots, C_{in}\}$ **do**

  $\delta_k^{\text{mode}} = \text{mode}\left(\{\delta^*_{i,k}\}_{i=1}^N\right)$

  $r_k = \frac{1}{N}\sum_{i=1}^N \mathbf{1}\left\{\delta^*_{i,k} = \delta_k^{\text{mode}}\right\}$

  if $r_k > \kappa$ **then**

    $\delta_k \leftarrow \delta_k^{\text{mode}}$

  else

    $\delta_k \leftarrow 0$

return PTS factors $\delta = [\delta_1, \ldots, \delta_{C_{in}}]$

Power-of-Two quantization of activations. To address these challenges, we introduce a channel-wise Power-of-Two Scaling (PTS) for activations. This approach scales activations on a per-channel basis using power-of-two scaling factors, effectively adjusting the dynamic range of each channel individually. The scaling is formulated as:

Method	Bits (W/A)	FFHQ (LDM-4)		LSUN-Bedroom (LDM-4)		LSUN-Church (LDM-8)	
		FID↓	sFID↓	FID↓	sFID↓	FID↓	sFID↓
Full Prec.	32/32	31.34	25.88	9.82	20.17	11.28	29.58
Q-Diffusion [28]	8/8	33.10	27.69	14.79	27.36	12.87	30.68
PTQD [13]	8/8	31.87	25.91	12.90	18.56	12.73	30.00
EDA-DM [35]	8/8	31.28	31.25	9.87	20.29	12.31	29.82
TFMQ-DM [18]	8/8	32.35	26.78	10.05	21.12	11.25	**29.18**
Ours	8/8	**26.78**	**20.77**	**8.52**	**14.33**	**11.16**	29.35
Q-Diffusion [28]	4/8	36.17	28.75	14.87	27.60	15.42	33.02
PTQD [13]	4/8	36.65	32.52	19.27	20.59	14.52	33.03
EDA-DM [35]	4/8	36.28	31.07	16.96	31.64	**11.40**	**28.80**
TFMQ-DM [18]	4/8	36.08	33.06	10.65	18.99	14.28	34.62
Ours	4/8	**30.37**	**22.72**	**9.73**	**15.94**	11.58	30.87
Q-Diffusion [28]	4/6	71.16	75.70	64.31	35.09	57.34	58.58
PTQD [13]	4/6	74.18	80.51	106.21	39.44	56.90	57.19
EDA-DM [35]	4/6	29.62	28.59	75.21	49.18	14.41	32.79
TFMQ-DM [18]	4/6	29.76	27.07	17.03	34.27	26.74	57.51
Ours	4/6	**26.38**	**20.01**	**11.26**	**16.97**	**14.28**	**32.76**

Table 2. Quantization results of unconditional generation across various datasets at 256×256 resolution. Values in **boldface** indicate the best results, while underlined values indicate the second-best. W/A indicates the bit-width of the weights and activations, respectively.

$$\tilde{\mathbf{X}} = \text{clamp}\left(\left\lfloor \frac{\mathbf{X}}{2^{\boldsymbol{\delta}} \odot s^{(\mathbf{X})}} \right\rceil, l, u\right) \tag{12}$$

where $\boldsymbol{\delta} \in \mathbb{R}^{C_{in}}$ contains channel-wise power-of-two exponents, and $2^{\boldsymbol{\delta}} = [2^{\delta_1}, 2^{\delta_2}, \ldots, 2^{\delta_{C_{in}}}]$. Note that PTS applies only to activations, while weights follow Eq. (3). Integration of PTS into matrix multiplication is formulated as:

$$\begin{aligned} \mathbf{Y}_{ij} &\approx s_i^{(\mathbf{X})} s_j^{(\mathbf{W})} \left(\sum_{k=1}^{C_{in}} 2^{\delta_k} \tilde{\mathbf{X}}_{ik} \cdot \tilde{\mathbf{W}}_{kj} \right) \\ &= s_i^{(\mathbf{X})} s_j^{(\mathbf{W})} \left(\sum_{k=1}^{C_{in}} \tilde{\mathbf{X}}_{ik} \cdot \left(\tilde{\mathbf{W}}_{kj} \ll \boldsymbol{\delta}_k \right) \right), \end{aligned} \tag{13}$$

where $\ll$ denotes the left-bit-shift operation. Eq. (13) shows that multiplying by a PTS factor is equivalent to shifting the bits of the product, which is highly efficient on hardware [1, 9, 65, 66]. In practice, bit-shifting is applied immediately after loading weights at kernel execution (rather than during multiply-accumulation), ensuring minimal overhead.

Selecting an appropriate PTS factor $\boldsymbol{\delta}$ is crucial for minimizing quantization error. If $\boldsymbol{\delta}$ is too small, large salient activations are not sufficiently scaled down, pushing them outside the quantization range and causing severe clamping errors. In contrast, if $\boldsymbol{\delta}$ is too large, the activations are overly compressed, increasing the rounding errors. A straightforward approach is to select $\boldsymbol{\delta}$ values that minimize quantization error over the calibration set. However, we found that this degrades the model's performance on unseen data, making it unsuitable for PTQ, where only a small calibration set is available. To overcome this, we propose a robust voting algorithm for selecting the PTS factors, which remains effective even with a small calibration set.

Robust voting algorithm for scaling factor selection. As illustrated in Fig. 6, the voting algorithm for PTS factors comprises the following two major components.

- *Candidate selection by per-channel quantization*. For each calibration sample and channel, we evaluate candidate power-of-two scaling factors $\{2^0, 2^1, ..., 2^D\}$ and select the factor $\delta_{i,k}^*$ that minimizes the quantization error.
- *Robust voting mechanism by thresholding*. After determining the best scaling factors per sample and channel, we aggregate them across all samples by computing the mode δ_k^{mode} for each channel k. We then compute the agreement ratio r_k , the fraction of samples that select δ_k^{mode}. if $r_k > \kappa$, we assign $\delta_k = \delta_k^{\text{mode}}$; otherwise, we default to $\delta_k = 0$, meaning no scaling is applied.

This conservative approach prevents significant activation distortion by rejecting scaling factors with low agreement, thereby avoiding excessive clamping errors that arise when inappropriate PTS factors push activations beyond their typical quantization range. By requiring consensus, the method mitigates overfitting to anomalies in small calibration datasets and ensures robust scaling factor selection. Algorithm 1 summarizes the full procedure.

5. Experiments

Implementation details. We apply channel-wise quantization for weights and static tensor-wise quantization for activations. For a fair comparison, all methods quantize each layer except the input and output layers, which remain in full precision per common practice[10]. Equivalent scaling is optimized in a layer-wise manner, inspired by AdaRound [38]. After learning the equivalent scaling, we employ BRECQ [29] for weight quantization. The setup of

Full Precision Q-Diffusion TFMQ-DM Ours

Figure 7. Visualization of samples on ImageNet 256×256 generated by full precision LDM [46] and W4A6 quantized models using Q-Diffusion [28], TFMQ-DM [18], and ours. Unlike other methods that often struggle to maintain plausible outputs, the proposed method successfully preserves the high sample quality of full precision models, even under extremely low-bit quantization.

the calibration data follows that of Q-Diffusion [28], except for Stable Diffusion. For Stable Diffusion, we do not employ classifier-free guidance when sampling the calibration set following TAC [62]. Additionally, instead of quantizing time embedding layers, we cache their output values, which is more efficient and effective than direct quantization [55]. Additional details are provided in the appendix.

Models and datasets. We evaluate our method on both unconditional and conditional image generation models. For unconditional generation, we use the LDM-8 [46] model trained on the LSUN [67] Church 256×256, and the LDM-4 model trained on the LSUN Bedroom 256×256 and FFHQ [23] 256×256. For a conditional generation, we use the LDM-4 model trained on the ImageNet [7] 256×256 and the text-conditional model StableDiffusion v1.4.

Evaluation metrics. We evaluate the performance of quantized models using several standard metrics to evaluate image quality, diversity, and fidelity. For all settings, FID [15] and sFID [40] are used to evaluate the quality and diversity of generated images. In conditional generation tasks, we also include IS [48], LPIPS [70], SSIM, and PSNR to further evaluate the sample quality under given conditions and how closely the generated images align with those produced by the full-precision model, ensuring the generated output faithfully reproduces the intended content. For text-conditional generation, we report CLIP score [14] to quantify alignment between generated images and text prompts.

5.1. Main results

Unconditional image generation. We generated 50K samples using a DDIM sampler [54] with an eta of 1.0. To reflect real-world deployment scenarios [2, 4, 24, 64] where computational cost is critical, we used a small number of sampling steps—specifically, 20 steps. As shown in Tab. 2, our method consistently demonstrates superior performance across all settings and outperforms most existing methods. Notably, baseline methods which do not consider outliers suffer significant performance degradation at W4A6, whereas our approach maintains high FID and sFID. On the LSUN Church, our method achieves comparable results but is slightly outperformed by EDA-DM [35] in the W4A8 setting. However, EDA-DM exhibits inconsistent performance across other datasets. Since EDA-DM is orthogonal to ours, integrating their data distribution alignment with our method could potentially yield even better results. Overall, our method demonstrates superior performance across diverse datasets by successfully managing outliers.

Method	Bits (W/A)	ImageNet (LDM-4) IS↑	FID↓	sFID↓	LPIPS↓	SSIM↑	PSNR↑
Full Precision	32/32	366.8	11.34	7.81	–	–	–
Q-Diffusion [28]	8/8	351.3	11.11	12.20	0.116	0.895	26.88
PTQD† [13]	8/8	349.6	10.94	11.28	–	–	–
EDA-DM [35]	8/8	303.2	11.85	23.00	0.385	0.722	19.19
TFMQ-DM [18]	8/8	**362.7**	11.17	8.19	0.072	0.933	29.18
Ours	8/8	357.9	**10.29**	**7.38**	**0.058**	**0.945**	**31.14**
Q-Diffusion [28]	4/8	330.4	9.34	12.39	0.170	0.846	23.97
PTQD† [13]	4/8	321.7	9.04	13.09	–	–	–
EDA-DM [35]	4/8	312.9	9.86	14.22	0.237	0.803	22.37
TFMQ-DM [18]	4/8	342.1	**9.51**	8.10	0.181	0.848	23.30
Ours	4/8	**350.8**	9.68	**7.19**	**0.124**	**0.886**	**25.90**
Q-Diffusion [28]	4/6	65.2	36.49	71.41	0.477	0.613	17.09
PTQD† [13]	4/6	73.1	33.73	60.80	–	–	–
EDA-DM [35]	4/6	46.8	48.23	106.05	0.573	0.572	15.91
TFMQ-DM [18]	4/6	225.6	9.61	10.19	0.336	0.730	19.83
Ours	4/6	**320.6**	**7.81**	**7.26**	**0.194**	**0.845**	**23.70**

Table 3. Quantization results of class conditional generation. †: PTQD utilizes noise correction during the sampling, making it unable to compare pixel level similarities even under the same seed.

Class-conditional image generation. We evaluate class-conditional image generation on the ImageNet dataset using 50K images, employing 20 denoising steps with eta of 0 and a classifier-free guidance (CFG) scale of 3.0. As

Methods	Bits (W/A)	MS-COCO (Stable Diffusion)						
		IS↑	FID↓	sFID↓	CLIP↑	LPIPS↓	SSIM↑	PSNR↑
Full Precision	32/32	40.23	20.38	62.55	31.27	–	–	–
EDA-DM [35]	4/8	37.78	20.61	63.67	31.10	0.460	0.518	14.33
TFMQ-DM [18]	4/8	37.58	19.36	**60.94**	31.18	0.468	0.54	15.12
Ours	4/8	**40.16**	**19.20**	60.96	**31.41**	**0.443**	**0.557**	**15.34**
EDA-DM [35]	4/6	27.26	33.23	67.06	29.73	0.623	0.360	12.34
TFMQ-DM [18]	4/6	16.22	104.95	126.13	25.32	0.691	0.361	12.69
Ours	4/6	**33.10**	**29.58**	**67.05**	**30.67**	**0.537**	**0.427**	**14.05**

Table 4. Quantization results of text-guided image generation.

shown in Tab. 3, our method consistently outperforms existing methods across all settings. Notably, at the low bit-width configuration of W4A6, where prior methods suffer significant performance degradation, our approach maintains high performance. EDA-DM, which showed promising results on specific datasets in the unconditional setting, fails to generalize to the large-scale ImageNet dataset. Furthermore, our method achieves superior results in LPIPS, SSIM, and PSNR across all settings, indicating that our quantized model generates outputs closest to those of the full-precision model under the same conditions. This is further supported by qualitative comparisons in Fig. 7.

Text-guided image generation. In this experiment, we sampled text prompts from MS-COCO [33] to generate 10K images at 512×512 resolution, using 50 steps with eta of 0 and a CFG scale of 7.5. As reported in Tab. 4, similar to the unconditional and class-conditional generation, existing methods show significant performance degradation at W4A6, whereas our method demonstrates strong performance across nearly all metrics. Specifically, our method achieves the best LPIPS, SSIM, PSNR, and CLIP scores, indicating that the outputs not only closely match the full-precision model in perceptual and structural similarity but also maintain strong semantic alignment with the given text.

5.2. Ablation study

We conducted an ablation study on LDM-4 on FFHQ at W4A8 to verify each component in our proposed method.

Analysis of the proposed method. Tab. 5 presents the impact of progressively adding each component of our method. Starting from the baseline, introducing Learned Equivalent Scaling redistributes quantization difficulty caused by outliers, resulting in substantial metric improvements. Adding *Adaptive Timestep Weighting* further enhances performance by prioritizing early denoising steps, which, despite lower quantization errors, significantly impact final quality due to error accumulation. Finally, applying PTS to skip connection layers with high inter-channel variance yields the best overall results.

Effect of adaptive timestep weighting. Tab. 6 shows the effectiveness of adaptive timestep weighting. Simple heuristics, such as linearly or quadratically increasing weights for early timesteps, underperform uniform weighting. While early steps have smaller errors but a strong impact, later steps with larger errors are equally important. A fixed weighting scheme may overlook critical steps, as some layers exhibit small errors in later timesteps. In contrast, our adaptive strategy dynamically adjusts to these variations, achieving superior performance.

Method	Bits (W/A)	IS↑	FID↓	sFID↓
Full Precision	32/32	3.70	31.34	25.88
Baseline	4/8	3.50	36.08	33.06
+ Learned Equivalent Scaling	4/8	3.60	33.46	26.29
+ Adaptive Timestep Weighting	4/8	3.69	31.83	24.39
+ Power-of-Two Scaling	4/8	**3.75**	**30.37**	**22.72**

Table 5. Ablation study on components of the proposed method.

Method	Bits (W/A)	IS↑	FID↓	sFID↓
Full Precision	32/32	3.70	31.34	25.88
Uniform	4/8	3.60	33.46	26.29
Linear	4/8	3.59	36.58	30.20
Quadratic	4/8	3.61	35.30	27.39
Adaptive Timestep Weighting	4/8	**3.69**	**31.83**	**24.39**

Table 6. Ablation study on timestep weighting strategy.

Method	Target layers	Bits (W/A)	IS↑	FID↓	sFID↓
Full Precision		32/32	3.70	31.34	25.88
w/o PTS	None	4/8	3.69	31.83	24.39
MSE	all layers	4/8	3.74	32.55	24.24
Voting	all layers	4/8	3.72	31.91	23.86
Voting	skip connect.	4/8	**3.75**	**30.37**	**22.72**

Table 7. Ablation study on PTS scaling.

Effect of the voting algorithm. Tab. 7 demonstrates the effectiveness of our voting algorithm for selecting PTS factors. The MSE-based approach, which directly minimizes quantization error, tends to overfit on a calibration set, leading to suboptimal performance. In contrast, our voting algorithm finds robust PTS factors by aggregating statistical consensus across samples, resulting in better generalization. Moreover, applying PTS only to skip connection layers, rather than all layers, proves to be more effective due to the high inter-channel variance present specifically in these layers. This targeted application further improves performance, as evidenced by the superior metrics achieved.

6. Conclusion

We present a new approach that addresses the outlier problem in diffusion model quantization. By integrating Learned Equivalent Scaling with an Adaptive Timestep Weighting, we redistribute quantization difficulty between weights and activations and prioritize critical timesteps. Additionally, we propose a Power-of-Two Scaling, accompanied by a robust voting algorithm, to handle layers with extreme outliers, such as skip connections. Extensive experiments in various datasets and generation tasks demonstrate that our method consistently outperforms existing post-training quantization techniques, even under ultra-low bit-width like W4A6, where prior methods often fail.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00439020, Developing Sustainable, Real-Time Generative AI for Multimodal Interaction, SW Starlab).

References

[1] Giorgos Armeniakos, Georgios Zervakis, Dimitrios Soudris, and Jörg Henkel. Hardware approximate techniques for deep neural network accelerators: A survey. *ACM Computing Surveys*, 55(4):1–36, 2022. 6

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 7

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 1

[4] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 7

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 2

[6] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[7] 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. 7

[8] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022. 3

[9] Mostafa Elhoushi, Zihao Chen, Farhan Shafiq, Ye Henry Tian, and Joey Yiwei Li. Deepshift: Towards multiplication-less neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2359–2368, 2021. 6

[10] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 2, 6

[11] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022. 1

[12] Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Efficientdm: Efficient quantization-aware fine-tuning of low-bit diffusion models. In *International Conference on Learning Representations*, 2024. 2

[13] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 6, 7

[14] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 7

[15] 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. *Advances in neural information processing systems*, 30, 2017. 4, 7

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3, 5

[17] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 1

[18] Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7362–7371, 2024. 1, 2, 3, 4, 6, 7, 8

[19] Jiwan Hur, Jaehyun Choi, Gyojin Han, Dong-Jae Lee, and Junmo Kim. Expanding expressiveness of diffusion models with limited data via self-distillation based fine-tuning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5028–5037, 2024. 1

[20] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022. 1

[21] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[22] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4350–4359, 2019. 2

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 7

[24] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024. 7

[25] Jun Haeng Lee, Sangwon Ha, Saerom Choi, Won-Jo Lee, and Seungwon Lee. Quantization for rapid deployment of deep neural networks. *arXiv preprint arXiv:1810.05488*, 2018. 2

[26] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 1

[27] Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. *arXiv preprint arXiv:2305.15583*, 2023. 3

[28] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023. 1, 2, 3, 4, 5, 6, 7

[29] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. 2, 6

[30] Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. In *Advances in Neural Information Processing Systems*, pages 76680–76691. Curran Associates, Inc., 2023. 2

[31] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 1

[32] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024. 2, 3, 5

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 8

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[35] Xuewen Liu, Zhikai Li, Junrui Xiao, and Qingyi Gu. Enhanced distribution alignment for post-training quantization of diffusion models. *arXiv preprint arXiv:2401.04585*, 2024. 1, 2, 3, 6, 7, 8

[36] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 1

[37] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[38] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020. 2, 6

[39] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. 1, 4

[40] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 7

[41] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1, 3

[42] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. *arXiv preprint arXiv:2308.15321*, 2023. 3

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 3

[44] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[45] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7(1):5, 2017. 5

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 5, 7

[47] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 1

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 7

[49] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1972–1981, 2023. 1, 2, 3

[50] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024. 1

[51] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1

[52] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3

[53] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 3

[54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 7

[55] Yang Sui, Yanyu Li, Anil Kag, Yerlan Idelbayev, Junli Cao, Ju Hu, Dhritiman Sagar, Bo Yuan, Sergey Tulyakov, and Jian Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. *arXiv preprint arXiv:2406.04333*, 2024. 7

[56] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 3

[57] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate post-training quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16026–16035, 2024. 1, 2, 3

[58] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414, 2022. 3

[59] Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*, 2023. 2, 3, 5

[60] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. In *NeurIPS*, 2024. 3

[61] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023. 2, 3, 5

[62] Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. *arXiv preprint arXiv:2407.03917*, 2024. 1, 2, 3, 7

[63] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022. 3

[64] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 7

[65] Haoran You, Xiaohan Chen, Yongan Zhang, Chaojian Li, Sicheng Li, Zihao Liu, Zhangyang Wang, and Yingyan Lin. Shiftaddnet: A hardware-inspired deep network. *Advances in Neural Information Processing Systems*, 33:2771–2783, 2020. 6

[66] Haoran You, Yipin Guo, Yichao Fu, Wei Zhou, Huihong Shi, Xiaofan Zhang, Souvik Kundu, Amir Yazdanbakhsh, and Yingyan Celine Lin. Shiftaddllm: Accelerating pretrained llms via post-training multiplication-less reparameterization. *arXiv preprint arXiv:2406.05981*, 2024. 6

[67] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 7

[68] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023. 1

[69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[71] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024. 3