SyRA-DiT: Synergistic Reconstruction-Aware Rotation for W4A4 Diffusion Transformers

Anonymous authors
Paper under double-blind review

Abstract

Post-training quantization (PTQ) enables low-precision deployment of pretrained Diffusion Transformers without retraining, yet aggressive W4A4 quantization remains challenging. Coarsely quantized weights are reused across denoising timesteps, while activation distributions vary over time, making layer reconstruction error difficult to minimize consistently throughout the denoising process. Group-wise quantization improves local range adaptation, but but its quantization fidelity remains highly sensitive to the within-group distribution. To this end, we propose SyRA-DiT, a reconstruction-driven PTQ framework that learns an orthogonal rotation to jointly reorganize weight and activation quantization groups across denoising timesteps, directly minimizing W4A4 layer reconstruction error on a limited calibration set via Cayley parameterization and straight-through estimation. Furthermore, SyRA-DiT uses the same per-timestep incoherence weights to construct the initialization covariance, the W4A4 reconstruction objective, and the GPTQ Hessian. The rotation is optimized to minimize the relative W4A4 reconstruction error, after which GPTQ quantizes the rotated weights using a Hessian constructed under the same temporal weighting. We evaluate SyRA-DiT on image, text-to-image, and video diffusion models under 4-bit weight and activation quantization, where it achieves state-of-the-art visual quality and quantitative performance. Notably, on ImageNet at 256×256 resolution with DiT-XL/2, SyRA-DiT remains competitive with previously reported W4A8 results across diverse sampling and guidance settings, demonstrating that PTQ remains effective even in the challenging W4A4 regime.

1 Introduction

Diffusion Transformers (DiTs) Peebles & Xie (2023) have emerged as a leading backbone for high-fidelity image and video generation, offering a Transformer-based alternative to convolutional U-Nets Ronneberger et al. (2015). This scalability comes at substantial inference cost, as attention and feed-forward layers incur rapidly growing computation and memory demands with model size, resolution, and sequence length Feng et al. (2025). Efficient deployment therefore calls for compressing DiTs without costly retraining Liu et al. (2021); Yang et al. (2026a); Lee et al. (2026).

Post-training quantization (PTQ) is attractive for diffusion models because it reduces storage and memory bandwidth while enabling low-bit computation using only a small calibration set Liu et al. (2021); Zhao et al. (2024); Chen et al. (2025); Xu et al. (2026). Existing methods improve PTQ through timestep-aware calibration and reconstruction, per-step activation scaling, channel grouping, and reparameterization via equalization or absorbable rotations. While effective at W8A8 and W4A8 Zhao et al. (2024), **these approaches degrade sharply when both weights and activations are quantized to 4 bits** Wang et al. (2025). Existing methods address quantization error through different surrogate criteria and reparameterization strategies: PTQ4DM Shang et al. (2023) and Q-Diffusion Li et al. (2023a) adopt BRECQ-style Li et al. (2021) local reconstruction over timestep-aware calibration data; TFMQ-DM aligns temporal embeddings Huang et al. (2024a); Q-DiT uses FID-guided group allocation Chen et al. (2025); PTQ4DiT Wu et al. (2024) and DiTAS Dong & Zhang (2025) balance channel-wise salience or activation peaks; and ViDiT-Q Zhao et al. (2024), HQ-DiT Liu & Zhang (2024), and VETA-DiT Xu et al. (2026) suppress outliers through absorbable rotations or Hadamard transforms. However, these strategies do not directly account for the jointly quantized computation of a DiT layer. At each timestep, the output error under group-wise

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

quantization is jointly governed by the *dynamic* activation quantization and a shared *static* weight quantization across the denoising trajectory Moon et al. (2024); Zhao et al. (2024); Wang et al. (2026). This creates an objective mismatch: existing criteria do not explicitly minimize the reconstruction error of the jointly quantized product across timesteps, leaving substantial residual error under W4A4. To address above challenging issues, we propose SyRA-DiT, a reconstruction-driven PTQ framework for fully 4-bit DiTs. Our contributions are threefold:

- First, we show that separately optimizing weight and activation quantization overlooks their coupled effects in DiTs. While weight quantization is static, activation quantization varies across denoising timesteps, and their interaction in the layer product can induce reconstruction error and outliers beyond those captured by optimizing either side in isolation.
- Second, we propose SyRA-DiT, a reconstruction-driven PTQ framework that learns a unified orthogonal rotation to jointly account for static weight and dynamic activation quantization. A shared timestep-aware incoherence mechanism further unifies rotation initialization, layer reconstruction, and GPTQ Hessian estimation, enabling robust W4A4 quantization by suppressing time-varying outliers.
- Third, extensive experiments across image, text-to-image, and video diffusion models demonstrate state-of-the-art W4A4 performance, competitive accuracy with prior W4A8 baselines, and practical efficiency gains, including lower memory usage and up to $1.74\times$ speedup over FP16 with CUDA graph execution.

2 Related Work

2.1 Low-Resolution DiT for Image and Video Generation

DiTs Peebles & Xie (2023) replace convolutional U-Net denoisers with Transformer backbones, enabling global token interactions and favorable scaling with model capacity. They have been widely adopted in image, text-to-image, and video generation, including PixArtChen et al. (2024), Latte Ma et al. (2024b), and Open-Sora Zheng et al. (2024), with video models extending DiTs to jointly capture spatial and temporal dependencies. However, higher resolutions, longer sequences, and iterative denoising substantially increase computation, memory, and bandwidth costs. Quantization reduces storage and inference overhead without architectural changes, with existing methods mainly divided into quantization-aware training (QAT) So et al. (2023); Zeng et al. (2025); Taniguchi et al. (2026) and PTQ. QAT better preserves accuracy at low bit-widths but requires costly retraining and training data, whereas PTQ quantizes pretrained models using only a small calibration set, making it more practical despite larger degradation under aggressive quantization. Given the high retraining cost of large-scale DiTs, we focus on PTQ as a more practical approach for efficient deployment.

2.2 Quantization Methods

DiTs are particularly challenging to quantize because activation distributions vary across denoising timesteps and both weights and activations exhibit structured channel-wise outliers Li et al. (2023b). Existing PTQ methods address these outliers along four directions. (1) *Granularity-based methods* isolate outliers within finer groups Shao et al. (2026); Wang et al. (2026); Li et al. (2026): Q-DiT Chen et al. (2025) performs layer-wise search over group sizes for input-channel weight quantization, while Shen et al. (2020); Moon et al. (2024) combines weight grouping with the calibration to improve local adaptation at the cost of additional overhead. (2) *Channel-equalization methods* rescale activation outliers into weights using timestep-aggregated statistics, as in the Spearman's ρ-weighted balancing of PTQ4DiT Wu et al. (2024) and temporal smoothing of DiTAS Dong & Zhang (2025), but largely redistribute rather than remove outliers. (3) *Transformation-based methods* Huang et al. (2025a); Yang et al. (2026b); Wang et al. (2026) use orthogonal rotations to spread outlier energy across channels: ViDiT-Q Zhao et al. (2024) combines scaling- and rotation-based static–dynamic channel balancing, while HQ-DiT Liu & Zhang (2024) combines Hadamard transforms with adaptive FP4 quantization. (4) *Decomposition-based methods* Sharify et al. (2026) separate outliers from the low-bit path; SVDQuant Li et al. (2024), for example, absorbs them into a high-precision low-rank branch and quantizes the residual to 4 bits with a co-designed inference engine. Despite this progression from isolating and redistributing outliers to explicitly separating them, preserving generation quality under W4A4 and lower precision remains challenging.

2.3 LOW-BIT LLM QUANTIZATION

Low-bit LLM quantization has developed several techniques later adapted to DiTs. (1) *Error-compensating and salience-aware methods*, such as GPTQ Frantar et al. (2022) and AWQ Lin et al. (2024), enable 3–4-bit weight quantization through second-order compensation or activation-aware scaling. (2) *Outlier-handling methods*, including LLM.int8() Dettmers et al. (2022), SmoothQuant Xiao et al. (2023), and OmniQuant Shao et al. (2024), mitigate activation outliers through mixed precision or activation-to-weight migration. (3) *Rotation-based methods*, such as QuaRot Ashkboos et al. (2024) and SpinQuant Liu et al. (2025), use orthogonal transformations to suppress outliers and enable W4A4 quantization. (4) *Vector and codebook methods*, including QuIP# Tseng et al. (2024) and AQLM Egiazarian et al. (2024), target sub-3-bit precision, while more extreme regimes rely on training-based approaches such as LLM-QAT Liu et al. (2024) and BitNet Ma et al. (2024a). These ideas have also influenced DiT quantization Li et al. (2023a); Zhao et al. (2024); Wu et al. (2024); Liu & Zhang (2024); Xu et al. (2026), particularly smoothing and Hadamard rotation. These advances provide useful insights into module-wise quantization. Drawing on these insights, we introduce DiT-specific techniques to reduce time-varying joint weight–activation quantization error.

3 PRELIMINARIES

3.1 QUANTIZATION

Consider a DiT operating over reverse diffusion timesteps $t \in \{1, \dots, T\}$. For a given linear layer at timestep t, let $X_t \in \mathbb{R}^{N \times C}$ denote its input activations, where N and C are the numbers of tokens and channels, respectively. Given $W \in \mathbb{R}^{C_{\text{out}} \times C}$ and $b \in \mathbb{R}^{C_{\text{out}}}$, the full-precision output is

$$Y_t = X_t W^\top + b. \tag{1}$$

Thus, the layer parameters (W) are fixed across steps, whereas X_t follows timestep-dependent activation distributions. We quantize activations and weights using k-bit asymmetric uniform quantization Yang et al. (2019); Nagel et al. (2020). Each row is partitioned along the channel dimension into non-overlapping groups of g consecutive channels, and each group is quantized independently. For an input group $u \in \mathbb{R}^g$, the quantization operator Q maps u to an integer-valued representation:

$$Q(u; s, z, k) = \text{clamp}\left(\text{round}\left(\frac{u}{s}\right) + z, 0, 2^k - 1\right). \tag{2}$$

Here, s and z denote the group-wise scale and zero-point, respectively; $\text{round}(\cdot)$ denotes rounding to the nearest integer; and $\text{clamp}(x, q_{min}, q_{max})$ clips x element-wise to the interval $[q_{min}, q_{max}]$. In the layer products below, Q_a and Q_w denote dequantized values of the form $s(q - z)$, where q is the integer code produced by the corresponding group-wise activation or weight quantizer, rather than an integer-valued operand. Activation quantization Q_a is performed dynamically: at each reverse-process step t, every token in X_t is partitioned into groups of size g, with a separate scale and zero-point (s, z) computed from the current activations for each group. In contrast, weight quantization Q_w is performed offline and remains fixed throughout inference. We quantize W with GPTQ Frantar et al. (2022) using calibration activations and reuse the resulting quantized weights across all reverse-process steps. The quantized layer output is

$$\hat{Y}_t = Q_a(X_t) Q_w(W^\top) + b. \tag{3}$$

In the work, we focus on W4A4 quantization, with 4-bit weights and activations.

3.2 RECONSTRUCTION LOSS

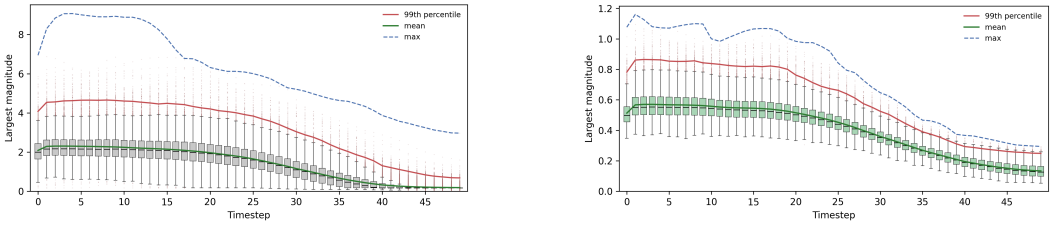
PTQ methods Frantar et al. (2022); Wu et al. (2024) for diffusion models typically retain layer- or block-wise reconstruction while refining timestep calibration, temporal alignment, activation scaling, or sampling-time error correction. For DiTs, recent work Zeng et al. (2025) further explores group-wise quantization, channel smoothing, and mixed-precision designs. To isolate weight and activation quantization effects, we adopt the standard group-wise reconstruction objective:

$$\mathcal{L}_{\text{rec}} = \frac{1}{TNC_{\text{out}}} \sum_{t=1}^{T} \left\| X_t W^\top - Q_a(X_t) Q_w(W)^\top \right\|_F^2. \tag{4}$$

This loss provides a flexible basis for designs that account for weight and activation quantization errors, their interactions, and temporal variations across timesteps.

4 SyRA-DiT: Synergistic PTQ for W4A4 DiTs

4.1 Sources of W4A4 Quantization Error in DiTs



(a) No Rotation (b) Ours

Figure 1: Boxplot of largest magnitude of activation values at different timesteps. Our method significantly reduce the outlier.

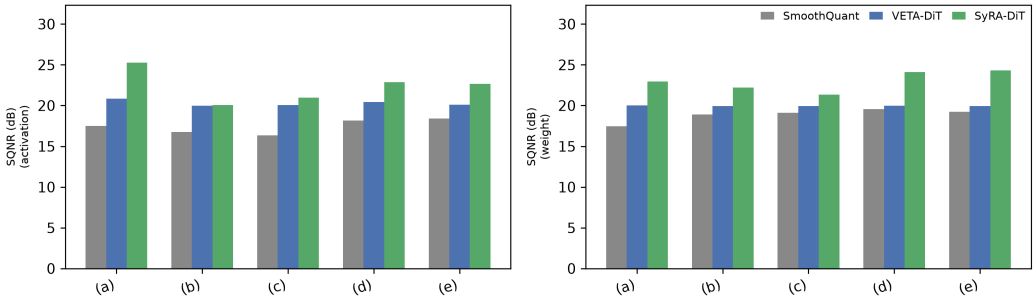


Figure 2: Layer-wise SQNR comparison of SmoothQuant (No), VETA-DiT (Fixed), and SyRA-DiT (Learnable) on representative DiT modules: (a) b2.attn.qkv, (b) b8.mlp.fc2, (c) b14.mlp.fc2, (d) b20.attn.proj, and (e) b26.attn.proj. SyRA-DiT consistently improves both weight and activation SQNR across layers and module types.

Under aggressive W4A4 quantization, we begin by showing that existing PTQ pipelines only partially control this error, primarily in the following two respects. First, existing approaches primarily optimize activation-side proxies, such as channel ranges, outlier magnitudes, or covariance structure, typically through rescaling. While these objectives can alleviate outliers in weights or activations, they largely overlook the distinct quantization behaviors of the two operands: **weight quantization is static, whereas activation quantization varies dynamically across denoising steps. Moreover, their interaction ($Q_a(X_t)Q_w(W^\top)$) in the quantized layer product can introduce additional reconstruction error and even new outliers that are not captured by optimizing either side in isolation**. This motivates a unified treatment of weights and activations that explicitly accounts for both their static–dynamic discrepancy and their interaction during quantized reconstruction, as realized by our learnable rotation. Figure 2 reports the resulting module-wise signal-to-quantization-noise ratio (SQNR) Huang et al. (2025b), measured separately for weight and activation reconstruction errors.

Second, the **time-varying data distribution** Shang et al. (2023); Zhao et al. (2024); Chen et al. (2025); Xu et al. (2026) presents a critical challenge during calibration and substantially affects key components of our method, including **outlier suppression, reconstruction, and GPTQ**. Inspired by incoherence processing Chee et al. (2023); Tseng et al. (2024), we adopt it to mitigate activation outliers. More importantly, we find both empirically and theoretically that the temporal structure of activations can be obscured during calibration. In particular, aggregating activations across timesteps when estimating covariance matrices or GPTQ statistics may fail to preserve the timestep-specific distributions encountered at inference time. Furthermore, if different temporal mixtures are used to determine the transformation and to calibrate weight rounding, the resulting 4-bit weights may become misaligned with the intended reconstruction objective. A detailed analysis is provided in

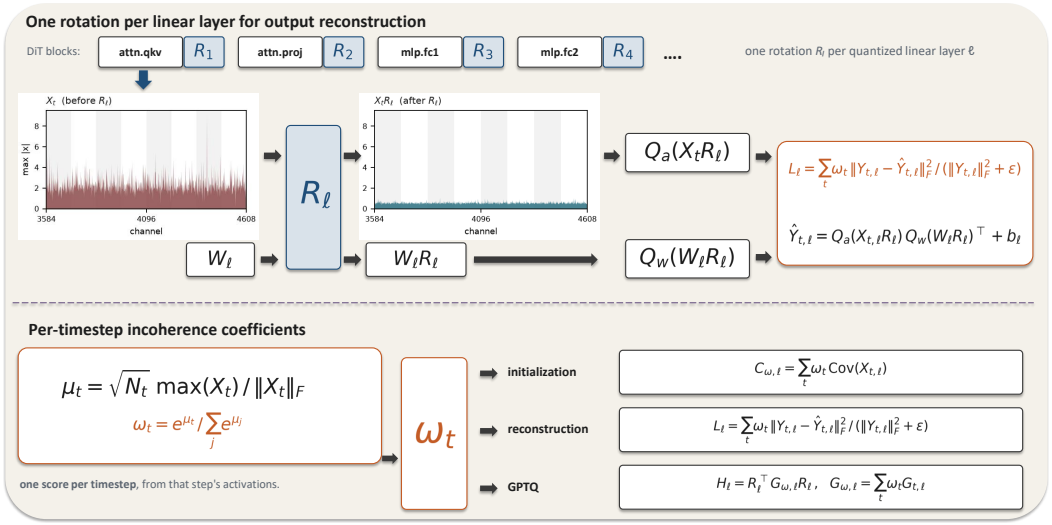


Figure 3: Overview of SyRA-DiT. Within each denoising step, joint rotation minimizes reconstruction error by jointly accounting for the static weight distribution and time-varying activation distribution, leading to stronger outlier suppression. Across steps, per-timestep incoherence coefficients further adapt to activation dynamics. The upper-right plot verifies the reduction of outliers, while the lower-right plot demonstrates strong W4A4 quantization performance.

Appendix A.1. Figure 1a reveals substantial temporal variation in the activation distributions, while Figure 1b demonstrates that SyRA-DiT consistently suppresses extreme activation magnitudes.

4.2 Synergistic Designs

To address these challenges, as shown in Figure 3, SyRA-DiT integrates three synergistic components: a shared orthogonal transformation for joint weight–activation quantization, a unified incoherent coefficients across diffusion timesteps, and GPTQ calibration within the transformed basis. Together, these components align the quantization basis, reconstruction objective, and weight rounding along a common denoising trajectory.

Shared orthogonal transformation. For a linear layer with input dimension C, we introduce $R \in \mathbb{R}^{C\times C}$ with $R^\top R = RR^\top = I$ and apply the same transformation to both activations and weights:

$$\hat{Y}_t(R) = Q_a(X_t R)\, Q_w(WR)^\top + b. \tag{5}$$

This transformation aims to preserve the full-precision function, with $(X_t R)(WR)^\top \approx X_t W^\top$. In particular, R transforms the channel coordinates and consequently the per-group dynamic ranges observed by both Q_a and Q_w. We therefore optimize a single orthogonal transformation directly against the joint W4A4 output reconstruction error, rather than relying solely on an activation-side or weight-side proxy. This design differs from learnable rotation methods in LLMs Ashkboos et al. (2024); Liu et al. (2025), which learn distinct rotations across different network modules rather than across diffusion timesteps. See Appendix B.2 for additional discussion.

Per-timestep incoherent coefficient. We adopt a uniform sampling distribution over the reverse-process steps during calibration Chee et al. (2023); Tseng et al. (2024). To prioritize timesteps exhibiting pronounced activation outliers, we define a peak-to-root-mean-square (RMS) score:

$$\mu_t = \frac{\sqrt{NC}\ \max_{ij} |(X_t)_{ij}|}{\|X_t\|_F}, \qquad \omega_t = \frac{\exp(\mu_t)}{\sum_{j=1}^{T} \exp(\mu_j)}, \tag{6}$$

where $\omega_t > 0$ and $\sum_{t=1}^{T} \omega_t = 1$. μ_t quantifies the severity of the activation peak relative to the RMS magnitude, with larger values assigning greater calibration weight to timestep t.

Trajectory-aware initialization. We initialize the rotation matrix using per-step activation statistics while preserving the temporal structure of the calibration trajectory. Let $\Sigma_t \in \mathbb{R}^{C\times C}$ denote the channel-wise covariance matrix computed across the N token rows of $X_t \in \mathbb{R}^{N\times C}$, and let $G_t = N^{-1}X_t^\top X_t$ denote its corresponding second-moment matrix. Using the timestep weights in Eq. 6, we form

$$\Sigma_\omega = \sum_{t=1}^{T} \omega_t \Sigma_t, \qquad G_\omega = \sum_{t=1}^{T} \omega_t G_t. \tag{7}$$

Let $\Sigma_\omega = U\Lambda U^\top$ denote its eigendecomposition, and let $H \in \mathbb{R}^{C\times C}$ be a normalized randomized Hadamard matrix Ashkboos et al. (2024); Zhao et al. (2024) satisfying $H^\top H = I$. We initialize from UH and parameterize a learnable residual rotation as

$$R(A) = UH\,\mathrm{Cayley}(A - A^\top), \tag{8}$$

where $A \in \mathbb{R}^{C\times C}$ and $\mathrm{Cayley}(S) = (I-S)(I+S)^{-1}$ denotes the Cayley transform Li et al. (2020). Because $A - A^\top$ is skew-symmetric, the Cayley transform yields an orthogonal matrix; thus, $R(A)$ remains strictly orthogonal throughout optimization, initialized at $R(A_0) = UH$. In contrast to calibration methods that concatenate or average activations across timesteps Xu et al. (2026), Eq. 7 forms the temporal mixture from per timestep statistics. This design retains the distinct contribution of each denoising step, yielding a unified basis shared across the entire trajectory.

Joint W4A4 reconstruction. Starting from the trajectory-aware initialization, we refine R directly with respect to the quantized layer output. With the pretrained weights fixed, we optimize

$$A^\star = \arg\min_{A\in\mathbb{R}^{C\times C}} \sum_{t=1}^{T} \omega_t \frac{\left\| Y_t - \hat{Y}_t(R(A)) \right\|_F^2}{\|Y_t\|_F^2 + \varepsilon}, \tag{9}$$

where $\varepsilon > 0$ is a small constant for numerical stability. During refinement, both the activation quantizer Q_a and weight quantizer Q_w use 4-bit asymmetric group-wise min–max fake quantization with group size $g = 128$. For each group, we set the quantization range from its minimum and maximum values, expanded when necessary to include zero, and map this range to 16 integer levels using an asymmetric scale and zero-point. Values are rounded to the nearest integer, clamped to $[0, 15]$, and subsequently dequantized with the corresponding group parameters. For activations, the quantization parameters are recomputed for each token group at every timestep. For weights, they are recomputed from the current rotated weights WR as R is updated, while the underlying pretrained weights W remain fixed. We approximate gradients through the rounding operations in both Q_a and Q_w using a straight-through estimator (STE). The resulting fake-quantized weights therefore provide a differentiable proxy for the distortion induced by subsequent W4 quantization, but need not coincide with the final GPTQ weights, which are obtained afterward through sequential error compensation. Because Eq. 9 evaluates the quantized matrix product directly, the learned rotation jointly adapts to activation- and weight-quantization errors across timesteps.

GPTQ calibration under the same temporal weighting. After optimizing $A^\star$, we fix $R^\star = R(A^\star)$ and apply GPTQ to $WR^\star$. Using the same timestep weights ω_t, the full-precision input second-moment matrix in the rotated basis is

$$G_\omega^R = {R^\star}^\top G_\omega R^\star, \tag{10}$$

which is proportional to the quadratic reconstruction Hessian used by GPTQ Frantar et al. (2022). This calibration statistic uses the same temporal weighting ω_t as the rotation initialization and STE refinement, but the corresponding objectives are not identical. In particular, Eq. 9 additionally normalizes each timestep by $\|Y_t\|_F^2+\varepsilon$ and evaluates reconstruction with quantized activations, whereas Eq. 10 is constructed from unnormalized full-precision input second moments. We therefore use G_ω^R as a temporally matched second-order approximation for the final GPTQ weight reconstruction, rather than as the exact Hessian of Eq. 9. The resulting GPTQ-quantized weight matrix is stored once and reused across all reverse-process steps.

Further analysis, including detailed algorithmic steps, evaluation of alternative rotation parameterizations, and additional interpretability results on activation distributions and outlier reduction across channels and timesteps, is provided in Appendices A.1–B.3.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Experimental setup. We evaluate the proposed SyRA-DiT on three representative generation tasks:image generation Russakovsky et al. (2015), text-to-image generation Lin et al. (2014), and video generation Zheng et al. (2024).

For class-conditional generation Peebles & Xie (2023), we use the official DiT-XL/2 checkpoint Peebles & Xie (2023) on ImageNet-1K at 256×256 resolution Russakovsky et al. (2015) and sample with DDIM Song et al. (2020) under multiple configurations that vary the number of sampling steps and classifier-free guidance (CFG) scale. Following prior evaluation protocols, we generate 10 images per class across all 1,000 classes, yielding 10K images in total, and report Fréchet Inception Distance (FID) Heusel et al. (2017), sFID, Inception Score (IS) Salimans et al. (2016), and Precision against the ImageNet-256 reference set.

For text-to-image generation, we use PixArt-ΣChen et al. (2024) at 1024×1024 resolution with DPM-Solver for 20 steps and CFG $= 4.5$. Evaluation uses MS-COCO val2014 captions, with FID and sFID computed against the full validation set of 40,504 images. To further characterize quantization-induced deviation from the full-precision (FP16) model, following Laidlaw et al. (2020); Zhao et al. (2024), we report relative FID (RFID), relative sFID (rsFID), Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018), and CLIP-based Maximum Mean Discrepancy (CMMD) Jayasumana et al. (2024) with respect to the corresponding FP16 generations.

For the video generation, we use Open-Sora v1.2 with STDiT3-XL/2 Zheng et al. (2024) and rectified-flow sampling Liu et al. (2022) with timestep transformation and CFG $= 7$. We evaluate on 946 VBench v0.1.2 prompts, generating landscape 240×426 videos of approximately 2 seconds (51 frames at 24 fps) using 30 rectified-flow steps and a motion score of 2, and report eight major VBench dimensions Huang et al. (2024b). We additionally generate 10 clips at 720p for qualitative comparison. For all three tasks, we follow the calibration protocols adopted in prior PTQ works. We compare against Q-DiT Chen et al. (2025), ViDiT-Q Zhao et al. (2024), and VETA-DiT Xu et al. (2026), reproducing all baselines from their official implementations, and include W4A8 results where applicable. Since ViDiT-Q is primarily designed for W4A8, W8A8, and mixed-precision quantization, we adapt its released implementation to the W4A4 setting without introducing additional W4A4-specific tuning. Further optimization tailored to W4A4 may improve its performance, but such tuning is beyond the scope of this work.

5.2 MAIN RESULTS

Image Generation on ImageNet. Table 1 compares SyRA-DiT with existing PTQ methods for DiT-XL/2 on ImageNet 256×256 across different sampling steps and CFG scales. SyRA-DiT consistently delivers the strongest overall W4A4 performance, with substantial gains over PTQ4DiT and Q-DiT and clear improvements over ViDiT-Q across IS, FID, and precision. Compared with the stronger VETA-DiT baseline, SyRA-DiT further improves IS and FID in all settings while maintaining comparable sFID. Both SyRA-DiT and VETA-DiT remain stable when reducing the sampling budget from 100 to 50 steps, while SyRA-DiT consistently preserves its performance advantage. These results validate the effectiveness of our reconstruction-driven learned rotation in jointly reorganizing weight and activation quantization groups across denoising timesteps, enabling accurate W4A4 quantization with only a small calibration set. Notably, SyRA-DiT remains competitive with previously reported W4A8 results Xu et al. (2026) across diverse sampling and guidance settings, demonstrating that PTQ remains effective even in the challenging W4A4 regime. The qualitative comparisons are provided in Figure 9 in Appendix B.5.

Text-to-Image Generation on COCO. Figure 4 further evaluates the fidelity of W4A4 quantization against the FP16 model using perceptual and distributional metrics. SyRA-DiT achieves the best results across all metrics, reducing RFID, rsFID, LPIPS, and CMMD from 42.7, 247.8, 0.420, and 0.139 with VETA-DiT to 35.6, 222.4, 0.332, and 0.103, respectively. It also substantially outperforms Q-DiT and ViDiT-Q, whose outputs exhibit much larger deviations from the FP16 reference. The qualitative comparison is consistent with these metrics: SyRA-DiT better preserves the overall composition, subject appearance, and fine visual details of the FP16 generation, while

Table 1: Performance comparison of DiT-XL/2 on ImageNet 256×256 under different sampling and guidance configurations. The best results among W4A4 methods are highlighted in bold.

Setting	Bit-width	Method	IS (↑)	FID (↓)	sFID (↓)	Precision (↑)
Step=100 CFG= 1.0	W16A16	FP	90.32	12.25	19.28	0.65
	W4A8	VETA-DiT	75.90	15.22	18.66	0.62
	W4A4	PTQ4DiT	2.08	304.62	129.68	0.07
		Q-DiT	1.93	256.13	429.47	0.01
		ViDiT-Q	25.11	72.23	44.57	0.32
		VETA-DiT	64.91	22.85	**20.50**	0.57
		SyRA-DiT (Ours)	**74.03**	**19.75**	22.45	**0.59**
Step=100 CFG= 1.5	W16A16	FP	168.08	4.56	17.72	0.79
	W4A8	VETA-DiT	152.53	5.29	17.50	0.77
	W4A4	PTQ4DiT	2.42	274.59	117.25	0.08
		Q-DiT	2.14	243.05	410.74	0.02
		ViDiT-Q	51.96	40.87	35.58	0.45
		VETA-DiT	137.98	7.22	**18.90**	0.74
		SyRA-DiT (Ours)	**156.22**	**6.15**	19.11	**0.75**
Step=50 CFG= 1.0	W16A16	FP	88.07	13.38	19.07	0.65
	W4A8	VETA-DiT	75.05	17.30	18.36	0.63
	W4A4	PTQ4DiT	2.05	298.57	126.64	0.08
		Q-DiT	1.81	262.58	421.08	0.01
		ViDiT-Q	18.13	84.40	57.46	0.28
		VETA-DiT	64.15	24.24	21.98	0.56
		SyRA-DiT (Ours)	**74.02**	**19.98**	**21.19**	**0.60**
Step=50 CFG= 1.5	W16A16	FP	164.61	4.86	17.65	0.80
	W4A8	VETA-DiT	147.12	5.53	17.65	0.77
	W4A4	PTQ4DiT	2.33	281.43	119.45	0.08
		Q-DiT	2.02	249.98	404.09	0.01
		ViDiT-Q	48.34	44.15	37.90	0.43
		VETA-DiT	135.66	7.90	19.11	0.73
		SyRA-DiT (Ours)	**156.51**	**6.28**	**18.96**	**0.75**

prior W4A4 methods introduce more noticeable structural and perceptual distortions. These results further demonstrate that the proposed reconstruction-driven rotation effectively preserves generation fidelity under aggressive W4A4 quantization. See Figure 10 in Appendix B.5 for additional qualitative results and visual comparisons.

Method	RFID ↓	rsFID ↓	LPIPS ↓	CMMD ↓
FP16	—	—	—	—
Q-DiT	120.6	286.0	0.735	1.275
ViDiT-Q	136.6	294.0	0.665	1.493
VETA-DiT	42.7	247.8	0.420	0.139
Ours	**35.6**	**222.4**	**0.332**	**0.103**



Figure 4: (**Left**) Text-to-image generation performance of PixArt-Σ Chen et al. (2024) on the COCO dataset. (**Right**) Qualitative comparison of images generated with W4A4 quantization.

Table 2: Performance comparison of Open-Sora on VBench. Notation * denotes mixed precision.

Method	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Scene	Overall Consistency
FP16	0.968	0.971	0.991	0.250	0.531	0.567	0.404	0.256
ViDiT-Q*	0.960	0.962	0.989	0.250	0.513	0.546	0.408	0.257
Q-DiT	0.959	0.958	**0.989**	0.208	0.506	0.539	0.343	0.257
ViDiT-Q	0.958	0.957	**0.989**	0.194	0.509	0.529	**0.382**	0.255
VETA	0.958	**0.960**	**0.989**	0.167	**0.512**	0.533	0.336	**0.260**
SyRA-DiT	**0.960**	0.959	**0.989**	**0.222**	0.508	**0.550**	0.374	0.257

Table 3: Ablation study for the key components of our SyRA-DiT.

Method	IS (↑)	FID (↓)	sFID (↓)	Prec. (↑)
FP (W16A16)	164.61	4.86	17.65	0.80
Baseline	2.02	249.98	404.09	0.01
+ Learnable rotation	143.13	7.80	19.26	0.72
+ Per-timestep incoherence coefficients	156.51	6.28	18.96	0.75

Video Generation with Open–Sora. Table 2 presents the performance comparison of Open-Sora on VBench. SyRA-DiT achieves strong performance under aggressive W4A4 quantization. Among all W4A4 methods, it achieves the best or second-best performance on 7 of 8 VBench metrics, while remaining competitive in both visual quality and temporal consistency. Moreover, SyRA-DiT remains highly competitive with the higher-precision W4A8 ViDiT-Q* baseline, demonstrating its effectiveness in preserving generation quality at substantially lower precision and establishing PTQ as a practical solution even in the challenging W4A4 regime. Additional qualitative results are provided in Figure 11 in Appendix B.5.

5.3 Ablation Study

Table 3 evaluates the key components of SyRA-DiT on ImageNet-256 under W4A4 PTQ. All variants use DiT-XL/2 at 256×256, grouped min-max quantization with group size 128, SmoothQuant with $\alpha = 0.5$, and GPTQ after rotation, under the same 50-step calibration and DDIM sampling protocol. The W4A4 baseline suffers a severe quality collapse, with an FID of 249.98 and an IS of 2.02, compared with an FID of 4.86 and an IS of 164.61 for the full-precision model. Adding the learnable rotation recovers most of the lost performance, reducing the FID to 7.80 and increasing the IS to 143.13. Incorporating incoherence weighting (per-timestep incoherence coefficients) further improves the results to an FID of 6.28, an sFID of 18.96, a precision of 0.75, and an IS of 156.51. Overall, the learnable rotation accounts for the majority of the recovery under aggressive W4A4 quantization, while timestep-aware weighting provides further consistent gains.

6 Conclusion

We presented SyRA-DiT, a reconstruction-driven PTQ framework for aggressive W4A4 quantization of Diffusion Transformers. By learning an orthogonal rotation that jointly reorganizes weight and activation quantization groups across denoising timesteps, SyRA-DiT directly optimizes the reconstruction objective that governs low-bit generation quality. Extensive experiments on image, text-to-image, and video diffusion models show consistent improvements over prior W4A4 methods in both quantitative metrics and perceptual fidelity, while remaining competitive with higher-precision W4A8 baselines. Beyond accuracy, our latency study shows that W4A4 computation yields substantial practical speedups once kernel-launch overhead is reduced, achieving a $1.74\times$ improvement over FP16 under CUDA graph execution together with lower memory usage. These results demonstrate that reconstruction-aware group reorganization, coupled with efficient execution, makes fully 4-bit PTQ a practical path toward accurate and efficient deployment of large diffusion models.

AI USE STATEMENT

Generative AI tools were used for language editing, including grammar and spelling corrections. The authors reviewed the final text and take full responsibility for the content of the manuscript.

ETHICS STATEMENT

This study evaluates post-training quantization on existing image and video benchmarks and does not involve human-subject studies or the collection of new personal data. Improved efficiency may facilitate broader deployment of generative models; downstream uses should therefore follow the safety and usage requirements of the underlying models and datasets.

REPRODUCIBILITY STATEMENT

Implementation code, configuration files, and preparation instructions are available at `https://anonymous.4open.science/r/SyRA-DiT-F17C/`. Pretrained weights and calibration tensors are downloaded or generated following the repository instructions. The main text describes the models, datasets, and evaluation settings; Appendix B.1 gives the calibration and inference algorithm.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36:4396–4429, 2023.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28306–28315. IEEE, 2025.

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.

Zhenyuan Dong and Sai Qian Zhang. Ditas: Quantizing diffusion transformers via enhanced activation smoothing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4606–4615. IEEE, 2025.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. *arXiv preprint arXiv:2401.06118*, 2024.

Weilun Feng, Haotong Qin, Chuanguang Yang, Xiangqi Li, Han Yang, Yuqi Li, Zhulin An, Libo Huang, Michele Magno, and Yongjun Xu. S2 q-vdit: Accurate quantized video diffusion transformer with salient data and sparse token distillation. *arXiv preprint arXiv:2508.04016*, 2025.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

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.

Feice Huang, Zuliang Han, Xing Zhou, Yihuang Chen, Lifei Zhu, and Haoqian Wang. Convrot: Rotation-based plug-and-play 4-bit quantization for diffusion transformers. *arXiv preprint arXiv:2512.03673*, 2025a.

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7362–7371. IEEE, 2024a.

Yushi Huang, Ruihao Gong, Xianglong Liu, Jing Liu, Yuhang Li, Jiwen Lu, and Dacheng Tao. Temporal feature matters: A framework for diffusion model quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10):8823–8837, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024b.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315. IEEE, 2024.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

Donghyun Lee, Jitesh Chavan, Duy Nguyen, Sam Huang, Liming Jiang, Priyadarshini Panda, Timo Mertens, and Saurabh Shukla. Orbitquant: Data-agnostic quantization for image and video diffusion transformers. *arXiv preprint arXiv:2607.02461*, 2026.

Guandong Li. Fastusp: A multi-level collaborative acceleration framework for distributed diffusion model inference. *arXiv preprint arXiv:2602.10940*, 2026.

Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17489–17499. IEEE, 2023a.

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.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17181–17190. IEEE, 2023b.

Zhiteng Li, Hanxuan Li, Junyi Wu, Kai Liu, Haotong Qin, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. Dvd-quant: data-free video diffusion transformers quantization. In *International Conference on Learning Representations*, volume 2026, pp. 35463–35475, 2026.

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.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Wenxuan Liu and Sai Qian Zhang. Hq-dit: Efficient diffusion transformer with fp4 hybrid quantization. *arXiv preprint arXiv:2405.19751*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 467–484, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024a.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024b.

Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16132–16141. IEEE, 2024.

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*, pp. 7197–7206. PMLR, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1981. IEEE, 2023.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45472–45496, 2024.

Yihua Shao, Deyang Lin, Minxi Yan, Siyu Chen, Fanhu Zeng, Minwen Liao, Ao Ma, Ziyang Yan, Haozhe Wang, Yan Wang, et al. Tr-dq: Time-rotation diffusion quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8869–8877, 2026.

Sayeh Sharify, Mahsa Salmani, and Hesham Mostafa. Dirotq: Rotation-aware quantization for 4-bit diffusion transformers. *arXiv preprint arXiv:2605.16732*, 2026.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8815–8821, 2020.

Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in neural information processing systems*, 36:48686–48698, 2023.

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

Shohei Taniguchi, Masahiro Suzuki, and Yutaka Matsuo. Quantization-aware diffusion models for maximum likelihood training. In *International Conference on Learning Representations*, volume 2026, pp. 52140–52157, 2026.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *Proceedings of machine learning research*, 235:48630, 2024.

Gang Wang, Siqi Cai, Zhenyu Li, and Guanghui He. Bsdit: Bit-serial diffusion transformer acceleration with dynamic rotating and adaptive step quantization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2026.

Haoxuan Wang, Yuzhang Shang, Zhihang Yuan, Junyi Wu, Junchi Yan, and Yan Yan. Quest: Low-bit diffusion model quantization via efficient selective finetuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15542–15551. IEEE, 2025.

Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. *Advances in neural information processing systems*, 37: 62732–62755, 2024.

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*, pp. 38087–38099. PMLR, 2023.

Qinkai Xu, Lin Yang, Li Li, Yuxiang Fu, et al. Veta-dit: Variance-equalized and temporally adaptive quantization for efficient 4-bit diffusion transformers. *Advances in Neural Information Processing Systems*, 38:152907–152938, 2026.

Divakar Kumar Yadav and Tian Zhao. Hybrid jit-cuda graph optimization for low-latency large language model inference. *arXiv preprint arXiv:2604.23467*, 2026.

Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7300–7308. IEEE, 2019.

Lianwei Yang, Haokun Lin, Yichen Wu, Zhenan Sun, and Qingyi Gu. Dapq-dit: Distribution-aware post-training quantization for efficient generative tasks in diffusion transformers. In *Proceedings of the 2026 International Conference on Multimedia Retrieval*, pp. 2371–2380, 2026a.

Lianwei Yang, Haokun Lin, Tianchen Zhao, Yichen Wu, Hongyu Zhu, Ruiqi Xie, Zhenan Sun, Yu Wang, and Qingyi Gu. Lrq-dit: Log-rotation post-training quantization of diffusion transformers for image and video generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026b.

Qian Zeng, Chenggong Hu, Mingli Song, and Jie Song. Diffusion model quantization: A review. *arXiv preprint arXiv:2505.05215*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

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

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

A Appendix

A.1 Timestep

B Temporal Mixture Statistics and GPTQ Calibration

Here, we establish two claims: (a) aggregating activations across timesteps before computing covariance or GPTQ statistics eliminates the timestep-specific component of the activation distribution; and (b) when rotation optimization and weight rounding are based on different temporal mixtures, the rounded weights minimize a quadratic form that is misaligned with the reconstruction objective. We further bound the resulting loss inflation by a generalized condition number, which reduces to one when the two temporal mixtures coincide.

We first recall the problem setup from Section 4.2. A linear layer

$$Y_t = X_t W^\top + b$$

is reused across denoising steps $t = 1, \dots, T$, where $X_t \in \mathbb{R}^{N \times C}$. We assume token alignment across timesteps: row i of each X_t corresponds to the same latent position of the same calibration sample. Given a shared rotation $R \in \mathrm{O}(C)$, the quantized layer output is

$$\hat{Y}_t(R) = Q_a(X_t R)\, Q_w(WR)^\top + b,$$

where Q_a is a dynamic per-token grouped min–max quantizer and Q_w denotes static weight quantization with GPTQ rounding. For each timestep, define the second moment, mean, and within-step covariance as

$$G_t = \frac{1}{N} X_t^\top X_t, \qquad \mu_t = \frac{1}{N} X_t^\top \mathbf{1}, \qquad C_t = G_t - \mu_t \mu_t^\top.$$

For weights $\alpha \in \Delta^{T-1}$, we distinguish two forms of temporal aggregation. The *timestep-weighted* statistics are

$$G_\alpha = \sum_t \alpha_t G_t, \qquad C_\alpha = \sum_t \alpha_t C_t,$$

which are respectively the second moment and within-step covariance obtained by first sampling a timestep $t \sim \alpha$ and then sampling a token from that timestep. By contrast, the temporal-average setup Xu et al. (2026) are computed from the token-wise weighted average

$$\bar{X}_\alpha = \sum_t \alpha_t X_t.$$

Specifically,

$$G_\alpha^{\mathrm{pool}} = \frac{1}{N} \bar{X}_\alpha^\top \bar{X}_\alpha, \qquad C_\alpha^{\mathrm{pool}} = G_\alpha^{\mathrm{pool}} - \bar{\mu}_\alpha \bar{\mu}_\alpha^\top, \qquad \bar{\mu}_\alpha = \sum_t \alpha_t \mu_t.$$

Thus, the rows of $\bar{X}_\alpha$ correspond to synthetic activations

$$\bar{x} = \sum_t \alpha_t x_t,$$

rather than activations observed at any individual inference step.

Finally, the rotation is optimized using the normalized reconstruction objective

$$\mathcal{L}(R) = \sum_{t=1}^{T} \omega_t \frac{\|Y_t - \hat{Y}_t(R)\|_F^2}{\|Y_t\|_F^2 + \varepsilon}, \qquad \tilde{\omega}_t := \frac{\omega_t}{\|Y_t\|_F^2 + \varepsilon}, \qquad G_{\tilde{\omega}} := \sum_t \tilde{\omega}_t G_t. \tag{P}$$

The effective timestep weights entering the quadratic reconstruction geometry are therefore $\tilde{\omega}_t$, rather than ω_t alone.

We formalize this analysis in the following propositions in the following.

Proposition 1 (Consistent GPTQ Hessian). *Given learned R, let $Z_t = X_t R$, $Q_a(Z_t) = Z_t + E_{a,t}$, $Q_w(WR) = WR + E_w$, then $\hat{Y}_t - Y_t = Z_t E_w^\top + E_{a,t}(WR)^\top + E_{a,t}E_w^\top$, and the $\hat{W}$-dependent first-order part of equation P is*

$$\sum_t \tilde{\omega}_t \|Z_t E_w^\top\|_F^2 = N \ \mathrm{tr}\left(E_w H_{\tilde{\omega}} E_w^\top\right), \qquad H_{\tilde{\omega}} = R^\top G_{\tilde{\omega}} R. \tag{11}$$

Thus the row-wise quadratic that GPTQ rounds against coincides with the weight part of equation P if its Hessian is the $\tilde{\omega}$-mixture second-moment matrix in the rotated basis.

Proof. By orthogonality of R,

$$Y_t = X_t W^\top + b = X_t R R^\top W^\top + b = Z_t (WR)^\top + b,$$

where $Z_t = X_t R$. Substituting

$$Q_a(Z_t) = Z_t + E_{a,t}, \qquad Q_w(WR) = WR + E_w$$

into

$$\hat{Y}_t = Q_a(Z_t)\, Q_w(WR)^\top + b$$

gives

$$\hat{Y}_t = Z_t (WR)^\top + Z_t E_w^\top + E_{a,t}(WR)^\top + E_{a,t}E_w^\top + b.$$

Hence,

$$\hat{Y}_t - Y_t = Z_t E_w^\top + E_{a,t}(WR)^\top + E_{a,t}E_w^\top.$$

The term $E_{a,t}(WR)^\top$ is independent of the weight-rounding error E_w, while $E_{a,t}E_w^\top$ is second order in the quantization errors. Thus, to first order, the component affected by GPTQ is $Z_t E_w^\top$.

For $A \in \mathbb{R}^{N \times C}$ and $B \in \mathbb{R}^{C_{\mathrm{out}} \times C}$,

$$\|AB^\top\|_F^2 = \mathrm{tr}\big((AB^\top)^\top AB^\top\big) = \mathrm{tr}\big(BA^\top AB^\top\big).$$

Taking $A = Z_t$ and $B = E_w$ yields

$$\|Z_t E_w^\top\|_F^2 = \mathrm{tr}\big(E_w Z_t^\top Z_t E_w^\top\big).$$

Since $Z_t = X_t R$,

$$Z_t^\top Z_t = R^\top X_t^\top X_t R = N\, R^\top G_t R.$$

Therefore, weighting by $\tilde{\omega}_t$ and summing over timesteps gives

$$\sum_t \tilde{\omega}_t \|Z_t E_w^\top\|_F^2 = N \ \mathrm{tr}\left(E_w R^\top \Big(\sum_t \tilde{\omega}_t G_t\Big) R E_w^\top\right) = N \ \mathrm{tr}\big(E_w H_{\tilde{\omega}} E_w^\top\big).$$

Let $e_r^\top$ denote the r-th row of E_w. Then

$$\mathrm{tr}(E_w H E_w^\top) = \sum_r e_r^\top H e_r.$$

GPTQ minimizes, for each output row r, a quadratic form in the row-wise weight-rounding error with a shared Hessian H. Hence, its quadratic objective matches the first-order weight-quantization component of Eq. equation P exactly when

$$H = H_{\tilde{\omega}}.$$

The per-step normalization in Eq. equation P enters this Hessian only through the effective weights

$$\tilde{\omega}_t = \frac{\omega_t}{\|Y_t\|_F^2 + \varepsilon}$$

applied to the per-step second-moment matrix. This completes the proof. □

Remark 1. *Proposition 1 characterizes the Hessian under which GPTQ optimizes the weight-rounding error for the same temporally weighted objective used to select R. If GPTQ instead uses a Hessian constructed from a different temporal statistic G_β—arising, for example, from a different timestep weighting or from the temporally averaged statistic G_α^{pool}, then it minimizes*

$$\mathrm{tr}(ER^\top G_\beta RE^\top),$$

whereas the reconstruction objective penalizes

$$\mathrm{tr}(ER^\top G_{\tilde{\omega}} RE^\top).$$

Their discrepancy is therefore

$$\mathrm{tr}\big(ER^\top (G_{\tilde{\omega}} - G_\beta) RE^\top\big).$$

Thus, GPTQ is optimized under a quadratic geometry that omits the between-timestep variation captured by $S_{\tilde{\omega}}$; the unmodeled error is confined to $\mathrm{range}(R^\top S_{\tilde{\omega}} R)$, i.e., the rotated directions along which the inputs vary across timesteps.

B.1 SyRA-DiT Algorithm

For a clearer overview of SyRA-DiT, Algorithm 1 summarizes the complete calibration and inference pipeline.

B.2 Is a Unified Rotation Sufficient for Weights and Activations

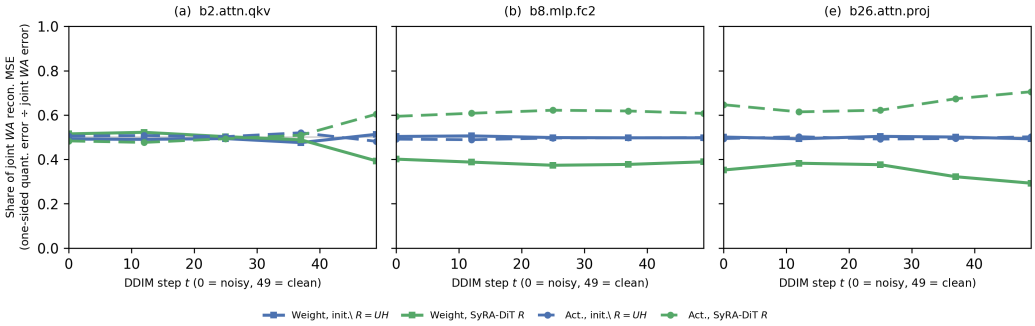


Figure 5: Weight and activation contributions to the joint W4A4 reconstruction error, comparing the initialization $R = UH$ with SyRA-DiT at randomly selected layers.

Our design fundamentally differs from learnable rotation methods for LLM quantization Ashkboos et al. (2024); Liu et al. (2025), which optimize module-specific rotations under a static inference regime. We further investigate variants with decoupled rotations for weights and activations, following a similar design philosophy to Liu et al. (2025). In our experiments, however, these decoupled rotations converge more slowly and yield inferior quantization performance. We attribute this to the loss of coordination between the activation and weight quantizers. By contrast, SyRA-DiT uses a shared rotation for both operands, allowing the optimization to directly account for their interaction in the quantized matrix product and consistently achieving the best performance.

We further analyze the weight and activation contributions to the joint W4A4 reconstruction error in Figure 5. As shown in Figure 5, panels (a), (b), and (e) decompose the joint weight–activation reconstruction MSE at each DDIM step t into weight-only and activation-only contributions, using a common denominator for both. Solid and dashed curves denote the weight and activation contributions, respectively. The blue curves correspond to the initialization $R = UH$, while the green curves correspond to the learned SyRA-DiT rotation with the same U and H but an optimized post-Hadamard factor. At initialization, the two contributions are approximately balanced. After optimization, SyRA-DiT redistributes the error between weights and activations in a layer- and timestep-dependent manner. For example, panel (b) stabilizes near a $0.4/0.6$ weight–activation split, whereas panels (a) and (e) exhibit stronger variation toward later denoising steps. These results indicate that the gains of SyRA-DiT do not arise from suppressing only weight or activation error; instead, the learned rotation jointly reshapes how 4-bit quantization error enters the layer output.

Algorithm 1 SyRA-DiT Algorithm

Require: Calibration activations $\{X_t\}_{t=1}^{T}$; full-precision W, b from pretrained DiT; SmoothQuant scale s; group size g; optimization steps K; $\varepsilon = 10^{-6}$

Ensure: Orthogonal transform $R^\star$ and quantized weight $\hat{W}$

$D_s \leftarrow \mathrm{diag}(s)$

$X_t \leftarrow X_t D_s$ for all t, $\quad W \leftarrow W D_s^{-1}$

$Y_t \leftarrow X_t W^\top + b$ for all t

Step 1: Trajectory-aware statistics

for $t = 1, \ldots, T$ **do**

  $\mu_t \leftarrow \sqrt{NC}\, \max_{ij} |(X_t)_{ij}| / \|X_t\|_F$

  $\bar{x}_t \leftarrow N^{-1} X_t^\top \mathbf{1}$

  $G_t \leftarrow N^{-1} X_t^\top X_t$

  $\Sigma_t \leftarrow G_t - \bar{x}_t \bar{x}_t^\top$

end for

$\omega \leftarrow \mathrm{softmax}(\mu_1, \ldots, \mu_T)$

$G_\omega \leftarrow \sum_{t=1}^{T} \omega_t G_t, \quad \Sigma_\omega \leftarrow \sum_{t=1}^{T} \omega_t \Sigma_t$

$U \leftarrow \mathrm{eigvecs}(\Sigma_\omega)$; freeze U

Draw and fix a normalized randomized Hadamard transform H

Step 2: Joint W4A4 rotation refinement

Initialize $A \leftarrow 0$

for $k = 1, \ldots, K$ **do**

  $S \leftarrow A - A^\top$

  $R \leftarrow UH(I - S)(I + S)^{-1}$

  for $t = 1, \ldots, T$ **do**

    $\hat{Y}_t \leftarrow Q_a^{\mathrm{STE}}(X_t R)\, Q_w^{\mathrm{STE}}(WR)^\top + b$

  end for

  $\mathcal{L} \leftarrow \sum_{t=1}^{T} \omega_t \dfrac{\|Y_t - \hat{Y}_t\|_F^2}{\|Y_t\|_F^2 + \varepsilon}$

  Update A with Adam using STE through the rounding operations

end for

$R^\star \leftarrow UH(I - S^\star)(I + S^\star)^{-1}, \quad S^\star = A^\star - A^{\star\top}$

Step 3: Consistent GPTQ calibration

$G_R \leftarrow R^{\star\top} G_\omega R^\star$

$\hat{W} \leftarrow \mathrm{GPTQ}(WR^\star; G_R, g)$

Inference

Output: For input x, compute $\hat{y} = Q_a(xD_s R^\star)\hat{W}^\top + b$

B.3 Explainable results

Figure 6 presents per-channel activation outliers reduction. Figure 7 compares the outlier severity score for activations (left) and weights (right) at DDIM step $t = 25$ across five DiT-XL/2 layers, labeled (a)–(e). The score is defined as $\max|Z|/\mathrm{RMS}(Z)$ for rotated activations $Z = XR$ after SmoothQuant scaling and $\max|W_{\mathrm{th}}|/\mathrm{RMS}(W_{\mathrm{th}})$ for rotated weights $W_{\mathrm{th}} = W_s R$, where $\mathrm{RMS}(\cdot) = \sqrt{\mathrm{mean}(\cdot^2)}$. Higher scores indicate sharper outliers relative to the typical tensor magnitude and therefore greater difficulty for per-group min–max quantization. Grouped bars compare SmoothQuant with $R = I$, VETA-DiT with the fixed rotation, and SyRA-DiT with learned layer-wise rotations. Both rotation-based methods generally reduce outlier severity relative to SmoothQuant, with the largest activation reductions occurring in the mid-depth MLP `fc2` layers, particularly panels (b) and (c).

Figure 8 compares signed activation distributions at block 14 MLP fc2, diffusion step 25, using 2,048 calibration tokens. Without rotation (gray), activations are strongly heavy-tailed and spike-dominated, with variance 0.032 and excess kurtosis 52.5. VETA-DiT (blue) substantially regularizes the distribution, reducing the variance to 0.012 and excess kurtosis to 0.40. SyRA-DiT (green) yields a similarly compact, near-Gaussian distribution with the same variance and lower excess kurtosis of 0.28. Together with the channel-max results, this shows that rotation suppresses activation outliers

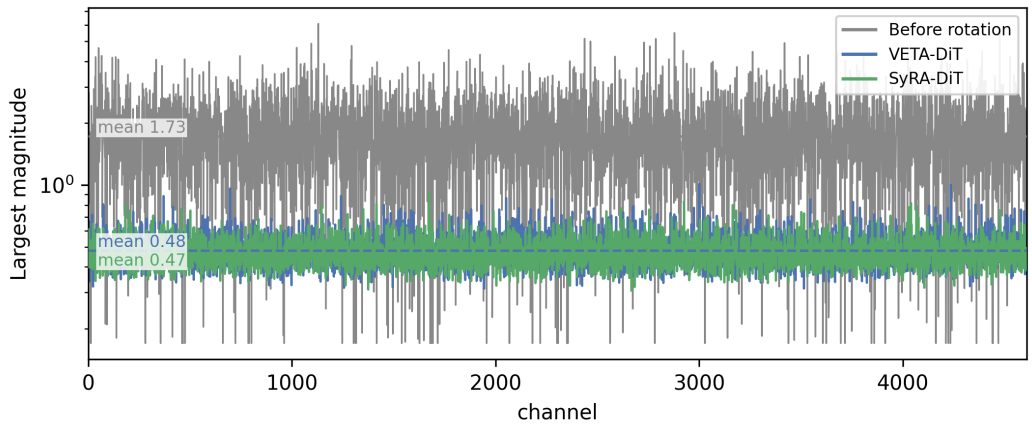


Figure 6: Per-channel activation outliers at `blocks.14.mlp.fc2` (b14.mlp.fc2) at DDIM timestep $t = 25$. We report the largest magnitude of the outlier in each input channel after SmoothQuant scaling and rotation $Z = XR$, comparing no rotation ($R = I$), VETA-DiT, and SyRA-DiT. Boxplots summarize 4608 channel-wise peaks from 2048 calibration tokens.

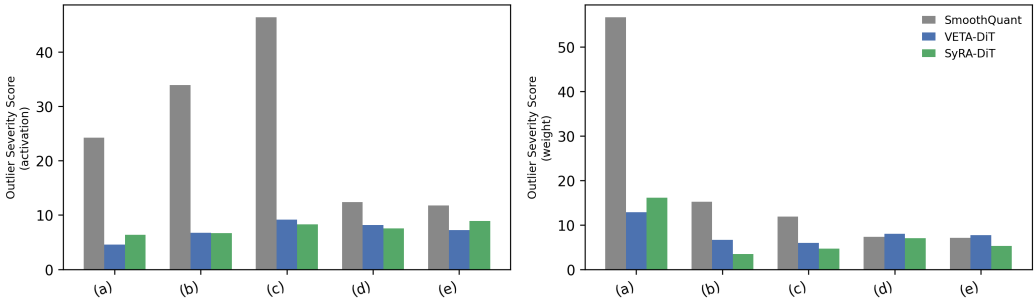


Figure 7: Outlier severity score for activations $Z = XR$ (left) and weights $W_{\text{th}} = W_s R$ (right) at DDIM timestep $t = 25$. Panels (a)–(e) correspond to b2.attn.qkv, b8.mlp.fc2, b14.mlp.fc2, b20.attn.proj, and b26.attn.proj, respectively. Bars compare SmoothQuant ($R = I$), VETA-DiT, and SyRA-DiT.

and redistributes mass toward the bulk, with SyRA-DiT achieving lighter tails and improved activation SQNR under W4A4 quantization.

B.4 Inference Latency

Following prior works Zhao et al. (2024); Yadav & Zhao (2026); Li (2026), we evaluate inference latency under both *Eager* and *CUDA Graph* execution settings. Table 4 reports the cost of generating a single ImageNet-256 image with DiT-XL/2 using 50-step DDIM sampling, classifier-free guidance of 1.5, and batch size 1 on an A100 GPU. All configurations execute the same 50 denoising steps. In the *Eager* setting, LayerNorm, GEMM, and attention kernels are launched separately at each step, whereas *CUDA graph* captures the DiT forward pass at a fixed guidance shape and replays it across subsequent steps, substantially reducing kernel-launch overhead. Since CUDA graph replay does not alter the model weights or arithmetic, peak memory remains nearly unchanged within each precision setting.

CUDA graph replay accounts for most of the latency improvement. For FP16, latency decreases from 1355.8 to 347.6 ms/img, corresponding to a $3.90\times$ speedup. For W4A4, latency drops from 1334.5 to 199.6 ms/img, yielding a $6.69\times$ speedup over W4A4 eager and a $6.79\times$ speedup over FP16 eager. More importantly, under the same CUDA graph setting, W4A4 is $1.74\times$ faster than FP16 (199.6 vs. 347.6 ms/img) while substantially reducing peak memory. In contrast, W4A4 pro-

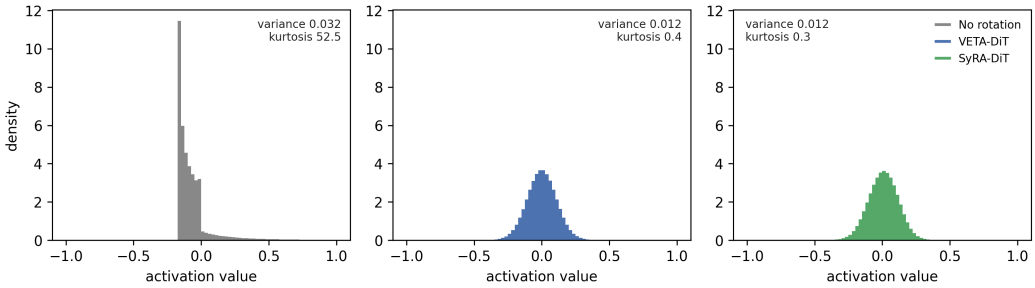


Figure 8: Activation distributions before and after rotation for `blocks.14.mlp.fc2` at DDIM timestep $t = 25$, using a 2048-token calibration subsample. Left: activations before SmoothQuant scaling and rotation. Center: activations after SmoothQuant and VETA-DiT rotation. Right: activations after SmoothQuant and SyRA-DiT rotation.

Table 4: End-to-end DiT cost for ImageNet-256 generation (DDIM 50 steps, CFG 1.5, batch 1). Both FP16 rows and both W4A4 rows use the same 50 denoising steps. "Eager" launches each GPU kernel separately. "CUDA graph" records that DiT forward once and replays it on later steps. Reduction and speedup are relative to FP16 eager.

Method	ms/img ↓	Memory (MB) ↓	Reduction ↑	Speedup ↑	vs. FP16 graph ↑
FP16 eager	1355.8	1659.0	1.00×	1.00×	0.26×
FP16 CUDA graph	347.6	1674.1	0.99×	3.90×	1.00×
W4A4 eager	1334.5	1069.8	1.55×	1.02×	0.26×
W4A4 CUDA graph	199.6	1072.8	1.55×	6.79×	1.74×

vides only a $1.02\times$ speedup in eager execution, as kernel-launch overhead dominates and largely masks the benefit of low-bit computation. These results highlight that reducing launch overhead is critical to realizing the practical efficiency gains of W4A4 quantization.

B.5 Additional Qualitative Results

We provide additional qualitative results to further evaluate SyRA-DiT. Most examples are generated under aggressive W4A4 quantization, with FP16 and selected W4A8 results included for comparison. Figure 9 shows ImageNet samples generated by DiT-XL/2, while Figure 10 presents corresponding results for PixArt-Sigma. Figure 11 further visualizes representative frames from randomly generated videos. Overall, these results consistently demonstrate that SyRA-DiT effectively preserves generation quality even under the challenging W4A4 quantization regime.

B.6 Limitations

Limitations. SyRA-DiT introduces a one-time calibration and rotation optimization cost, and its realized latency benefits depend on the availability and efficiency of low-bit kernels and execution backends. Although our experiments cover multiple Diffusion Transformer settings, we have not evaluated substantially different generative architectures, such as U-Net-based diffusion models, state-space models, or autoregressive image/video generators. In addition, our system-level evaluation is primarily conducted on NVIDIA GPUs; performance and practical speedups on other hardware platforms, including AMD GPUs, TPUs, mobile NPUs, and other dedicated AI accelerators, remain to be investigated.

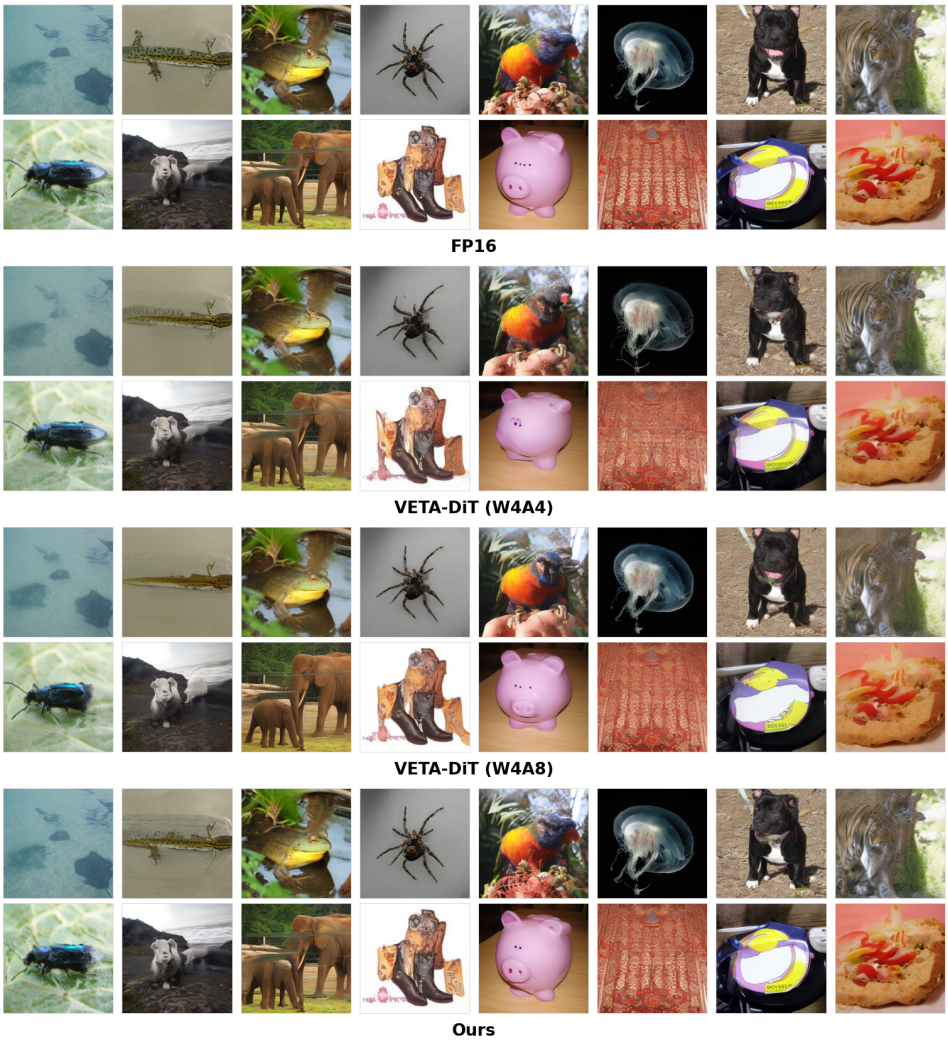


Figure 9: The qualitative results on ImageNet.

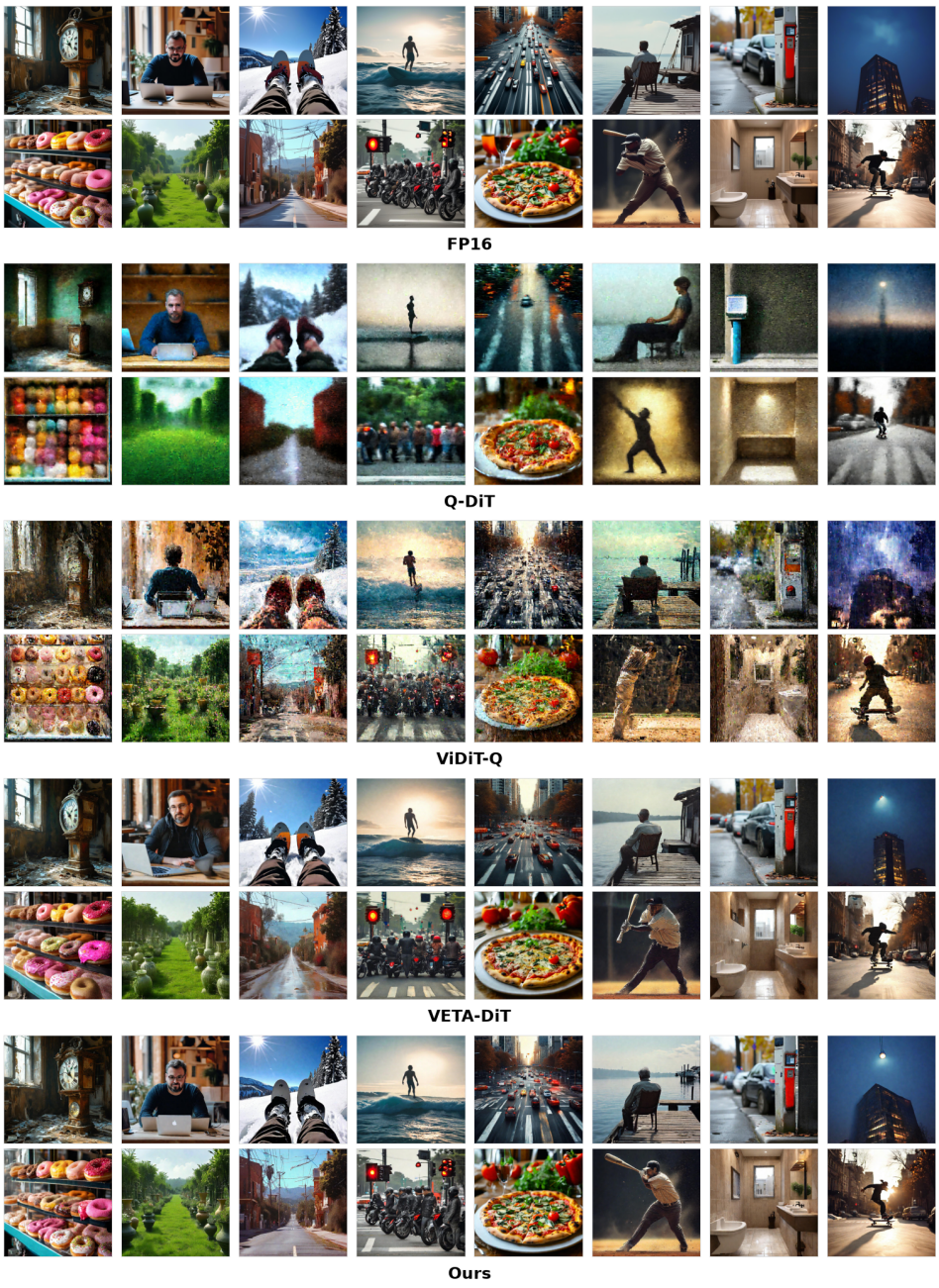


Figure 10: The qualitative results on COCO dataset.



Figure 11: The qualitative results on VBench.